K-Linkable Ring Signatures and Applications in Generalized Voting

Wonseok Choi[1,2], Xiangyu Liu[1,2], Lirong Xia[3], and Vassilis Zikas[2]

[1] Purdue University
[2] Georgia Institute of Technology
[3] Rutgers University
wonseok@purdue.edu, liu3894@purdue.edu, lirong.xia@rutgers.edu, vzikas@gatech.edu

Abstract. *Linkable ring signatures* (LRS) allow a user to sign anonymously on behalf of a ring, while maintaining linkability—two signatures from the same signer are publicly identified, i.e., linked. This linkability makes LRS suitable to prevent double-voting in classical, *plurality* voting protocols—each voter casts one vote and the candidate with the most votes wins the election.

Several voting scenarios rely on (generalized) rules rather than plurality. For example, in *ranked voting,* voters submit a ranking of the candidates, and the outcome is a function of these rankings. Such generalized voting rules are common in social choice theory, and have recently found their way into blockchain governance, e.g., for prioritizing (voting on) proposed (candidate) projects. However, unlike plurality voting, using LRS for voters to sign their votes (rankings) does not guarantee vote privacy as one can observe the rankings of each individual voter, which, depending on the scoring rule, is more information than what the outcome of the election offers.

We introduce *k-linkable ring signatures* (k-LRS) as a primitive for simultaneously achieving anonymity and privacy in generalized voting. A k-LRS scheme has the following properties:

(k-)Anonymity: a user can sign anonymously (on behalf of the ring) up to k times, so that even an unbounded adversary cannot link his signatures.

(k-)Linkability: If any signer signs more than k times, all his signatures are publicly linked *(individual k-linkability)*; and, any set of c signers cannot generate more than $k \cdot c$ unlinked signatures *(collective k-linkability)*.

We provide two constructions of k-LRS: one is from the DDH, and the other is from SIS (hence post-quantum). Finally, we show how k-LRS can be applied to a broad range of voting rules, including *score voting*, *multi-voting*, and *Borda*. Our protocols are non-interactive voting—each voter just posts a message on a bulletin board—which highlights the potential of k-LRS in blockchain-governance scenarios.

Table of Contents

1 Introduction

The overwhelming majority of the literature on cryptographic voting considers so-called *plurality voting* [Cha81, BT94a]—each voter casts a vote for one out of many candidates, and the candidate with the most votes wins. Furthermore, the focus is on eliminating trust in third parties by means of a (usually interactive) protocol among the voters [BY86, CGS97, LWW04]. Such a protocol would typically assume voters have access to a *bulletin board* (in short, BB), as well as several hybrid functionalities (like a voter registration authority, an election authority, etc. [KZZ15b, Adi08, KZZ15a, CZZ+16], and primarily aim to ensure *(vote) privacy*—the protocol does not leak to the adversary anything beyond the outcome of the election. Other desirable properties like receipt-freeness [AOZZ15] and public verifiability [KZZ15b], are also guaranteed by existing protocols.

Parallel to (and independently of) the cryptographic literature, social choice theory has long studied voting, albeit from an orthogonal angle. The goal here is to study the functionality and voter behavior in a setting, where one assumes a fully trusted authority that privately collects votes, performs the tally, and announces the result, e.g, as in national elections. A central theme here is to investigate alternative (generalized) voting and scoring/tallying rules that can improve such functionality and incentivize some desired voter behavior, e.g., improve fairness, boost participation, increase truthfulness, etc. Due to the difference in scope, with only a new notable exception, e.g., [LLXZ20], the question of vote privacy against an authority that cannot be trusted to keep the votes private (e.g., puts them on a BB) has received little attention in this literature.

The advent of blockchains has created a common venue for the two, traditionally disjoint disciplines to innovate. Indeed, the plurality rule typically falls short from implementing the functionality one typically needs in blockchain governance [CLZ23, KL22, KAPS20, VH21, GMS21]. For example, the ability to rank proposals according to the users' preferences is at the core of several decentralized autonomous organizations (DAOs); incentivizing honest behavior and participation to the blockchain's governance mechanism is key to the security of the blockchain itself. The above makes the social-choice-theory findings on generalized voting rules highly relevant for such innovation.

However, the model of centralized, complete trust used in social choice theory does not correspond to such governance applications, where users are typically interested in privacy of the voting process. At the same time in such a setting one is typically interested in non-interactive solutions, where voters post their preferences/votes, and in particular in such solutions which avoid making assumptions about trusted authorities. Indeed, such assumptions go against the principles of modern blockchain systems.

Thus, the motivating question of our work is:

> *How can we design non-interactive, voter-private (and anonymous)*[4] *voting mechanisms, which do not assume complex trusted-setup authorities, rely on standard cryptographic assumptions, and allow for implementing generalized voting rules?*

To answer the above question, we introduce and instantiate a new cryptographic signing primitive, termed *k-linkable ring signatures* (in short, k-LRS), an adaptation of classical linkable ring signatures [LWW04], that is tuned for solving the above problem. Recall that standard ring signatures [RST01] enable a member of a group to sign a message as an *anonymous* group member, i.e., without revealing its identity. In a nutshell, our k-linkability property ensures that group members can sign up to k times while preserving anonymity, but if they issue more than that many signatures (on related messages/events) then this is detected and their signatures are *linked*. As we shall see, this asymmetry allows the design of simple, non-interactive private (and anonymous) voting mechanisms without assuming a trusted authority, but merely assuming an *anonymous (posting) bulletin board (ABB)*. Such an ABB could, for instance, be obtained by posting to a blockchain ledger using an anonymous communication network, like TOR. Importantly, in this work, we aim for unconditional privacy, i.e., privacy against unbounded adversaries. Besides being a theoretically interesting goal, unconditional privacy is, in our opinion, a desirable property for anonymity of voting data residing on a long-lived, public-state system like a blockchain ledger.

[4] In the context of voting, vote privacy, or simply, privacy, is a stronger property than anonymity, as the latter only requires that a vote cannot be traced back to the voter.

1.1 Our Contributions

To better motivate the concept of k-linkability, and in particular its suitability for non-interactive generalized voting over an anonymous bulletin board (ABB), we discuss why standard (1-)linkable ring signatures fall short from solving the problem. To this direction, let us start with the following naïve solution using standard (existentially unforgeable) signatures: Each voter signs their vote (via digital signatures) and posts to the ABB the message. Clearly, the above does not preserve voter anonymity (hence also privacy): Any known association of a user with his public key links the user to his vote.

Instead, one can use ring signatures [RST01], which were designed exactly to preserve anonymity in such situations. In a nutshell, ring signatures allow voters to post anonymously to the ABB. However, this anonymous signing introduces a new issue: In standard (plurality) voting, every voter is supposed to vote for exactly one candidate. But the anonymity of ring signatures, which, recall, was introduced to solve the problem, now becomes a problem of its own, as one cannot catch users who post more than one vote to the ABB.

One could rectify this by assuming an (election) registration authority that issues unique anonymous voting-credentials—so that if one uses the credentials twice, then they will be exposed [KZZ15b]. However, this solution requires trust to such a registration authority, which is what we are aiming to remove. Another idea would be to use linkable ring signatures [LWW04]. Such signatures are aimed exactly at solving the above conundrum: as long as every signer signs (at most) one message, they enjoy the anonymity guarantees of ring signatures. But once more than one signatures are issued by a signer on related messages (e.g., more than one vote on the same candidate) then this is detected and these votes are linked (and can, therefore, be removed). We refer to such signatures as *1-linkable ring signatures*.

The above works for plurality, but fails for voting rules where the voter is expected to give more structured votes, e.g., indicate his approval for each candidate (a.k.a., *approval voting*). In such settings, we are interested in aggregate approval (by all voters) without revealing each individual's preference. Hence, one cannot sign and post his approval. This is exactly where k-linkable ring signatures come to the rescue.

k-Linkable Ring Signature (k-LRS). In a nutshell, a k-linkable ring signature (k-LRS) scheme is an extension of standard 1-linkable ring signatures to allow anonymity of signing for up to k times. More concretely, it has the following attractive properties.

- *k(-Sign)-Anonymity.* A user can sign anonymously, with the same key, on behalf of a ring/group under one event label—this could be a candidate's name— up to k times. ($k \in \mathbb{Z}^+$ is a parameter of the scheme.) k-LRS inherits all advantages of (ordinary) LRS schemes, including no group manager, the flexibility of the ring choice, etc.
- *k-Linkability.* A collusion of c users cannot generate more unlinked signatures than they are allowed to, i.e., to generate more than $k \cdot c$ signatures that are unlinked.
- *Non-slanderability:* This is a standard property of LRS which ensures that a malicious signer (or coalition of signers) cannot "incriminate" honest signers. In particular, if a signer/public-key has signed no more than k signatures (under some event label) then these signatures will not be part of any linked sets no matter what malicious signers do.[5]

Looking ahead, the literature (on (1-)linkable ring signature) has considered two versions of linkability. The initial version, which we refer to as *weak linkability* [LWW04, TW05, LASZ14] only requires the infeasibility of generating two unlinked signatures given one secret key. This can be seen as linkability [BGK+21, XLAZ24] for adversaries who corrupt any *one* of the singers in the ring. As demonstrated in [LW06, BGK+21] and by the attack [HC24] on the linkability of DualDory [BEHM22], this notion is insufficient against adversaries that corrupt more than one party. In particular, we describe an attack on Liu

[5] Note that this property, which, in standard signatures follows trivially from existential unforgeability, is not as easy to achieve on a ring s setting, where signers' signatures are anonymous within the ring.

et al.'s weak linkable ring signatures [LASZ14], which allows a set of *two* colluding parties to generate *three* signatures without being pairwise linked. This attack makes weak linkability inappropriate for our goals.[6]

Notwithstanding, more recent work [BKP20, BGK+21, XLAZ24] introduced the stronger linkability property for (1)-LRS: any set of c signers/public-keys cannot generate more than $c \cdot k$ signatures (even when colluding) without being pairwise detected/linked. Our definition of k-linkability follows the latter, stronger paradigm.

Instantiations of k-LRS. Equipped with the definition of k-linkable signatures, we proceed to instantiations. We provide instantiations based on the discrete logarithm (DL) assumption and the decisional Diffie-Hellman (DDH) assumption, as well as a post-quantum construction, based on the short integer solutions (SIS) assumption. Our constructions improve the state of the art in several ways: First, they are the first to achieve the notion of k-linkability for $k \geq 2$. Second, with the exception of our DDH-based construction, our instantiations achieve unconditional anonymity and strong linkability. As a side result, in Appendix A we describe an attack showing that the prior work of unconditionally-anonymous LRS (which, recall, are for $k = 1$) by Liu et al. [LASZ14] only achieve weak linkability. Our results on k-LRS are summarized in Table 1.

Table 1. Summary of our k-LRS constructions. All our constructions achieve (computational) unforgeability, k-linkability, and non-slanderability. Note that unconditional anonymity is not strictly stronger than computation anonymity here due to the restriction of signing times in total. See Sec. 1 for more details.

	Unforgeab.	**Linkab.**	**Non-Sland.**	**Anon.**	**Assump.**
Scheme 1 (Sec. 6 and E.1)	✓	✓	✓	uncond.	DL
				comp.	DDH
Scheme 2 (Sec. 6 and E.2)	✓	✓	✓	uncond.	SIS
				comp.	

Application in Voting. We next show how k-LRS can be used to implement private (and, therefore, anonymous) non-interactive voting using (only) an anonymous bulletin board (ABB). We provide an appropriate definition for the security of such non-interactive voting over ABB in Sec. 4 and associated constructions. Our constructions focus on score voting and multi-voting (see Sec. 1.3 for an overview). Since they are non-interactive, our construction follows the following paradigm: Each voter posts an appropriate vector of k-linkable signatures to the ABB, and the outcome is computed on the state of ABB after everyone has posted. The novelty of our construction lies in devising appropriate (combinations of) events so that the scheme satisfies the desired rules.

Our voting schemes can be instantiated with any of our (unconditional) anonymous k-LRS schemes and achieve (unconditional) privacy. All our schemes achieve computational public verifiability (under the assumption of the associated k-LRS scheme, i.e., DDH, or SIS). That is, every voter can verify that his vote is accounted for—this follows trivially from the fact that the result of the voting is computed based on public data that appear on the ABB—so the voter needs to simply verify that his vote appears on the ABB. Table 2 gives a summary of our results in generalized voting.

In Appendix J, we also show how k-LRS can be used to instantiate additional voting rules. Along the way we also provide a simple extension of k-LRS, which we term multi-event $\boldsymbol{k}$-LRS (in short, $\boldsymbol{k}$-MLRS where $\boldsymbol{k}$ is a vector) which incorporates multiple event into the definition of (k-)LRS in Appendix F. Ranked voting, including the Borda count, can be realized by our notably simple Mode 3 using $(1, 1)$-MLRS in Appendix I.

We note that for a typical governance scenario, k_1 (the number of votes each voter can cast, namely the maximum rank) would typically be much smaller (constant) in the number of voters n. As blockchain

[6] [BGK+21] discusses the insufficiency of the security model in [LASZ14] (and [LW06, ACST06]), but does not provide a concrete attack. To the best of our knowledge, ours is the first work to provide an actual linkability attack on [LASZ14].

Table 2. Our non-interactive ABB-based voting protocols: three types of voting rules (modes), i.e., score-voting, multi-voting, and the Borda count. For Mode 1 and 2, k is the parameter of the underlying k-LRS scheme (or $\boldsymbol{k}$ for Mode 3 using $\boldsymbol{k}$-MLRS), and k_1, k_2, and q are voting parameters, where k_1 represents the number of votes each voter can cast, $k_2 \leq k_1$ defines the maximum votes a voter can allocate to a single candidate, and q denotes the total number of candidates. Anonymity—i.e., the protection of voters' identity—of Mode 1 and 2 is unconditional and follows directly from the unconditional anonymity of k-LRS (more precisely, 1-LRS). All schemes have (computational) public verifiability. With regards to privacy, for Mode 1 we get unconditional privacy, whereas for Mode 2, the amount of privacy (unconditional or computational) depends on how we set the parameter k of our k-LRS. For Mode 3, anonymity and privacy are both computational. The last column reports the complexity of detecting violations of the voting rules—by using the k-linkability property.

	Security		**Pub.-Ver.**	**Detect. Complexity**
	Anon.	Privacy		
Mode 1, Sec. 4.3 (e.g., score voting)	uncond.	uncond. ($k = O(k_1)$)	✓	$\binom{nk_1}{k_1+1}$
Mode 2, Sec. 4.5 (e.g., multi-voting)	uncond.	uncond. ($k = O(k_1^2 + k_1 q)$)	✓	$\binom{nk_1}{k_1+1} + O(nk_1 q)$
		comp. ($k = O(k_1 + k_2 + 1)$)		$\binom{nk_1}{k_1+1} + O(n(k_1 + q))$
Mode 3, App. I (e.g., Borda count)	comp.	comp. ($\boldsymbol{k} = (1, 1)$, MLRS)	✓	$\binom{nq}{2}$

governance systems scale, they tend to rely on long voting periods, e.g., Algorand,[7] and/or to transition to a form of representation voting, where voters allocate their voting power to a (much smaller) number of representatives who vote on their behalf, e.g., Cardano.[8] Thus we believe that even in their current, unoptimized, and theory-tuned format, our constructions might be of practical interest for an appropriate range of parameters. In fact, we conjecture that the super-linear detection complexity is a limitation of (unconditional) privacy in this non-interactive and no-setup scenario, see Remark 6.

Finally, we show that our voting protocols have optimal behavior with respect to certain properties that are relevant for generalized voting rules. In particular, our (unconditionally) private schemes have the following limitations:

- They do not identify under-voters—i.e., voters that vote less, e.g., for fewer candidates, than what they are supposed to.
- Although we can link (and remove) signatures that violate the quota of k-signatures per party, we cannot identify the (public key) of the party that triggered such a violation.

As we prove in (the following) Theorems 7 and 12, these are not shortcomings of our schemes, but they apply to any non-interactive unconditionally private voting scheme over an ABB without additional setup. Theorem 12 relies on a connection between (k)-LRS and the notion of traceable signatures, which may be of independent interest.

Theorem 1. *(Informally, Theorem 7) For anonymous bulletain board (ABB)-based non-interactive voting protocols, it is impossible to achieve privacy and unvervote identification simultaneously.*

Theorem 2. *(Theorem 12) Traceable ring signatures cannot achieve unconditional anonymity.*

1.2 Related Work

Linkable Ring Signatures. Ring signatures (RS) [RST01] allow a user to sign on behalf of a group, and the identity of the real signer is hidden in the group (i.e., anonymity). Different from group signatures [CH91]

[7] https://governance.algorand.foundation
[8] https://developers.cardano.org/docs/get-started/cardano-cli/governance

where there is a group manager for generating keys and tracing when necessary, there is no group manager in ring signatures, and a signer can generate a signature without the participation or even knowledge of the other users. Furthermore, in most ring signatures schemes [RST01, AOS02, ZK02, DKNS04], anonymity holds unconditionally.

Classical ring signatures are unlinkable, meaning that it is impossible to determine whether two signatures were generated by the same signer. Liu et al. [LWW04] weakened anonymity by introducing linkable ring signatures (LRS), where there is a link algorithm that can publicly determine whether two signatures originate from the same signer (linked) or not (unlinked). With this linkability, LRS has found extensive applications in e-voting [CLW08], e-cash [TW05], ad-hoc network authentication [LWW04], private payment [LRR+19], etc. However, in most existing LRS schemes [LWW04, TW05, LW05, ACST06, BKP20], a pseudoidentity of the signer (see the technical overview below) is contained in the signature for the sake of public linkability, and hence the unconditional anonymity is sacrificed.

There are some works of LRS with unconditional anonymity [LASZ14, BH18, YWM+22, BBG+22, GPS25, Har25]. In [LASZ14], Liu et al. took a new form of key pairs where there are multiple secret keys corresponding to a public key, and proposed the first LRS scheme with unconditional anonymity. Boyen and Haines [BH18] considered the forward security of LRS. Balla et al. [BBG+22] extended Liu et al.'s scheme to the designated-verifier setting. Ye et al. [YWM+22] proposed the LRS scheme over NTRU lattice. However, the schemes in [BH18, YWM+22, BBG+22] all apply the same approach as [LASZ14] and achieve only *weak* linkability, which is not sufficient for most applications, including e-voting.

Two recent and concurrent works [GPS25, Har25] considered unconditionally anonymous LRS with standard linkability. Grontas et al. [GPS25] adopted the same sequential structure as in [LWW04] (and also in our work), and obtained a DH-based LRS scheme. Hara [Har25] proposed the first unconditionally anonymous LRS scheme in the standard model. However, Hara's scheme [Har25] does not support event-oriented signing (see Remark 1) and hence it is one-time-used.

Relation with k-Times Traceable Ring Signatures. The most related work with our k-LRS is the k-times (full) traceable ring signature (k-TRS) proposed by Bultel and Lafourcade in [BL16]. A k-TRS scheme allows a signer to sign on behalf of the ring within k-times, and once a signer exceeds the bound k, all signatures from it will be publicly traced. Moreover, the real identity of the signer is tracked, just as in traceable ring signatures [FS07].

We have seen that the scheme in [BL16] can be viewed as an extension of the "using LRS k times" paradigm (see the technical overview below). By introducing an extra secret/public key pair to trace all signatures from the actual signer, this construction suffers from a shortcoming in that it is stateful, i.e., the signer has to record how many times they have signed. The scheme achieves computational anonymity only due to the traceability (cf. Theorem 12). Moreover, we specify that the scheme in [BL16] does not achieve the traceability as it claimed (though it is still a weaker version of traceability as the weak linkability in [LASZ14]), and we show the concrete attack in Appendix B and a detailed comparison in Appendix G.1.

Relation with k-Times Anonymous Authentication. Teranishi et al. [TFS04] proposed the concept of k-times anonymous authentication (k-TAA) that allows users to authenticate (to some application provider) anonymously, as long as times they are authenticated is within a fixed number k. Both k-TAA and k-LRS aim for the same goal of k-times anonymous authentication. But they have some differences as follows.

- In k-TAA [TFS04, TS06, CHK+06], a group manager generates a global secret/public key pair and issues identification tokens to group members. From this perspective, k-TAA is more akin to the k-extension of a group signature [CH91]. In contrast, k-LRS does not rely on a trusted authority for global key generation, and the signer can flexibly select a group of users to hide their identity when signing.
- To avoid exceeding the limitation of authentication, a user in k-TAA has to track of how many times their credentials have been used, resulting in a stateful scheme. In contrast, our k-LRS is completely stateless.
- In k-LRS, if a user signs k' times and $k' > k$, then all k' signatures from this signer are linked and publicly detectable. However, in k-TAA, the $(k+1)$-th authentication fails, and the verifier does not know anything about the previous k authentications from this user.

– There is no tracing algorithm in k-LRS.

Blockchain and Voting A transaction protocol in a blockchain system can use linkable ring signatures to provide both confidentiality and privacy [SALY17, YSL+20]. There are few works [LJW+19, RAVR21] to implement voting systems using blockchain and ring signatures to be distributed and decentralized.

Receipt-freeness is one of the important requirements for e-voting to ensure a voter cannot prove how they voted to a third party, even if they wish to do so voluntarily. The concept has been widely studied [AOZZ15, BHM08, BT94b, DKR06, DKR10, HS12, HS00, JV06, JCJ10, KT09, KTV11, KTV12, MH96, Oka97, SK95]. Juels et al. [JCJ10] formalized coercion-resistance, which is stronger notion than receipt-freeness, and it becomes one of the necessary conditions for e-voting to prevent several types of attacks such as vote selling or blackmail. Jonker and Pieters [JP10] further studied anonymity and coercion-resistance in e-voting. Langer et al. [LJP10] studied more about security semantics and explored anonymity and verifiability in terms of (un)linkability between a voter and their votes.

Other Methods for Privacy Equipped with (ordinary) LRS for authenticity, there are some other methods to protect the privacy in voting, including homomorphic encryption, shuffling [Gro03, PBD05, Nef01], and multi-party computation (MPC) [Yao82, GMW87], etc. However, they are not as simple as our k-LRS method due to the following reasons.

– For the homomorphic encryption and the shuffling solutions, every voter encrypts their message/vote information before signing and uploading it, and then there is a designated authority that "mixes" the encrypted messages and then releases the secret key so that everyone knows the voting result. However, this requires a trusted authority that is responsible for properly generating a public key or a common reference string and then releasing the corresponding secret key, which deviates from the decentralization principle.
– MPC appears to be a good solution for the privacy problem. However, information-theoretic MPC needs an honest majority of the parties [GMW87] (recall that we pursuit unconditional anonymity), which is exorbitant in our voting application. More importantly, the vote-and-go property (i.e., a voter can finish the voting in one shoot) cannot be achieved since most MPC protocols are interactive.

1.3 Technical Overview

In this subsection, we briefly overview our techniques.

Why the Trivial Construction from LRS to k-LRS Does Not Work? Considering the concept of k-LRS in mind, one might initially think of a simple way to construct k-LRS from k (ordinary) LRS schemes as follows. The signer $S^{(j)}$ ($j \in [n]$ and n is the size of the ring) generates k distinct secret/public key pairs $((sk_1^{(j)}, pk_1^{(j)}), ..., (sk_k^{(j)}, pk_k^{(j)})$ for LRS, and then sets its secret key and public key as

$$sk^{(j)} = (sk_1^{(j)}, ..., sk_k^{(j)}), \text{ and } pk^{(j)} = (pk_1^{(j)}, ..., pk_k^{(j)}).$$

To sign for the i-th time under some event e, the signer $S^{(j)}$ uses their i-th secret key $sk_i^{(j)}$ to sign the message on behalf of the public key ring $(pk_1^{(1)}, ..., pk_k^{(1)}, ..., pk_1^{(n)}, ..., pk_k^{(n)})$. One can obviously sign k times using k secret keys they hold. Furthermore, if a signer signs more than k times, say $(k+1)$ times, then at least two of the signatures among these $(k+1)$ signatures will be linked due to the security of the underlying LRS scheme.

However, this straightforward construction has two drawbacks. First, the signing process is stateful, as a signer must record their state to know how many times they have signed for the same event. Second, and more importantly, the link algorithm can only identify two signatures from the same secret key but not all from the signer, which is essential in many applications, particularly voting systems: Once cheating is detected—i.e., if a signer has signed more than k times—all $k+1$ (or more) signatures from this malicious signer should be deemed invalid and removed from the voting profile. However, in the simple construction

above, it is unclear whether the signer has generated an additional fraudulent signature except for the two linked signatures. We refer to Appendix G.1 for more discussions on the necessity of stateless signatures and k-linkability (linking all $k+1$ signatures).

Toward Unconditional Anonymity. In this work, we consider unconditional anonymity for (k-linkable) ring signatures, i.e., the anonymity does not rely on any hardness assumption. We first review how existing LRS schemes are constructed. Let (sk, pk) be a pair of keys for the LRS scheme. In most existing LRS schemes [LWW04, LW05, TW05, BKP20], a signature contains a pseudoidentity $pid = f_{\mathsf{e}}(sk)$ (a.k.a. linkability tag [TW05]), where $f_{\mathsf{e}}(\cdot)$ is a one-way function indexed by the event label e, and the verification algorithm will check the well-formedness of pid.

- On one hand, two signatures from the same signer (under the same event) will be linked due to the same (or nearly identical[9]) pid.
- On the other hand, the identity of the signer (i.e., sk) is hidden due to the one-wayness of $f_{\mathsf{e}}(\cdot)$, ensuring anonymity.

Take the discrete logarithm (DL) based construction as an example. Let $\mathbb{G}$ be a cyclic group of order q, and g be a generator. A secret/public key pair is in the form of $(sk, pk) = (x, X = g^x) \in \mathbb{Z}_q \times \mathbb{G}$, and $pid = g_{\mathsf{e}}^x \in \mathbb{G}$ is the pseudoidentity leaked in the signature. Here g_{e} is another random generator of $\mathbb{G}$ determined by some public function $g_{\mathsf{e}} = H(\mathsf{e})$.[10] Moreover, according to the decisional Diffie-Hellman (DDH) assumption, given the public key g^x and the event-related parameter g_{e}, the pseudoidentity $pid = g_{\mathsf{e}}^x$ is computationally indistinguishable from a random element in $\mathbb{G}$, thus preserving anonymity.

However, since the information of sk is leaked via $pid = f_{\mathsf{e}}(sk)$, anonymity must rely on the hardness assumption such that pid is pseudorandom, and therefore it is not unconditionally anonymous. It seems that to provide linkability, the anonymity of ring signatures must degrade from unconditional (as in [RST01]) to computational (as in [LWW04]). Designing unconditionally anonymous LRS remained an open problem for a decade since the concept of LRS was proposed.

The breakthrough work by Liu et al. [LASZ14] introduced the first LRS scheme with unconditional anonymity. It follows the pseudoidentity construction to achieve linkability as before. But remarkably, the scheme takes a "multi-to-one" paradigm for key pairs, i.e., for a public key, there are multiple secret keys that correspond to that public key. More precisely, in Liu et al.'s scheme, a secret/public key pair is in the form of

$$(sk, pk) = ((x_1, x_2), X = g_1^{x_1} g_2^{x_2}) \in \mathbb{Z}_q^2 \times \mathbb{G},$$

where g_1, g_2 are two random generators of $\mathbb{G}$. The pseudoidentity in the signature (w.r.t. event e) is $pid = g_{\mathsf{e}}^{x_1} \in \mathbb{G}$, where $g_{\mathsf{e}} = H(\mathsf{e})$. The well-formedness of pid will be checked during verification.

- Unconditional anonymity: The secret key sk consists of two parts (sk_1, sk_2), and only a portion of the secret key (i.e., sk_1) is compromised from the pseudoidentity. Furthermore, as long as the second part sk_2 is uniformly distributed, the public key is uniformly distributed. Therefore, from pid, even an information-theoretic adversary cannot link it to any specific public key or, equivalently, to any signer's identity (as long as the signer does not sign twice).
- Linkability: Given a secret/public key pair, it is infeasible to compute another secret key that corresponds to the public key.

Extension to the k-LRS Setting. Inspired by [LASZ14], we are aiming to design k-LRS with unconditional anonymity. Recall that in k-LRS, a signer can sign at most k times without being traced. We apply the same "multi-to-one" idea as that in [LASZ14], and extend it as follows: a signer generates $k+1$ partial secret keys $sk_1 = x_1, ..., sk_{k+1} = x_{k+1}$, and aggregates them into one public key

$$pk = g_1^{x_1} \cdot g_2^{x_2} \cdots g_{k+1}^{x_{k+1}},$$

[9] In some isogeny-based or lattice-based schemes [BKP20], two signatures are linked if the pseudoidentities contained in them are close enough.

[10] $H(\cdot)$ is modeled as a random oracle in security proofs.

where $g_1, ..., g_{k+1}$ are $k+1$ random generators of $\mathbb{G}$. For the i-th signing w.r.t. event e, the signature contains a pseudoidentity of the form $pid = g_{\mathsf{e}}^{x_i}$. It is easy to see that the leakage of the first k partial secret keys $(x_1, ..., x_k)$ does not threaten anonymity as long as x_{k+1} is uniformly distributed, and thus pk as well. Moreover, according to the $(k+1)$-find representation assumption [Bra93] (which is implied by the DL assumption), given $(x_1, ..., x_{k+1})$ and $pk = \prod_{i\in[k+1]} g_i^{x_i}$, it is infeasible to compute a distinct $(x'_1, ..., x'_{k+1}) \neq (x_1, ..., x_{k+1})$ satisfying $pk = \prod_{i\in[k+1]} g_i^{x'_i}$. Therefore, a malicious signer cannot provide another pseudoidentity in the signature, and linkability is guaranteed.

It seems we are done! However, the above idea has two flaws.

- The signing algorithm is still stateful. A signer has to record a signing state to avoid the case of two signatures containing the same pseudoidentity w.r.t. the same i-th partial secret key.
- If a malicious signed $k+1$ times, of which the first two contain the same pseudoidentity $g_{\mathsf{e}}^{x_1}$, and the following $k-1$ signatures contain $g_{\mathsf{e}}^{x_2}, ..., g_{\mathsf{e}}^{x_k}$, respectively, then how to design the link algorithm to identify all these $k+1$ signatures from many other signatures that contain other different pseudoidentities? Recall that $x_1, ..., x_{k+1}$ are sampled independently.

To address these two issues, we modify the pseudoidentity so that it contains information from all k partial secret keys. Consequently, if $k+1$ pseudoidentities are generated from one signer, some information will overlap, allowing these $k+1$ signatures to be identified. To this goal, we change the function $f_{\mathsf{e}}(sk_1, ..., sk_k)$ to a probabilistic algorithm, and introduce k random coefficients $a_1, ..., a_k \in \mathbb{Z}_q^*$ when signing. Therefore,

$$f_{\mathsf{e}}(sk_1, ..., sk_k; a_1, ..., a_k) := \prod_{i\in[k]} g_{\mathsf{e},i}^{a_i \cdot x_i},$$

where $(g_{\mathsf{e},1}, ..., g_{\mathsf{e},k}) = H(\mathsf{e})$ are k random generators of $\mathbb{G}$. Meanwhile, the final signature includes the coefficients $a_1, ..., a_k$ as well.

The signing algorithm is non-stateful now. For each signing, the signer samples a new group of coefficients $(a_1, ..., a_k)$ independently. From k signatures, which contain k groups of coefficients and k pseudo-identities, one can recover $g_{\mathsf{e},1}^{x_1}, ..., g_{\mathsf{e},k}^{x_k}$ consequently, as long as all these k groups of coefficients form an invertable matrix in $(\mathbb{Z}_q^*)^{k\times k}$, which happens with overwhelming probability. In the following, we call $epk_i = g_{\mathsf{e},i}^{x_i}$ $(i \in [k])$ *event-related public keys*. With such a function $f_{\mathsf{e}}(sk_1, ..., sk_k)$, we can design the link algorithm as follows.

Link: Given $(k+1)$ signatures, Link outputs 1 (linked) if any group of k signatures among them results in the same event-related public keys $epk_1, ..., epk_k$.

The above two flaws have been solved by introducing such coefficients in signing, and the unconditional anonymity holds as before (as long as one signer does not sign more than k times). Additionally, due to the collision resistance of secret key, an adversary cannot find two secret keys that map to the same public key, and hence cannot generate an unlinked pseudoidentity.

Sequential Structure v.s. Parallel Structure. For the security of LRS and k-LRS, (no matter whether it achieves computational anonymity or unconditional anonymity), the following linkability is required.

Linkability (informal). The adversary, who corrupts at most s signers, cannot generate more than ks signatures such that every $(k+1)$ of them are unlinked (for (ordinary) LRS, $k = 1$).

Many works considering unconditional anonymity [LASZ14, BH18, BBG+22, YWM+22] adopt a weaker definition of linkability.

Weak linkability (informal). The adversary, who corrupts one signer, cannot generate more than k (for (ordinary) LRS, $k = 1$) signatures such that these signatures are unlinked.

For most applications of (k-)LRS, especially the application in voting systems, weak linkability is not sufficient at all since it is feasible and motivated for two signers to collude and generate more valid votes

(namely, more unlinked and valid signatures) more than they are allowed to. For Liu et al.'s scheme [LASZ14], we show a concrete attack against the linkability in Appendix A.

We now investigate deeper into Liu et al.'s LRS scheme to understand why it achieves only *weak* linkability. Suppose there are n signers in the ring. Except for the part of pseudoidentity, the core idea for designing an (L)RS scheme is to generate a 1-out-of-n proof [CDS94], showing that the signer holds one secret key corresponding to one public key among total n public keys. The LRS scheme in [LASZ14] takes the well-known Schnorr identification scheme [Sch91]. In more detail, Let $(x, X = g^x)$ be a pair of secret/public keys. In the identification scheme, the first message from the prover (i.e., the signer in LRS) is a commitment $cmt = g^r$ for random $r \in \mathbb{Z}_q$, and given a random challenge $ct \in \mathbb{Z}_q$, the response is computed as $rsp = r + x \cdot ch$. The verification passes if $g^{rsp} = cmt \cdot X^{ch}$. Following the Fiat-Shamir transform [FS86] the identification scheme can be transferred into a signature scheme by setting $ch = H(cmt, \mathsf{msg})$. Moreover, there exists a perfect simulator that outputs a valid transcript (cmt, ch, rsp) which distributes identically as the honestly generated one.

Following the method in [CDS94], Liu et al.'s LRS scheme takes what we called *parallel structure* for the 1-out-of-n proof. The high-level idea is, the signer δ first generates a commitment $cmt^{(\delta)}$, and then simulates other $(n-1)$ members' transcript $\{(cmt^{(j)}, ch^{(j)}, rsp^{(j)})\}$. Fixed n commitments, a random challenge $ch^{(\delta)}$ (for the signer δ's commitment) is fixed according to

$$ch^{(1)} + ch^{(2)} + ... + ch^{(n)} = ch_0,$$

where $ch_0 = H(cmt^{(1)}, ..., cmt^{(n)}, \mathsf{msg})$, and $H(\cdot)$ is a random oracle. With the knowledge of the secret key, δ is able to compute a valid response $rsp^{(\delta)}$. The final signature consists of all n members' transcripts.

Liu et al.'s scheme [LASZ14] takes some optimization on the construction above to reduce the signature size. The final signature is in the form of

$$\sigma = (pid, cmt, ecmt, ch^{(1)}, ..., ch^{(n)}, rsp_1, rsp_2),$$

where $pid = g_{\mathsf{e}}^{x_1^{(\delta)}}$ is the pseudoidentity of the signer δ (w.r.t. event e), cmt is a commitment, and $ecmt$ is an event-related commitment (i.e., it is related to event e). The verification algorithm will check whether the following equations hold.

$$cmt = g_1^{rsp_1} g_2^{rsp_2} \prod_{j\in[n]} (X^{(j)})^{ch^{(j)}}, \tag{1}$$

$$ecmt = g_{\mathsf{e}}^{rsp_1} \prod_{j\in[n]} (pid)^{ch^{(j)}}, \tag{2}$$

$$ch^{(1)} + ch^{(2)} + ... + ch^{(n)} = H(pid, cmt, ecmt, \mathsf{msg}). \tag{3}$$

If signer δ wants to generate a signature, then they has to fix other $(n-1)$ members' challenges before determining cmt, as otherwise it cannot offer a response rsp making Eq. (1) hold, since it does not know the discrete logarithm of $\prod_{j\in[n]\setminus\{\delta\}} (X^{(j)})^{(ch^{(j)})}$. Conditioned on (1), from Eq. (2) we can deduce that pid must be in the form of $g_{\mathsf{e}}^{x_1^{(\delta)}}$.[11] Therefore, a malicious signer δ cannot generate a valid signature in which $pid \neq g_{\mathsf{e}}^{x_1^{(\delta)}}$, and weak linkability holds as a result.

However, if two malicious signers δ and γ collude, then with the knowledge of two secret keys $(x_1^{(\delta)}, x_2^{(\delta)})$ and $(x_1^{(\gamma)}, x_2^{(\gamma)})$, they have more flexibility in choosing $ch^{(\delta)}$ and $ch^{(\gamma)}$, as long as $(ch^{(\delta)} + ch^{(\gamma)})$ is determined due to Eq. (3). Consequently, they are able to generate a valid signature in which $pid \neq g_{\mathsf{e}}^{x_1^{(\delta)}}$ and $pid \neq g_{\mathsf{e}}^{x_1^{(\gamma)}}$, indicating that linkability does not hold in Liu et al.'s scheme.

The key point for the insecurity above is that if the adversary $\mathcal{A}$ controls more than one secret key, then Eq. (3) is not sufficient to restrict the choice of $ch^{(1)}, ..., ch^{(n)}$. With a flexible choice of challenges, rsp_1, rsp_2, and pid are not limited to be in the forms as before, even conditioned on Eq. (1) and (2). To overcome this

[11] Actually, this deduction relies on the 2-FindRep assumption, see Appendix A for more details.

problem, we apply a so-called *sequential structure* for the 1-out-of-n proof to provide a stricter restriction on challenges. Now, an LRS signature is in the form of

$$\sigma = (pid, (cmt^{(1)}, ecmt^{(1)}, ch^{(1)}, rsp_1^{(1)}, rsp_2^{(1)}), ..., (cmt^{(n)}, ecmt^{(n)}, ch^{(n)}, rsp_1^{(n)}, rsp_2^{(n)})).$$

The verification algorithm checks whether the following equations hold for every $j \in [n]$:[12]

$$cmt^{(j)} = g_1^{rsp_1^{(j)}} g_2^{rsp_2^{(j)}} (X^{(j)})^{ch^{(j)}}, \tag{4}$$

$$ecmt^{(j)} = g_{\mathsf{e}}^{rsp_1^{(j)}} (pid)^{ch^{(j)}}, \tag{5}$$

$$ch^{(j)} = H(pid, cmt^{(j-1)}, ecmt^{(j-1)}, \mathsf{msg}). \tag{6}$$

The j-th challenge is determined by the $(j-1)$-th commitment, and all n challenges and n commitments form a ring connected by the hash function. Thanks to such a sequential structure, the choice of challenges are strictly restricted by Eq. (6). Together with (4) and (5), the pseudoidentity must be in the correctness form. Even if the adversary holds more than one secret key, it does not bring any extra advantage when forging an ill-formed *pid*, and the linkability holds as a result.

Taking All Together. We formalize a new primitive called $(k+1)$-aggregatable identification scheme ($(k+1)$-AIS), and then show a generic construction of k-LRS with unconditional anonymity from $(k+1)$-AIS. Informally, a $(k+1)$-AIS scheme has the following properties.

- Aggregation. Given $(k+1)$ secret/public key pairs $(sk_1, pk_1), ..., (sk_{k+1}, pk_{k+1})$, all public keys can be aggregated as a new public key $\widetilde{pk}$. Furthermore, $((sk_1, ..., sk_{k+1}), \widetilde{pk})$ forms a new key pair of the identification scheme.
- Collision resistance of secret key. It is computationally infeasible to find two different secret keys $(sk_1, ..., sk_{k+1})$ and $(sk'_1, ..., sk'_{k+1})$ that correspond to the same $\widetilde{pk}$.
- Uniformity. Fixed any partial secret keys $sk_1, ..., sk_k$, the aggregated public key is uniformly distributed if sk_{k+1} is uniformly distributed.
- Homomorphism. The computation of the validity check for a transcript (cmt, ch, rsp) is homomorphic on some coefficient space $\mathbb{F}$.

Except for the DL-based construction,we give another instantiation of $(k+1)$-AIS from the short integer solution (SIS) assumption. In our construction of k-LRS, a signature is in the form of

$$\sigma = (a_1, ..., a_k, pid, (cmt^{(1)}, ecmt^{(1)}, ch^{(1)}, rsp^{(1)}), ..., (cmt^{(n)}, ecmt^{(n)}, ch^{(n)}, rsp^{(n)})),$$

where $a_1, ..., a_k$ are random coefficients over $\mathbb{F}$, pseudoidentity $pid = f_{\mathsf{e}}(sk_1, ..., sk_k; a_1, ..., a_k)$, and $(cmt^{(j)}, ecmt^{(j)}, ch^{(j)}, rsp^{(j)})$ is a transcript w.r.t. the j-th member in the ring. Moreover, $ch^{(j)} = H(pid, cmt^{(j-1)}, ecmt^{(j-1)}, \mathsf{msg})$ for all $j \in [n]$.

As for security, the k-LRS scheme constructed above has unforgeability, unconditional or computational anonymity, linkability, and non-slanderability. Especially:

- Anonymity. If a signer signs no more than k times in total, then the pseudoidentity contained in the signature leaks only the information of the first k partial secret keys, and hence unconditional anonymity of k-LRS is guaranteed due to the uniformity of $(k+1)$-AIS. Additionally, computational anonymity still holds even if a signer issues more than k signatures in total, as long as no more than k signatures are issued for any single event.
- Linkability. The stronger notation of linkability for k-LRS is achieved, due to the sequential structure of the scheme.

[12] The n-th member in the member also serves as the 0-th member.

Application in Voting. When applying an LRS scheme into a voting system, a vote is actually a message-event-signature tuple. For example, a vote for candidate C in the election event "2025Election" is in the form of

$$(\underbrace{\text{Vote:}C}_{\text{message}}, \underbrace{\text{2025Election}}_{\text{event label}}, \underbrace{\sigma}_{\text{signature}}),$$

where σ is an LRS signature on message "C" under event label "2025Election". Thanks to the security of LRS, the identity of the real voter/signer is hidden from the signature, and double-voting is publicly detectable due to the linkability of two signatures.

From k-LRS we directly obtain a Mode 1 voting scheme, where each voter/signer is allowed to sign up to k times. If a malicious voter votes/signs more than k times, then all votes from them will be detected due to the linkability of signatures, and hence be discarded. For example, in the scenario of score voting where each voter is allowed to give the candidate an (integer) score ranging 0 to k, a vote for candidate C with score s is in the form of

$$((\underbrace{\text{Vote:C}}_{\text{message}}, \underbrace{\text{2025Election-}C}_{\text{event}}, \underbrace{\sigma_1}_{\text{signature}}), ..., (\underbrace{\text{Vote:C}}_{\text{message}}, \underbrace{\text{2025Election-}C}_{\text{event}}, \underbrace{\sigma_s}_{\text{signature}})).$$

All s message-event-signature tuples are separated as long as $s \leq k$. Therefore, privacy is guaranteed, and hence, the scoring distribution is hidden.

Multi-Event Settings. Now suppose there are 5 candidates A, B, C, D, E, and each voter is allowed to select 3 candidates without repetition (e.g., it is not allowed to select A twice). Actually, there are two limitations in this scenario:

1. Each voter can vote at most 3 times totally in this voting event.
2. Each candidate can be selected at most once.

The first restriction can be guaranteed via a 3-LRS scheme. However, how can we simultaneously limit the number of votes cast for the same candidate?

Recall that in k-LRS, the signing bound is related to an event label $\mathtt{e}$, i.e., a signer can sign up tp k times without being linked under a particular $\mathtt{e}$. To deal with the case where there are more than two bounds, a straightforward way is to let the signing be related to different events, which is exactly the idea of a multi-event LRS scheme (see Appendix F for the formal definition).

Let M be the number of event indexes (i.e., how many different bounds). In a $(k_1, ..., k_M)$-MLRS scheme, a signature is generated under M events $\mathtt{e}_1, ..., \mathtt{e}_M$. For any $\pi \in [M]$, $k_\pi + 1$ signatures whose π-th event label are the same will be linked if they came from the same signer. Now the voting problem above can be dealt with a $(3, 1)$-MLRS scheme, where a vote for candidate C is in the form of

$$(\underbrace{\text{Vote:}C}_{\text{message}}, \underbrace{\text{2025Election}}_{\text{event-1}}, \underbrace{C}_{\text{event-2}}, \underbrace{\sigma}_{\text{signature}}).$$

If a malicious voter votes more than 3 times (say, 4 times), then the four signatures will be linked under the first event label "2025Election". Moreover, if the voter votes more than once (say, 2 times) on the same candidate (say, candidate C), then the two signatures will be linked under the second event label "C". In this work, we show a generic construction of $(k_1, ..., k_M)$-MLRS from k-LRS schemes.

Mode 2 Voting: Removing All Property. If there are multiple events (e.g., for score votes, each event is defined with each candidate), Mode 1 voting can only detect ballots from the same event, but cannot detect *all* ballots made from a single signer. Taking the above example, if a malicious voter votes on "A", "B", and "A", respectively, then the first and the third votes from it will be discarded due to the linkability w.r.t. event "A". However, the second vote for "B" remains, which is not desirable in many cases.

We come up with a novel method to solve this problem. The high-level idea is to let a vote for candidate A (i.e., under event label "A") contain more information beyond "A" so that such information can be used

to correlate this vote to other linked votes on another candidate B (i.e., under event label "B"). If the voter obeys the voting rule, all the votes from it are unlinked, and unconditional anonymity still holds.

Suppose there are 5 candidates A, B, C, D, E, and each voter is allowed to select 3 candidates without repetition (e.g., it is not allowed to select A twice). One vote for A is in the form of

$$(\underbrace{\text{Vote:}A}_{\text{message}}, \underbrace{\text{2025Election}, A, B, C, D, E}_{\text{event labels}}, \underbrace{\sigma_{A,1}, ..., \sigma_{A,3}, \sigma_B, \sigma_C, \sigma_D, \sigma_E, \tilde{\sigma}}_{\text{signatures}}),$$

where

1. $\sigma_{A,1}, ..., \sigma_{A,3}$ are three 5-LRS signatures under event "A", and
2. σ_B (resp., σ_C, σ_D, and σ_E) is a 5-LRS signature under event "B" (resp., under events "C", "D", and "E"), and
3. $\tilde{\sigma}$ is a 3-LRS signature under event "2025Election".

The signature $\tilde{\sigma}$ is used to bound the voting times for "2025Election", and the signatures $\sigma_{A,1}, ..., \sigma_{A,3}, \sigma_B, \sigma_C, \sigma_D, \sigma_E$ are used to bound the voting time on each candidate. If a voter behaves honestly, then for a candidate A, there are a total of five signatures at most (three from a vote for A and two from the two other votes on some other candidates). Since we use a 5-LRS scheme, all these signatures will not be linked. However, if the voter votes on A twice, then there are 6 signatures (from the two votes) under the same event label "A", and the two votes will be discarded. Furthermore, a vote for some other candidate will also be detected since it contains one signature under event "A".[13]

For a more general case where each voter gets $k = k_1$ votes and at most k_2 votes on the same candidate is allowed, we can use a k-LRS scheme and a t-LRS scheme. A vote for candidate C is required to contain x t-LRS signatures under event "C", one t-LRS signature under the event of each other candidate, and one k-LRS signature under the event of the whole voting process. Here we set $k_2 x + (k - k_2) \leq t$ and $(k_2 + 1)x > t$. The first inequality is set for functionality/correctness when the voter behaves honestly, and the second inequality is set in case the voter is malicious.

Negative Results. After addressing the issue of overvotes, we further consider whether it is feasible to identify undervoting (i.e., when a malicious voter casts less than k votes and then abstains). Unfortunately, it appears impossible to detect such behavior if we aim to preserve privacy. The formal statement is shown in Sec. 5. We emphasize that this negative result applies only in non-interactive voting scenarios—where a voter casts votes without interacting with others—yet, this is the most practical setting in many applications. An interesting future direction would be to explore efficient interactive voting systems that achieve privacy and unconditional anonymity simultaneously.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter. For $a, b \in \mathbb{Z}$ with $a < b$, define $[a, b] := \{a, a + 1, ..., b\}$. For $c \in \mathbb{Z}^+$ $[c] = [1, c]$. Denote by $x := y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$} \mathcal{S}$ the operation of sampling x uniformly at random from a set $\mathcal{S}$. For a distribution $\mathcal{D}$, denote by $x \leftarrow \mathcal{D}$ the operation of sampling x according to $\mathcal{D}$. For an algorithm $\mathcal{A}$, denote by $y \leftarrow \mathcal{A}(x; r)$ or $y := \mathcal{A}(x; r)$, the operation of running $\mathcal{A}$ with input x and randomness r and assigning the output to y. If the randomness is uniformly sampled, we simply denote as $y \leftarrow \mathcal{A}(x)$. For a ring of users R, we use PK_R to denote the assembly of users' public keys in R. "PPT" is short for probabilistic polynomial-time.

We use bold lower-case letters to denote vectors (e.g., $\mathbf{x}$), and bold upper-case letters to denote matrices (e.g., $\mathbf{A}$). Unless there is a specific description, all vectors are column vectors.

For random variables X and Y defined over $\mathcal{S}$, the min-entropy of X is defined as $\mathbf{H}_\infty(X) := -\log(\max_{s \in \mathcal{S}} \Pr[X = s])$, and the statistical distance between X and Y is defined as $\Delta(X, Y) := 1/2 \sum_{s \in \mathcal{S}} |\Pr[X = s] - \Pr[Y = s]|$.

[13] Actually, we use an extended link algorithm of k-LRS here, which can tell whether k' signatures are signed from one signer for any $k' \geq k + 1$.

Voting Rules. While we briefly introduced *plurality voting* [Cha81, BT94a] in the introduction, we are interested in other voting rules in this work:

- *approval voting* [BF78]: each voter indicates which candidates they "approved" of and the candidate with the most approvals is chosen as the winner;
- *Borda count*: the Borda count is a kind of ranked voting.[14] In the Borda count, each voter assigns points to candidates based on their ranking;
- *score voting*, a.k.a. *range voting*: each voter scores candidates within a specified range (e.g., 0 to 10[15]);
- *multi-voting* [HS00]: each voter can cast up to k_1 votes in total, but no more than k_2 votes for any single candidate, where k_1 and k_2 are arbitrary positive integers.

Note that approval voting is a special case of score voting, and score voting, in turn, is a special case of multi-voting.

3 k-Linkable Ring Signatures

In this section we formally define k-linkable ring signatures (k-LRS).

Definition 1 (k-LRS). *Let $k \in \mathbb{N}^+$. A k-linkable ring signature (k-LRS) scheme consists of the following five algorithms. Namely, $k-\mathsf{LRS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Link})$.*

- $pp \leftarrow \mathsf{Setup}(1^\lambda, k)$. *The setup algorithm takes as input the security parameter 1^λ and the upper bound of unlinked signing times k, and outputs a public parameter pp.*
- $(sk, pk) \leftarrow \mathsf{Gen}(pp)$*: The key generation algorithm takes as input pp, and outputs a pair of secret and public keys (sk, pk). W.l.o.g., we assume pk contains pp.*
- $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$*: The signing algorithm takes as input a group of public keys PK_R, the secret key $sk^{(\delta)}$ of user $\delta \in R$, an event label e and a message msg, and outputs a signature σ.*
- $1/0 \leftarrow \mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma)$*: The verification algorithm Ver takes as input PK_R, e, msg and σ, and outputs a bit indicating the validity of σ.*
  We say a signature σ (w.r.t. event e, message msg and ring R) is valid (resp., invalid), if $\mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma) = 1$ (resp., $\mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma) = 0$).
- $1/0 \leftarrow \mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1})$*: The link algorithm that takes as input an event label e and $(k+1)$ signatures $\sigma_1, ..., \sigma_{k+1}$, and outputs a bit, where 1 indicates that these $(k+1)$ signatures are linked to one actual signer, and 0 indicates unlinked.*

Correctness.

1. *Let $k, N \in \mathbb{N}^+$. For any $R \subseteq [N]$ and $\delta \in R$, any event e and message msg, it holds that*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, k) \\ (sk^{(1)}, pk^{(1)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp) : \mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma) = 0 \\ \sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg}) \end{array}\right] \leq negl(\lambda).$$

2. *Let $k, N \in \mathbb{N}^+$. For any $R_i \subseteq [N]$ and $\delta_i \in R_i$ ($i \in [k+1]$), any event e and messages $\mathsf{msg}_1, ..., \mathsf{msg}_{k+1}$, it holds that*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, k) \\ (sk^{(1)}, pk^{(1)}), ..., (pk^{(N)}, sk^{(N)}) \leftarrow \mathsf{Gen}(pp) \quad : \mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1}) = 1 \\ For\ i \in [k+1] : \sigma_i \leftarrow \mathsf{Sign}(PK_{R_i}, sk^{(\delta_i)}, \mathsf{e}, \mathsf{msg}_i) \end{array}\right] = 1, \text{ if } \delta_1 = ... = \delta_{k+1};$$

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, k) \\ (sk^{(1)}, pk^{(1)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp) \quad : \mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1}) = 1 \\ For\ i \in [k+1] : \sigma_i \leftarrow \mathsf{Sign}(PK_{R_i}, sk^{(\delta_i)}, \mathsf{e}, \mathsf{msg}_i) \end{array}\right] \leq negl(\lambda), \text{ otherwise.}$$

[14] Ranked voting depends only on which of two candidates is preferred by a voter.
[15] Range voting was originally proposed with scores of 0 or 1, and it can be a range of integers without loss of generality.

If $k = 1$, then k-LRS degrades to (ordinary) LRS.

Remark 1 (On the universality of the link algorithm). In the definition above the link algorithm Link works only for two signatures under the same event label e (a.k.a. prefix in [BEHM22, HC24]). In some previous works, Link is defined only in the case two signatures are on behalf of the same ring (e.g., [LWW04, LW05]). Here we take a more generic definition as that in [TWC+04, LASZ14] by introducing the event label as an extra input in the signing. The definition in previous works is covered by ours when $\mathsf{e} = PK_R$.

Remark 2 (Extended link algorithm). It is easy to modify Link to an extended algorithm that predicates whether k' signatures are linked for any $k' \geq k+1$. In the application of voting systems, this extended link algorithm will be used to identify all votes from a dishonest user, if they voted more than k times.

As for security, we require unforgeability, (unconditional or computational) anonymity, linkability, and non-slanderability for k-LRS. Let $N \in \mathbb{N}^+$ is the total number of users in the system.

Unforgeability. The adversary cannot forge a signature for a new message on behalf of the ring R in which it is not included.

(Unconditional) anonymity. Given at most k ring signatures, the unconditional or computational adversary cannot detect which signer actually signed them.

Linkability. The adversary who corrupts at most s signers cannot output $(ks+1)$ different signatures w.r.t. the same event such that, every $(k+1)$ of them are unlinked.

Non-slanderability. After seeing k signatures w.r.t. the same event from one honest signer, the adversary cannot generate a valid forgery that makes these $k+1$ signatures linked.

More formally, we define the security properties via the following experiments. Here $N \in \mathbb{N}^+$ denotes the total number of users in the system.

Definition 2 (Unforgeability of k-LRS). *Let $k-\mathsf{LRS}$ be a k-LRS scheme. We say $k-\mathsf{LRS}$ has unforgeability, if $\mathsf{Adv}^{unforg}_{\mathcal{A},k}(\lambda) := \Pr[\mathsf{Exp}^{unforg}_{\mathcal{A},k}(\lambda) = 1] \leq negl(\lambda)$ holds for all PPT adversary $\mathcal{A}$, where the experiment $\mathsf{Exp}^{unforg}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 1.*

$\mathsf{Exp}^{unforg}_{\mathcal{A},k}(\lambda)$:	$\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, k)$	If $R \notin [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	$\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}, \mathsf{msg}, \sigma)\}$
$(R^*, \mathsf{e}^*, \mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot,\cdot,\cdot,\cdot),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	Return σ
If $R^* \subseteq [N] \ \wedge \ R^* \cap \mathcal{S}_{corr} = \varnothing \ \wedge \ (R^*, \mathsf{e}^*, \mathsf{msg}^*, \cdot) \notin \mathcal{S}$ $\wedge \ \mathsf{Ver}(PK_{R^*}, \mathsf{e}^*, \mathsf{msg}^*, \sigma^*) = 1$: output 1	$\textsc{Corr}(j)$:
Otherwise: output 0	$\mathcal{S}_{corr} := \{j\}$
	Return $sk^{(j)}$

Fig. 1. The unforgeability experiment of k-LRS.

Definition 3 ((Unconditional) anonymity of k-LRS). *Let $k-\mathsf{LRS}$ be a k-LRS scheme. We say $k-\mathsf{LRS}$ has anonymity, if $\mathsf{Adv}^{anony}_{\mathcal{A},k}(\lambda) := |\Pr[\mathsf{Exp}^{anony}_{\mathcal{A},k}(\lambda) = 1] - 1/2| \leq negl(\lambda)$ holds for all PPT adversary $\mathcal{A}$, where the experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 2.*

If $\mathsf{Adv}^{anony}_{\mathcal{A},k}(\lambda) \leq negl(\lambda)$ even for all powerful adversary $\mathcal{A}$, then we say $k-\mathsf{LRS}$ has unconditional anonymity.

$\mathsf{Exp}^{anony}_{\mathcal{A},k}(\lambda)$:	$\underline{\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})}$:
$b \xleftarrow{\$} \{0,1\}$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
$pp \leftarrow \mathsf{Setup}(1^\lambda, k)$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
$\mathcal{S}_{pair} := \varnothing$	$++\,\mathcal{L}[\delta, \mathsf{e}]$, $++\,\mathcal{T}[\delta]$
For $j \in [N]$:	Return σ
$(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	
$\mathcal{T}[j] := 0$	$\underline{\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}, \mathsf{msg}_0, \mathsf{msg}_1)}$:
For all δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$	If $R \nsubseteq [N] \ \vee \ \delta_0 \notin R \ \vee \ \delta_1 \notin R$: return $\perp$
$b' \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot,\cdot,\cdot,\cdot),\textsc{Chall}(b,\cdot,\cdot,\cdot,\cdot,\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	If $\mathcal{T}[\delta_0] \geq k \ \vee \ \mathcal{T}[\delta_1] \geq k$: return $\perp$
	If $\mathcal{L}[\delta_0, \mathsf{e}] \geq k \ \vee \ \mathcal{L}[\delta_1, \mathsf{e}] \geq k$: return $\perp$
If $b = b'$: output 1	$++\,\mathcal{L}[\delta_0, \mathsf{e}]$; $++\,\mathcal{L}[\delta_1, \mathsf{e}]$
Otherwise: output 0	$\sigma_0 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_{0\oplus b})}, \mathsf{e}, \mathsf{msg}_0)$
	$\sigma_1 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_{1\oplus b})}, \mathsf{e}, \mathsf{msg}_1)$
	Return (σ_0, σ_1)

Fig. 2. The (unconditional) anonymity experiment of k-LRS, where the highlighted codes are only shown in the unconditional anonymity experiment.

Remark 3 (On the definition of (unconditional) anonymity). In our definition above, even an all-powerful adversary $\mathcal{A}$ is granted access to the signing oracle. This seems redundant at first glace, since such adversary $\mathcal{A}$ can extract a secret key from a public key and simulate the signing oracle by itself. However, we emphasize that a single public key may correspond to multiple secret keys, as is the case in this work. Consequently, access to the signing oracle allows $\mathcal{A}$ to obtain information about the specific secret key held by the signer, which may help to break the anonymity.

A drawback of unconditional anonymity is that, during the security experiment, a secret key can be used to sign at most k times anonymously, even for different events. This is due to the fact that each signature must embed some information about the secret key sk to ensure linkability. However, the entropy of sk is inheriently limited, as its length is polynomial in the security parameter. Consequently, an unbounded adversary could extract sufficient information about sk from a large number of signatures, ultimately enabling the detection of another signature issued by the same sk.

Definition 4 (Linkability of k-LRS). *Let $k-\mathsf{LRS}$ be a k-LRS scheme. We say $k-\mathsf{LRS}$ has unforgeability, if $\mathsf{Adv}^{link}_{\mathcal{A},k}(\lambda) := \Pr[\mathsf{Exp}^{link}_{\mathcal{A},k}(\lambda) = 1] \leq negl(\lambda)$ holds for all PPT adversary $\mathcal{A}$, where the experiment $\mathsf{Exp}^{link}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 3.*

$\mathsf{Exp}^{link}_{\mathcal{A},k}(\lambda)$:	$\underline{\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})}$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, k)$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	If $\mathcal{L}[\delta, \mathsf{e}] \geq k$: return $\perp$
For all δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$	$++\,\mathcal{L}[\delta, \mathsf{e}]$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
$(\mathsf{e}^*, \{(R_i^*, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[\ell]}) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot,\cdot,\cdot,\cdot),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	$\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}, \mathsf{msg}, \sigma)\}$
	Return σ
If $\big(\forall i \in [\ell] : (R_i^*, \mathsf{e}^*, \mathsf{msg}_i^*, \cdot) \notin \mathcal{S} \wedge \mathsf{Ver}(PK_{R_i^*}, \mathsf{e}^*, \mathsf{msg}_i^*, \sigma_i^*) = 1\big)$	
$\wedge \ (\forall G \subseteq [\ell], \lvert G\rvert = k+1 : \mathsf{Link}(\mathsf{e}^*, \{\sigma_i^*\}_{i\in G}) = 0)$	$\underline{\textsc{Corr}(j)}$:
$\wedge \ \lvert\mathcal{S}_{corr}\rvert \leq \lfloor(\ell-1)/k\rfloor$: output 1	$\mathcal{S}_{corr} := \mathcal{S}_{corr} \cup \{j\}$
Otherwise: output 0	Return $sk^{(j)}$

Fig. 3. The linkability experiment of k-LRS.

Remark 4 (Weak linkability of k-LRS). In the definition above, if the adversary $\mathcal{A}$ can corrupt at most one user, then linkability degrades to weak linkability (as defined in [LASZ14] in case $k = 1$). Weak definition

is not sufficient for most applications, especially in the voting systems, since it is feasible and motivated for two signers to collude to generate more unlinked signatures than they are allowed to, and hence control more votes and break the fairness of voting.

Definition 5 (Non-slanderability of k-LRS). *Let $k-\mathsf{LRS}$ be a k-LRS scheme. We say $k-\mathsf{LRS}$ has unforgeability, if $\mathsf{Adv}^{non\text{-}sl}_{\mathcal{A},k}(\lambda) := \Pr[\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k}(\lambda) = 1] \leq negl(\lambda)$ holds for all PPT adversary $\mathcal{A}$, where the experiment $\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 4.*

$\underline{\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k}(\lambda)}$:	$\underline{\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})}$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, k)$	If $R \notin [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	If $\mathcal{L}[\delta, \mathsf{e}] \geq k$: return $\perp$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	$++\,\mathcal{L}[\delta, \mathsf{e}]$
For all δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
$(\mathsf{e}^*, \delta^*, \{(R_i^*, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[k+1]})$	$\mathcal{S} := \mathcal{S} \cup \{(R, \delta, \mathsf{e}, \mathsf{msg}, \sigma)\}$
$\leftarrow \mathcal{A}^{\textsc{Sign}(\cdot,\cdot,\cdot,\cdot),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	Return σ
If $\{(R_i^*, \delta^*, \mathsf{e}^*, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[k]} \subseteq \mathcal{S}$	$\underline{\textsc{Corr}(j)}$:
$\wedge\ (R_{k+1}^*, \delta^*, \mathsf{e}^*, \mathsf{msg}_{k+1}^*, \cdot) \notin \mathcal{S}$	$\mathcal{S}_{corr} := \{j\}$
$\wedge\ \mathsf{Ver}(PK_{R_{k+1}^*}, \mathsf{e}^*, \mathsf{msg}_{k+1}^*, \sigma_{k+1}^*) = 1$	Return $sk^{(j)}$
$\wedge\ \mathsf{Link}(\mathsf{e}^*, \{\sigma_i^*\}_{i\in[k+1]}) = 1 \ \wedge \ \delta^* \notin \mathcal{S}_{corr}$: output 1	
Otherwise: output 0	

Fig. 4. The non-slanderability experiment of k-LRS.

4 Applications in Voting Systems

Now we show how k-LRS can be used to build anonymous non-interactive voting.

4.1 Non-Interactive Voting over Anonymous BB

In this paper, we focus more on multi-party protocols between voters. We assume only eligible voters are registered correctly in a given voting event, so we ignore registrars and corresponding pre-computations. Our voting protocols are purely multi-party protocols between voters only and without tallying authority or verification phase since the protocols can be proven to be correct and secure without any of those.

We assume public key infrastructure (PKI) and anonymous bulletin board (ABB) model, i.e., we assume there is a PKI where everyone has access to the public keys of others, and there is an ABB on which everyone can write privately, and there is no direct communication channel between any two parties except ABB. We recall the definition of ABB in Appendix H. For simplicity, we assume there is no network delay (the consideration of network delays is vital, but orthogonal to the goals of this work).

Definition 6 (Voting specification). *Let $\mathcal{V}, \mathcal{E}, \mathcal{D}, \mathsf{S}$ denote the set of voters, the vote space, the decision space, and the tally function, respectively. A* voting specification *$\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{D}, \mathsf{S})$ consists of the following components:*

- *Voters: The voters are (represented by IDs) from the set $\mathcal{V}$ (throughout this paper, $\mathcal{V} = [n]$). The voters cast their votes and receive their outcome at the end.*
- *Vote: A vote $\mathsf{vote}^{(u)} \in \mathcal{E}$ of a voter $u \in [n]$ is a vector defined over a domain $\mathcal{E}$.*
- *Voting rule: the voting rule specifies how the decision is computed from the votes cast by the voters. The voting rule is represented by a deterministic function $\mathsf{S}(\cdot) : \mathcal{E}^n \to \mathcal{D}$ that takes a tuple of vectors from $\mathcal{E}$ (on from each voter) and outputs a decision from a decision space $\mathcal{D}$.*

– *Profile: A profile* $P = (\mathsf{vote}^{(1)}, ..., \mathsf{vote}^{(n)}) \in \mathcal{E}^n$ *is a vector in* $\mathcal{E}^n$*. It represents the ordered sequence of all votes cast by the voters.*

A dishonest voter may attempt to cast a vote outside the domain $\mathcal{E}$. Ideally such votes should be either ignored, or mapped to a (set of) default value(s) according to the scoring rule (e.g., treating them as casting a score of 0 in the case of a scoring vote).

Example 1. In plurality voting, the vote domain $\mathcal{E}$ is defined over $\mathcal{E} \subset \{0,1\}^q$ such that $\mathsf{vote} \in \mathcal{E} \Leftrightarrow hw(\mathsf{vote}) = 1$ (the hamming weight of vote) if each digit denotes each candidate among q candidates and voters are eligible to cast their vote for only one candidate. If $q = 3$ and $\mathsf{vote} = 010$, then the vote is for the second candidate. ◁

In this paper, we primarily focus on more complex voting rules such as the Borda count, score voting, and multi-voting. We define a specific specification to describe these rules:

Definition 7 (k-voting specification). *A* k-voting specification $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{D}, \mathsf{S})$ *is a voting specification such that* $\mathcal{E}$ *can be decomposed into* k *components, i.e.,* $\mathcal{E} \subseteq (\mathcal{E}')^k$ *for a domain* $\mathcal{E}'$*.*

Example 2. Note that the voting specification in the previous example can also be a q-voting specification. For other examples, in score voting, $\mathcal{E}$ can be $\mathcal{E} = [0,k] \subseteq [0,1]^k$ if there is only one candidate and one can give a score from 0 to k. In the same setting but when there are q candidates, $\mathcal{E} = [0,1]^{(k \cdot q)}$ so that it can be viewed as $\mathcal{E} = [0,k]^q$. In the Borda count where there are $k = q$ candidates, it can be $\mathcal{E} = S_k \subseteq [k]^k$ where S_k is the finite symmetric group defined over $[k]$. ◁

Definition 8 (ABB-based (non-interactive) voting protocols). *Let* $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ *be a voting specification, and* Π *be an* n*-party protocol defined over* Γ*, where* $\mathcal{V}$ *are the participants with their inputs in* $\mathcal{E}$*, and outputs a decision value contained in* $\mathcal{D}$*. Voters in* $\mathcal{V}$ *can call* S *as a subroutine of* Γ *during the running.*

We call Π *an ABB-based voting protocol if it satisfies the following properties.*

1. *Voters have read/write access to an anonymous bulletin board (ABB). Any pair of two voters in* $\mathcal{V}$ *cannot communicate with each other except by writing on and reading from ABB.*
2. *The output of* Π *for each voter is computed via a public function that takes as input the final state of ABB and outputs the decision (tally) from the decision space* $\mathcal{D}$*.*

We say that an ABB-based voting protocol is non-interactive *if the voters only write once on the ABB in the protocol.*

ABB-based non-interactive voting offers several advantages, including the *vote-and-go* property, where a voter can complete the voting process in a single step. In this paper, we primarily focus on this class of voting protocols.

Note that a vote in the protocol can be an arbitrary string of any length, potentially falling outside the domain $\mathcal{E}$ if we consider a malicious adversary so that a voter does not follow the instruction. The input of voters in the protocol is intended to be a profile $P \in \mathcal{E}^n$, but it might be not for the above case. However, when the context is clear, by abuse of notation, we will keep using a term "profile" $P = (\mathsf{vote}^{(1)}, ..., \mathsf{vote}^{(n)})$ to denote an input of a protocol even if it could be not an element of $\mathcal{E}^n$. Additionally, by definition, honest parties only receive messages through the ABB. However, we emphasize that corrupted parties can coordinate their views and actions in an arbitrary manner.

Definition 9 (Correctness of voting protocols). *We say a voting protocol* Π *is correct with respect to a voting specification* $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$*, if the following properties are satisfied except with negligible probability.*

1. *(Completeness) For any profile* $P \in \mathcal{E}^n$*, in an honest execution of* Π *with inputs* P*—i.e., if the* n *voters honestly run* Π *with initial inputs* P*—the output of* Π *(for every voter) equals* $\mathsf{S}(P)$*.*

2. *(Robustness) There exists an extractor* E *that maps a malicious PPT adversary* $\mathcal{A}$ *attacking* Π *(who is able to corrupt any number of voters and have them behave arbitrarily during the protocol) to an augmented semi-honest PPT adversary* $\mathcal{A}' = \mathsf{E}(\mathcal{A})$ *for* Π *(i.e., an adversary who is able to corrupt the same number of voters and modify their inputs but run the protocol honestly) such that the output of any honest voter in an execution of* Π *against adversary* $\mathcal{A}$ *is distributed identically to the output of the same honest voter with the same input in an execution of* Π *against adversary* $\mathcal{A}'$*.*

Remark 5. The first property (completeness) in Def. 9 ensures that the protocol Π will return the correct result according to the rule S, while the second property (robustness) asserts that the adversary cannot influence the result except by altering the inputs of corrupted users it controls, i.e., the adversary cannot "disrupt" the execution of Π.

Additionally, we note in passing that the two properties in the above definition are for different experiments. Completeness is for an honest execution, and hence the probability of error is taken over just the choice of randomness for n voters. Robustness considers the extractor and adversary, hence is the probability is taken over the choices of the randomness of uncorrupted voters, $\mathcal{A}$, and E.

Example 3. Let $n = 2$ and the voters be Arthur and Bertha, corresponding to 1 and 2, respectively. Suppose there are five candidates, Alice, Bob, Carol, Dick, and Eve, and voters are supposed to score each candidate from 0 to 9, i.e., $\mathcal{E} = [0, 9]^5$.

$$P = (\mathsf{vote}^{(1)}, \mathsf{vote}^{(2)})^T = \begin{bmatrix} 1\,2\,3\,4\,5 \\ 9\,0\,9\,0\,9 \end{bmatrix},$$

states that Arthur assigns 1 to Alice, 2 to Bob, 3 to Carol, 4 to Dick, and 5 to Eve, and Bertha assigns 9 to Alice, 0 to Bob, 9 to Carol, 0 to Dick, and 9 to Eve.

Assume that the voting is specified such that the decision (tally) ranks the candidates in descending order based on the sum of their scores. Then, given P as before, the total score for Alice is 10, that of Bob is 2, and so on. Therefore,

$$\mathsf{S}(P) = (\text{Eve, Carol, Alice, Dick, Bob}).$$

Consider a voting protocol Π for the above voting specification. If Π is correct, then the output of Π (for honest users) should be $\mathsf{S}(P)$. In this paper we consider ABB-based non-interactive voting protocols. Therefore, there is a public function $\mathsf{S}_{ABB}(\cdot)$ such that with high probability $\mathsf{S}_{ABB}(P_{ABB}) = \mathsf{S}(P)$. Here P_{ABB} denotes the final state of ABB after a running of protocol Π with initial input P. For example, given a profile P as above, if Π is designed so that every voter casts their votes along with a (ring) signature on ABB (as the case in this paper), then the state of ABB might be

$$P_{ABB} = \begin{bmatrix} (9, 0, 9, 0, 9, \sigma^{(2)}) \\ (1, 2, 3, 4, 5, \sigma^{(1)}) \end{bmatrix}$$

up to a permutation for the rows, where $\sigma^{(1)}$ (resp., $\sigma^{(2)}$) denotes a signature from voter 1 (resp., voter 2).

Now, consider the scenario where Bertha is acting maliciously. In addition to her own vote $(9, 0, 9, 0, 9)$, she also intends to cast an additional vote $(8, 0, 8, 0, 8)$. Specifically, Bertha takes two inputs: $\mathsf{vote}^{(2a)} = (9, 0, 9, 0, 9)$ and $\mathsf{vote}^{(2b)} = (8, 0, 8, 0, 8)$, and attempts to impersonate two voters. Following the above example, after a permutation by ABB, a state P_{ABB} could be

$$P_{ABB} = \begin{bmatrix} (9, 0, 9, 0, 9, \sigma^{(2a)}) \\ (1, 2, 3, 4, 5, \sigma^{(1)}) \\ (8, 0, 8, 0, 8, \sigma^{(2b)}) \end{bmatrix}.$$

Due to the correctness of Π, there must be a machanism to identify and discard the malicious inputs $(9, 0, 9, 0, 9)$ and $(8, 0, 8, 0, 8)$ while retaining the valid input $(1, 2, 3, 4, 5)$ left, resulting in a tally of (Eve, Dick, Carol, Bob, Alice). This can be interpreted as the existence of an extractor E that efficitively maps the malicious adversary Bertha into a passive adversary holding an input $\mathsf{vote}^{(2)} = (0, 0, 0, 0, 0)$. The correctness is ensured by the linkability of k-LRS, as $\sigma^{(2a)}$ and $\sigma^{(2b)}$ are linked, having been signed by the same voter Bertha. ◁

Definition 10 (Anonymity of voting protocols). *Let $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a voting specification and Π be a protocol for it. We say Π has (unconditional) anonymity, if for any $i, j \in [n]$, any PPT (information-theoretic) adversary $\mathcal{A}$ who corrupts a group of voters except i, j, any profiles $P = (\mathsf{vote}^{(1)}, \ldots, \mathsf{vote}^{(n)})$ and $P_{[i,j]}$, it holds that*

$$\left|\Pr\left[\mathcal{A}^{\Pi^P} \to 1\right] - \Pr\left[\mathcal{A}^{\Pi^{P_{[i,j]}}} \to 1\right]\right| \leq negl(\lambda),$$

where λ is the security parameter, and

$$P_{[i,j]} = (\mathsf{vote}^{(1)'}, ..., \mathsf{vote}^{(n)'}) \text{ s.t. } \begin{cases} \mathsf{vote}^{(i)'} = \mathsf{vote}^{(j)}, \\ \mathsf{vote}^{(j)'} = \mathsf{vote}^{(i)}, \\ \mathsf{vote}^{(\iota)'} = \mathsf{vote}^{(\iota)}, \ \textit{otherwise.} \end{cases}$$

Example 4. Following the previous example, let

$$P = (\mathsf{vote}^{(1)}, \mathsf{vote}^{(2)})^T = \begin{bmatrix} 1\,2\,3\,4\,5 \\ 9\,0\,9\,0\,9 \end{bmatrix}.$$

Then P indicates that Arthur (the first voter) assigns 1 to Alice, 2 to Bob, 3 to Carol, 4 to Dick, and 5 to Eve, and Bertha (the second voter) assigns 9 to Alice, 0 to Bob, 9 to Carol, 0 to Dick, and 9 to Eve. Now, let's consider

$$P_{[1,2]} = \begin{bmatrix} 9\,0\,9\,0\,9 \\ 1\,2\,3\,4\,5 \end{bmatrix}.$$

Then $P_{[1,2]}$ indicates that Arthur gives 9 to Alice, 0 to Bob, 9 to Carol, 0 to Dick, and 9 to Eve, and Bertha gives 1 to Alice, 2 to Bob, 3 to Carol, 4 to Dick, and 5 to Eve. Anonymity ensures that the running with P and the running with $P_{[1,2]}$ are indistinguishable from the attacker. This captures the unlinkability between voters and their votes. ◁

Definition 11 (Privacy of voting protocols). *Let $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a k-voting specification (i.e., $\mathsf{vote}^{(j)} = (\mathsf{vote}_1^{(j)}, ..., \mathsf{vote}_k^{(j)})$ for $j \in [n]$), and Π be a protocol for Γ. We say Π has (unconditional) privacy, if for any $(i, j) \in [n]^2$, any $G \subseteq [k]$, any PPT (information-theoretic) adversary $\mathcal{A}$ who corrupt a group of voters except i, j, any profiles $P = (\mathsf{vote}^{(1)}, \ldots, \mathsf{vote}^{(n)})$ and $P_{[i,j,G]}$, it holds that*

$$\left|\Pr\left[\mathcal{A}^{\Pi^P} \to 1\right] - \Pr\left[\mathcal{A}^{\Pi^{P_{[i,j,G]}}} \to 1\right]\right| \leq negl(\lambda),$$

where λ is the security parameter, and

$$P_{[i,j,G]} = \begin{pmatrix} (\mathsf{vote}_1^{(1)'}, ..., \mathsf{vote}_k^{(1)'}) \\ \cdots \\ (\mathsf{vote}_1^{(n)'}, ..., \mathsf{vote}_k^{(n)'}) \end{pmatrix} \text{ s.t. } \begin{cases} \mathsf{vote}_\ell^{(i)'} = \mathsf{vote}_\ell^{(j)}, \ \ell \in [G] \\ \mathsf{vote}_\ell^{(j)'} = \mathsf{vote}_\ell^{(i)}, \ \ell \in [G] \\ \mathsf{vote}_\ell^{(\iota)'} = \mathsf{vote}_\ell^{(\iota)}, \ \textit{otherwise.} \end{cases}$$

Example 5. Following the previous example, let

$$P = (\mathsf{vote}^{(1)}, \mathsf{vote}^{(2)})^T = \begin{bmatrix} 1\,2\,3\,4\,5 \\ 9\,0\,9\,0\,9 \end{bmatrix}.$$

Then P indicates that Arthur gives 1 to Alice, 2 to Bob, 3 to Carol, 4 to Dick, and 5 to Eve, and Bertha gives 9 to Alice, 0 to Bob, 9 to Carol, 0 to Dick, and 9 to Eve. Now consider

$$P_{[1,2,G=\{1,4,5\}]} = \begin{bmatrix} 9\,2\,3\,0\,9 \\ 1\,0\,9\,4\,5 \end{bmatrix}.$$

Then $P_{[1,2,G]}$ indicates that Arthur gives 9 to Alice, 2 to Bob, 3 to Carol, 4 to Dick, and 9 to Eve, and Bertha gives 1 to Alice, 0 to Bob, 9 to Carol, 4 to Dick, and 5 to Eve. Privacy ensures that the execution with P and the execution with $P_{[1,2,G]}$ are indistinguishable to the attacker. In other words, not only are identities of Arthur and Bertha kept confidential, but it is also private whether the voter who assigns 9 to Alice assigns 2 or 0 to Bob. ◁

4.2 Instantiating Different Voting Rules

This section describes applications of our k-LRS in various voting scenarios. Let $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a voting specification we will implement together with a fixed number $\mathcal{V} = [n]$. Every participant in our protocol obtains an entire profile P (which may contain arbitrary messages from malicious users so that P may not be in $\mathcal{E}^n$) just by looking ABB together with an identified overvote set $P' \subset P$. With S, one can get a desired result by running $\mathsf{S}(P \setminus P')$. This part is trivial, so we will focus on the steps of getting P', i.e., we will define $\mathcal{E}$ precisely, but we will not specify $\mathcal{D}$ and S since literally they can be anything depending on users' choice. In this paper, we propose two voting modes to ensure privacy.

Mode 1 (Score voting, a.k.a. range voting). Each voter can vote their preference as an integer score (where there is only a single candidate).

As the first simple but powerful use case, we can immediately obtain Mode 1 voting via k-LRS. This mode allows to cast a vote as an integer score score from 0 to k with *privacy*, which has never been achieved in e-voting literature. Note that this rule is equivalent to allowing each user to cast or abstain from voting up to k times, where each vote implies giving an additional score of 1. Thanks to k-LRS, everyone can easily verify if one casts their vote more than k times, namely, if one does *overvote*, which will be formally defined later. This mode can be extended to cover any number of voting candidates by simply running the mode in parallel per candidate. For this mode, the specification is required to be $\mathcal{E} = [0, k] \in [0, 1]^k$.

Mode 2 (Multi-voting). Each voter is eligible to cast their votes up to k_1 times with a restriction that at most k_2 votes are allowed to each candidate. Once a voter breaks this rule, then all their ballots will be discarded. Suppose there are q voting candidates. One can obtain an enhanced version of the parallel running of Mode 1 by letting $k_1 = q \cdot k_2$. While the parallel execution of Mode 1 can only verify overvotes for a single candidate (thus preserving the anonymity of the dishonest voter's ballots for other candidates), Mode 2 can verify overvotes across all candidates. For this mode, we restrict $\mathcal{E} \subseteq [0, q]^{k_1}$ such that $\mathsf{vote} = (\mathsf{vote}_1, \ldots, \mathsf{vote}_{k_1}) \in \mathcal{E}$ if and only if $\{\mathsf{vote}_i\}_{i \in I} = \{0\}$ or $|\{\mathsf{vote}_i\}_{i \in I}| > 1$ for all $I \subseteq [1, k_1]$ such that $|I| > k_2$. In other words, $\mathsf{vote}_i = 0$ implies that one abstains from casting their i-th vote, allowing vote_i to be zero more than k_2 times among all choices of i. Meanwhile, $\mathsf{vote}_i = j \in [1, q]$ indicates that the (partial) vote is for the j-th candidate, with the number of repetitions not exceeding the threshold k_2.

We emphasize that the mode can also implement the Borda count by simply letting $k_2 = 1$ and distinguishing vote_i by position to denote i-th preference for each i.

In the remainder of this section, we will provide formal protocol descriptions and proof of the correctness and privacy of the modes. However, before doing so, we first discuss some details below.

Registering/precomputation. In practice, eligibility to vote also can be an issue. There should be a publicly available PKI, and only eligible voters should be able to register their public key to the PKI. We omit the details on this since our primary concern is MPC protocol among (eligible) voters, but we emphasize that the proof of eligibility can be simply done by submitting a certificate from registrars.

Random strings. In the following voting systems, every vote is a (set of) valid message-event-signature tuple(s), and the message contains the voting preference and a random string $str \in \{0, 1\}^\lambda$ (just like the serial number on banknotes and checks) where λ is the security parameter of our signature scheme. When counting the result, if two votes or two message-event-signature tuples have the same random string, then one will be discarded. This results in a negligible correctness error in the voting systems.

Our SIS-based k-LRS and $(k_1, ..., k_M)$-MLRS schemes (Appendix D.1) do not achieve *strong* unforgeability, i.e., after seeing a signature σ on some message msg and event e, it is possible to generate another valid signature σ' on the same msg and e. If there is no random string in the signed message, then after seeing several votes (message-event-signature tuples) to a candidate, the adversary can generate more signatures on the same message and event, increase the number of votes for that candidate and hence break the correctness of voting.

Limitations and allowing abstention. Note that if undervoting is possible, meaning the voting specification requires casting at least one vote (or a positive score), any ABB-based non-interactive voting protocol

cannot simultaneously achieve correctness, privacy, and overvote identification, as established by the impossibility result (Sec. 5). However, if the specification permits abstention, undervoting is effectively eliminated, allowing all of these properties to be achieved. Another important note is about coercion-resistant security [JCJ10]. Since we consider ABB-based non-interactive voting protocol, which is a protocol without tallying authorities, it is impossible to guarantee coercion-resistant security. This can be easily seen from our assumption since we assume no trusted party at all. However, it does not limit the potential of our k-LRS schemes; this is a limitation of the security assumption. To achieve coercion resistance, we need more complicated cryptographic tools and more assumptions, e.g., a trusted tally server that provides dummy ballots [LQT20]. On the other hand, our k-LRS schemes can be implemented in such voting systems, which could be an interesting future work.

4.3 Mode 1 Voting from k-LRS

We design a mode 1 voting protocol from our k-LRS scheme. Thanks to the k-LRS scheme, the protocol is extremely simple. In a nutshell, each voter 1) registers their public key to PKI, 2) generates vote via k-LRS scheme and uploads it to ABB, and 3) reads ABB, discards overvotes via Link algorithm, and computes a voting rule. Precisely, each voter $u \in [n]$ runs the following protocol:

Preparing phase. u generates a key pair (sk_u, pk_u) of a k-LRS scheme and registers pk_u to PKI.

Voting phase. Let $0 \leq s \leq k$ be the score intended by u. The voter u does the following for each $i \in [s]$:

1. Sample random strings $str_i \in \{0,1\}^\lambda$.
2. Sign the message ("VOTE:" $||$ str_i) under a predetermined event label, e.g., ("2025 ELECTION"), on behalf of $\mathcal{V}$.

Let $(\sigma_1, ..., \sigma_s)$ be the signatures obtained by the above procedure. u gets their ballot

$$\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, \mathsf{vote}_2^{(u)}, \ldots, \mathsf{vote}_s^{(u)})$$

where $\mathsf{vote}_i^{(u)} = (str_i || \sigma_i)$ for all $i \in [s]$. As the last step of this phase, u uploads $\mathsf{vote}^{(u)}$ to ABB.

Counting phase. When the voting phase is done, anyone participating in the protocol can read ABB so that u can also check the validity of all message-event-signature tuples using the public key list of $\mathcal{V}$ and the linking algorithm and discard all ill-formed tuples. For every $(k+1)$ signatures among all signatures obtained from ABB, u invokes the Link algorithm to check whether these $(k+1)$ signatures are linked. Whenever u finds linked $(k+1)$ signatures, then u can further find all other linked signatures and discard all of them. Repeat doing this until every $(k+1)$ signatures are unlinked.

Finally, u runs S with the valid messages (that form a profile) from valid message-event-signature tuples and outputs a decision.

Analysis. We prove the following theorem.

Theorem 3. *If the underlying k-LRS scheme in Mode 1 voting has unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for Mode 1 voting has correctness and (unconditionally) privacy.*

Proof (of Theorem 3 (Security of Mode 1 Voting)). Recall that the protocol is non-interactive, and therefore, malicious voters cannot influence the result except by uploading their votes to the ABB. Obviously, the votes should satisfy the specific form defined by the rule; otherwise, they will be discarded immediately during the counting phase.

We present the following two facts to show that the counting results are correct.

1. All votes from honest voters (who vote/sign no more than k times) will remain. This is guaranteed by the randomness of str and the non-slanderability of k-LRS. Recall that if two message-event-signature tuples contain the same random string, then one of them will be discarded. While for honestly generated votes, this happens with negligible probability. Moreover, according to the non-slanderability, even after seeing at most k signatures from an honest voter, an attacker still cannot forge a signature (with a non-repeated random string) that makes those $(k+1)$ signatures linked.

2. All votes from dishonest voters (who vote/sign more than k times) will be identified and discarded. Due to the linkability of k-LRS, if a malicious voter votes mote than k times, then all signatures contained in the votes will be linked and discarded. Moreover, thanks to the non-slanderability, malicious voters cannot invalidate votes from honest voters, as analyzed above.

The voting system has anonymity due to the (unconditional) anonymity of k-LRS. Furthermore, the s message-event-signature tuples from a specific voter is confused among all ℓ (valid) tuples after the permutation by ABB, and no information is revealed except that the candidate gets exactly a score of ℓ, and hence the privacy is guaranteed. □

Remark 6 (On the complexity and hardness of efficient link). One potential concern when using k-LRS in a voting scheme that produces a total of ℓ ballots is the need to invoke the Link algorithm $\binom{\ell}{k+1}$ times, which becomes impractical for large k. While our primary contributions lie in proposing and formalizing the concept of k-linkability, addressing the weak linkability problem, and demonstrating the feasibility of ABB-based non-interactive voting with privacy, we also would like to discuss the hardness of designing an efficient linking algorithm. Notably, all existing linkable ring signatures (i.e., 1-LRS schemes)—e.g., [LWW04, LW05, TW05, ACST06, LASZ14, SALY17, DV09, LAZ19, HKSS22]—that we are aware of employ similar Link algorithms, requiring $\binom{\ell}{2}$ invocations to determine whether any two signatures are linked among ℓ signatures. This provides strong evidence of the inherent difficulty in constructing a "$\mathsf{superlink}$" algorithm—one that could *efficiently* process all ℓ signatures and directly output any linked signature tuples if they exist, within less than $\binom{\ell}{k+1}$ operations. Assuming there exists no such $\mathsf{superlink}$ algorithm, we can argue that there exists no better way to achieve private ABB-based non-interactive voting without a trusted setup: any k ballots should look like independent while any linked $k+1$ ballots should be detectable.

4.4 Approaches of Reducing the Complexity of Linking

In this subsection we explore some approaches to reduce the complexity $\binom{\ell}{k+1}$ in Mode 1 voting, though coming with trade-offs in functionality or security.

Variant 1: Use deterministic pseudoidentity additionally. Recall that to avoid a stateful signing algorithm and also to achieve the "removing all" property, we employ a re-randomized pseudoidentity instead of a deterministic pseudoidentity (see the technical overview in Sec. 1.3). In many e-cash systems (e.g., [CHL05]), a signature contains two tags: one for detecting oversigning and the other for identifying all fraud signatures if oversigning occurs. Following this approach, we can extent a user's secret key to $(\hat{sk}_1, ..., \hat{sk}_{k+1}, sk'_1, ..., sk'_k)$, where $(\hat{sk}_1, ..., \hat{sk}_{k+1})$ is a secret key of k-LRS, and $sk'_1, ..., sk'_k$ are k secret keys from an ordinary LRS scheme (i.e., 1-LRS). In this setting, a user's i-th signature includes both a re-randomized pseudoidentity $f_{\mathsf{e}}(\hat{sk}_1, ..., \hat{sk}_k; a_1, ..., a_k)$ (as shown in our construction) and a deterministic pseudoidentity $f_{\mathsf{e}}(\hat{sk}_i)$ (as in the trivial solution of using LRS k times). The deterministic pseudoidentity enables efficient detection of over-votes and the identification of two fraudulent votes. Furthermore, the re-randomized pseudoidentity helps identify the remaining $k-1$ fraudulent votes, reducing the total complexity to $\binom{\ell}{2} + \binom{\ell-2}{k-1}$. However, this approach doubles the size of the secret key and make the signing stateful, and the unconditional security is sacrificed.[16]

In more precisely, we consider the following relaxed variant of Mode 1 voting.

Preparing phase. u generates a key pair $(\hat{sk}_u, \hat{sk}_u)$ of a k-LRS scheme, and k key pairs $(sk'_{u,1}, sk'_{u,1}), ..., (sk'_{u,k}, pk'_{u,k})$ of an LRS scheme, and registers $pk_u = (\hat{pk}_u, pk'_{u,1}, ..., pk'_{u,k})$ to PKI.

Voting phase. Let $0 \le s \le k$ be the score intended by u. The voter u does the following for each $i \in [s]$:

1. Sample random strings $str_i \in \{0,1\}^\lambda$.

[16] By applying an unconditionally anonymous LRS scheme (e.g., our k-LRS scheme with $k=1$) instead, we can still achieve unconditional security. However, this will triple the size of the secret key size.

2. Use $\hat{sk}_u$ to sign the message ("VOTE:" $||$ str_i) under the predetermined event label on behalf of $\mathcal{V}$ and obtain the signature $\hat{\sigma}_i$.
3. Use $sk'_{u,i}$ to sign the message ("VOTE:" $||$ str_i $||$ $\hat{\sigma}_i$) under the predetermined event label on behalf of $\mathcal{V}$ and obtain the signature σ'_i.

Let $(\sigma_1, ..., \sigma_s)$ be the signatures obtained by the above procedure, where $\sigma_i := (\hat{\sigma}_i, \sigma'_i)$. Then u gets their ballot

$$\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, \mathsf{vote}_2^{(u)}, \ldots, \mathsf{vote}_s^{(u)})$$

where $\mathsf{vote}_i^{(u)} = (str_i||\sigma_i)$ for all $i \in [s]$. As the last step of this phase, u uploads $\mathsf{vote}^{(u)}$ to ABB.

Counting phase. When the voting phase is done, anyone participating in the protocol can read ABB so that u can also check the validity of all message-event-signature tuples using the public key list of $\mathcal{V}$ and the linking algorithm and discard all ill-formed tuples. Suppose there are ℓ well-formed ballots $(str||\hat{\sigma}||\sigma')$ left. u first invokes the (deterministic) link algorithm of the LRS scheme to check if σ' contained in the two ballots are linked. If so, then u further invokes the Link algorithm of the k-LRS scheme to identify all other fraudulent votes by checking $\hat{\sigma}$, and u removes all linked ballots. Repeat this until σ' in every pair of ballots are unlinked.

Finally, u runs S with the valid messages (that form a profile) from valid message-event-signature tuples and outputs a decision.

Analysis. Let $|\mathcal{V}| = n$ and assume $\ell \approx nk$, then the counting phase above has a computation complexity approximate to $\binom{\ell}{2} + \binom{\ell-2}{k-2}$. For the security, we prove the following theorem in Appendix K.1.

Theorem 4. *If the underlying k-LRS scheme and LRS scheme in Mode 1 voting has unforgeability, anonymity, linkability, and non-slanderability, then the stateful voting protocol for Mode 1 voting has correctness and privacy.*

Variant 2: Partition the anonymous ring. Another approach to reduce the complexity of $\binom{\ell}{k+1}$ is to partition the voter set $[n]$ properly.[17] Say, $[n]$ is divided by m subsets: $N_1, ..., N_m$ with the same cardinality. If we run our voting protocol among voters in N_i, then the number of Link calls will be $\binom{kn/m}{k+1}$, assuming $\ell \approx nk$. By combining all the results from each partition, the total number of Link calls among all partitions will be reduced to $O(n^{k+1}/m^k)$, though at the cost of reducing anonymity. Note that each voter's k ballots are now "hidden" among kn/m ballots, which maybe sufficient for an appropriate choice of m. For even larger k, similarly, we can divide k into smaller numbers at the cost of reducing privacy. We take the variant of partitioning the anonymity ring as an example and formally present the scheme as follows.

Preparing phase. u generates a key pair (sk_u, pk_u) of a k-LRS scheme and registers pk_u to PKI. Let $\mathcal{V} = (v_1, ..., v_n)$ be the universe of all voters, sorted in alphabetical order. Given that $m \mid n$, partition $\mathcal{V}$ into m disjoint subsets $\mathcal{V}_1, ..., \mathcal{V}_m$, where for each $j \in [m]$, $\mathcal{V}_j = (v_{(j-1)n/m+1}, ..., v_{jn/m})$.

Voting phase. Let $0 \le s \le k$ be the score intended by u. The voter u does the following for each $i \in [s]$:

1. Sample random strings $str_i \in \{0,1\}^\lambda$.
2. Sign the message ("VOTE:" $||$ str_i) under the predetermined event label on behalf of $\mathcal{V}_j$, where $\mathcal{V}_j$ denotes the subset of voters to which u belongs.

Let $(\sigma_1, ..., \sigma_s)$ be the signatures obtained by the above procedure. u gets their ballot

$$\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, \mathsf{vote}_2^{(u)}, \ldots, \mathsf{vote}_s^{(u)})$$

where $\mathsf{vote}_i^{(u)} = (str_i||\sigma_i)$ for all $i \in [s]$. As the last step of this phase, u uploads $\mathsf{vote}^{(u)}$ to ABB.

Counting phase. When the voting phase is done, anyone participating in the protocol can read ABB so that u can also check the validity of all message-event-signature tuples using the public key list of $\mathcal{V}$ and the linking algorithm and discard all ill-formed tuples. For every $(k+1)$ signatures among all signatures obtained from ABB, u invokes the Link algorithm to check whether these $(k+1)$ signatures are linked,

[17] The idea of partitioning has been explored in some works, e.g., [BEHG20, CEL+23].

within the corresponding sub-ring of size n/m. Whenever u finds linked $(k+1)$ signatures, then u can further find all other linked signatures and discard all of them. Repeat doing this until every $(k+1)$ signatures are unlinked.

Finally, u runs S with the valid messages (forming a profile) from valid message-event-signature tuples and outputs a decision.

Analysis. Let $|\mathcal{V}| = n$ and assume $\ell \approx nk$, then the counting phase above has a computation complexity approximate to $m \cdot \binom{kn/m}{k+1}$. The security is shown by the following theorem, whose proof is similar to Theorem 3 and we safely omit here.

Theorem 5. *If $m \mid n$ and the underlying k-LRS scheme in Mode 1 voting has unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for Mode 1 voting has correctness and (unconditionally) privacy among the sub-ring of size n/m.*

4.5 Mode 2 Voting from k-LRS and t-LRS

Suppose there are q candidates $C_1, ..., C_q$. Let $k = k_1$ and $x, t \in \mathbb{N}^+$ such that $k_1 + k_2(x-1) \leq t$ and $(k_2+1)x > t$. Note that there always exists a fixed solution to $(x,t) = (k_1, k_1k_2 + k_1 - k_2)$; however, we let (x,t) be a parameter that can be freely chosen since one could further optimize in implementation level depending on the value of (k_1, k_2). For example, if $k_1 = k_2 = k$, we can let $(x,t) = (1,k)$.

The only difference between Mode 1 and Mode 2 is how to generate ballots that contain signatures. In Mode 1, a ballot can be divided into parts, with each part only containing one signature, and they look independent of each other. However, in Mode 2, we combine many signatures into one part and sign again for this entire part of a ballot. The details are in below:

Preparing phase. u generates a key pair (sk_u, pk_u) of a k-LRS scheme and a key pair $(\hat{sk}_u, \hat{pk}_u)$ of t-LRS, and registers $(pk_u, \hat{pk}_u)$ to PKI.

Voting phase. W.l.o.g., assume u wants to produce k number of ballots. If u wants to produce $k' < k$ ballots, then they just skips $k - k'$ iterations. For each $i \in [k]$, let $c_i \in [q]$ and C_{c_i} be the candidate u wants to vote for. User u does the following:

1. For $j \in [x]$,
   (a) Sample a random string $str_{i,j} \in \{0,1\}^\lambda$.
   (b) Use $\hat{sk}_u$ to sign the message ("VOTE:" || C_{c_i} || $str_{i,j}$) under an event label ("CANDIDATE:" || C_{c_i}) on behalf of $\mathcal{V}$.

   Let $\tau_i = ((C_{c_i}||str_{i,1}||\sigma_{i,1}), \ldots, (C_{c_i}||str_{i,x}||\sigma_{i,x}))$ be the (abstract) message-signature pairs obtained by the above process.
2. For $j \in [q] \setminus \{c_i\}$,
   (a) Sample random strings $str'_{i,j} \in \{0,1\}^\lambda$.
   (b) Use $\hat{sk}_u$ to sign the message ("VOTE:" || C_{c_i} || $str'_{i,j}$) under an event label ("CANDIDATE:" || C_j) on behalf of $\mathcal{V}$. (Note: Be aware of the event label!)

   Let $\tau'_i = \left((C_{c_j}||str'_{j,1}||\sigma'_{j,1})\right)_{j \in [q]\setminus\{c_i\}}$ be the message-signature pairs obtained by the above process.
3. samples a random string $str''_i \in \{0,1\}^\lambda$, and
4. uses sk_u to sign the message ("VOTE:" || C_i || str''_i || τ_i || τ'_i) under an event label ("VOTE") on behalf of $\mathcal{V}$, and gets a signature σ''_i.

u gets the ballot $\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, \mathsf{vote}_2^{(u)}, \ldots, \mathsf{vote}_k^{(u)})$, where

$$\mathsf{vote}_i^{(u)} = (C_{c_i}||str''_i||\tau_i||\tau'_i||\sigma''_i)$$

for $i \in [k]$. As the last step of this phase, u uploads $\mathsf{vote}^{(u)}$ to ABB.

Counting phase. This phase is almost the same as Mode 1, but there are two types of signatures, so one should use two different link functions from k-LRS and t-LRS with respect to the category of signatures.

Remark 7 (On the necessity of using distinct random strings). We emphasize that in Mode 2 voting protocol above, it is essential to sample a fresh random string for every signing operation, including the generation of the inner signatures $\sigma_{i,j}, \sigma'_{i,j}$ and the outer signature σ''_i. A vote $\mathsf{vote}_i^{(u)} = (C_{c_i}||str''_i||\tau_i||\tau'_i||\sigma''_i)$ is considered valid only if all random strings within it are distinct. To see the reason, consider an attack where the adversary, intending to vote for candidate C, samples a single random string str and signs the message ("VOTE:" $||$ C $||$ str) under an event label ("CANDIDATE:" $||$ C) only once. Then, leveraging the malleability of the signature scheme, the adversary generates $x-1$ additional signatures on the same message and event label (for a non-strongly unforgeable signature scheme, it is feasible to generate another valid signature after seeing one valid message-signature pair). Note that this does not break the linkability since the message and the label remain unchanged. However, if repeated strings within a vote are allowed, the adversary can cast more than k_2 votes for the same candidate while keeping those votes unlinked, as the adversary signs only once instead of x times as required. To protect the protocol from such attacks, we enforce the use of distinct random strings when verifying the validity of each vote.

Analysis. We prove the following theorem in the Appendix K.1.

Theorem 6. *If the underlying k-LRS and t-LRS schemes in Mode 2 voting have unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for Mode 2 voting has correctness and privacy.*

Remark 8. One drawback of Mode 2 voting is that it achieves computationally privacy rather than unconditionally privacy, even the underlying k-LRS scheme has unconditional anonymity. This is because the secret key $\hat{sk}_u$ of t-LRS is used to sign signatures under different event labels, and the total signing time may exceed the threshold t (but for an honest voter, the total signing times on each event is still bounded by t), and thus unconditional anonymity does not hold anymore (see Remark 3 for more discussions). Nevertheless, we can still achieve unconditional anonymity by additionally restricting $k_1(q+x-1) \leq t$ to guarantee that the total number of signatures (even under different event) does not exceed t. This will result in large parameters x and t and hence it is practical only when q and $k_1 = k_2$ are small (since there is a solution for x and t only when $k_1 = k_2$). To better present the idea of Mode 2 voting, we consider the scheme in the computational anonymity setting.

5 Negative Result on Privacy

The Mode 2 protocol (Sec. 4.5) achieves nearly all desirable properties, except that undervotes are possible, i.e., a malicious voter may cast fewer than the required k votes. In this section, we formally prove that for an ABB-based non-interactive voting protocol (cf. Def. 8) with privacy, it is impossible to protect from undervote behaviors. Recall that we assume public key infrastructure (PKI) and anonymous bulletin board (ABB) model. That is, we assume a PKI where everyone has access to the public keys of others, and an ABB on which everyone can write privately. Moreover, there is no direct communication channel between any two parties except ABB.

Let $q \in \mathbb{N}^+$, and $C_1, ..., C_q$ be q candidates of the voting event.

Definition 12 (Overvote and undervote). *Let $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathcal{D}, \mathsf{S})$ be a k-voting specification (i.e., a well-formed vote is in the form of $\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, ..., \mathsf{vote}_k^{(u)})$ for $u \in [n]$). Let P be an initial input of a voting protocol Π with respect to Γ. We say P has overvote (resp., an undervote), if there exists $\mathsf{vote}^{(u)} = (\mathsf{vote}_1^{(u)}, ..., \mathsf{vote}_{k'}^{(u)}) \in P$ such that $k' > k$ (resp., $k' < k$), i.e., there is a malicious voter who casts more (resp., less) than k votes.*

We say the voting protocol Π has overvote (resp., undervote) detection if the running of Π with initial inputs P additionally outputs an indicator bit b, showing whether P contains an overvote (resp., an undervote).

We say the voting protocol Π has overvote (resp., undervote) identification if the running of Π with initial inputs P additionally outputs a set $L \subseteq P$ consists of all overvotes (resp., undervotes) from the malicious voter in P.

Example 6. Consider a 3-voting specification among two voters be Arthur and Bertha (indexed by 1 and 2, respectively). Suppose there are six candidates, A, B, C, D, E, and F, and the voters are required to select their top three without ranking them. Each candidate can only be selected once by a voter. Therefore, a profile

$$P = (\mathsf{vote}^{(1)}, \mathsf{vote}^{(2)})^T = \begin{bmatrix} A & B & C \\ D & F & F \end{bmatrix},$$

indicates that Arthur casts on A, B, and C, and Bertha casts on D, E, and F, respectively.

Now consider an ill-formed profile

$$P' = \begin{bmatrix} A & B & C & A \\ D & E & F & \end{bmatrix},$$

which indicates that malicious voter Arthur casts two votes for A, and one vote each for B and C. If Π has overvote identification, then there should be a mechanism to identify all votes (i.e., (A, B, C, A)) from the dishonest Arthur, leading to a voting tally based on the votes $[D, E, F]$ left.

Consider another ill-formed profile

$$P'' = \begin{bmatrix} A & B & \perp \\ D & E & F \end{bmatrix},$$

which indicates that malicious voter Arthur casts only two votes (for A and B, respectively), and abstains from casting his third vote. If Π has undervote identification, then there should be a mechanism to identify all votes (i.e., (A, B)) from the dishonest Arthur, again leading to a voting tally based on the votes $[D, E, F]$ left. ◁

Theorem 7 (Negative result on privacy). *Let $\Gamma = (\mathcal{V} = [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a k-voting specification, and Π be an ABB-based non-interactive voting protocol for Γ. If Π has overvote identification and privacy, then it is impossible for Γ to have undervote identification.*

The proof is deferred to Appendix K.2.

Remark 9. The negative result in Theorem 7 holds only for non-interactive settings. For interactive voting settings, it might be possible to achieve both privacy and undervote detectable. For example, after uploading $\mathsf{vote} = (\mathsf{vote}_1, ..., \mathsf{vote}_k)$, each voter may generate a non-interactive zero-knowledge (NIZK) proof to show that they indeed follow the voting rule and assign exactly k votes. However, to hide the correlation among different votes, the instance of the NIZK language needs to include all votes from other voters. That said, a voter needs to wait for the publication of others' voting results before generating proof and completing the voting process, and this does not satisfied the definition of non-interactive voting.

Remark 10. In general, overvote is considered to be more pernicious than undervote, and in many cases undervote is a permitted choice (as there is usually an option to abstain). Therefore, when designing a voting scheme, avoiding overvote is a more relevant goal—even undervote cannot be guaranteed simultaneously.

6 Instantiating k-LRS from Aggregatable Identification Schemes

In this section we introduce aggregatable identification schemes (AIS), and show a generic construction of k-LRS from $(k+1)$-AIS. We defer the construction from the SIS assumption and the security analysis in Appendix D due to space limitation.

6.1 Identification Schemes

Definition 13 (Identification schemes). *An identification scheme (also referred to as a Sigma protocol) is a three-move protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$.*

- $pp \leftarrow \mathsf{Setup}(1^\lambda)$. *The setup algorithm takes as input the security parameter and outputs a randomly sampled public parameter* $pp \in \mathcal{PP}$, *which defines the key space* $\mathcal{SK} \times \mathcal{PK}$, *the extended secret key space* $\ddot{\mathcal{SK}}$, *the commitment space* $\mathcal{CMT}$, *the challenge space* $\mathcal{CH}$ *and the response space* $\mathcal{RSP}$. *Here* $\ddot{\mathcal{SK}}$ *is used for defining the special soundness below, and* $\mathcal{SK} \subseteq \ddot{\mathcal{SK}}$.
- $(sk, pk) \leftarrow \mathsf{Gen}(pp)$. *The key generation algorithm takes as input* pp *and outputs a pair of secret and public keys* (sk, pk). *In more details,* Gen *first samples a random* sk *from* $\mathcal{SK}$, *and then computes the corresponding* pk *from* sk *via a one-way function defined by* pp, *i.e.,* $pk := f_{pp}(sk)$.
- $\langle \mathcal{P}, \mathcal{V} \rangle$. *The interactive 3-move protocol between a prover* $\mathcal{P}$ *with* sk *and a verifier* $\mathcal{V}$ *with* pk.
  - *In the first move,* $\mathcal{P}$ *sends out a commitment* cmt *computed from* pp *and randomness* ρ, *i.e.,* $cmt \leftarrow \mathsf{Commit}(pp; \rho)$.
  - *In the second move,* $\mathcal{V}$ *sends a challenge* $ch \in \mathcal{CH}$.
  - *In the third move,* $\mathcal{P}$ *sends a response* rsp *computed from* sk, ch, *and* ρ, *i.e.,* $rsp \leftarrow \mathsf{Response}(sk, ch, \rho)$.
  - *There is a verification algorithm* $\mathsf{Ver}(pp, pk, (cmt, ch, rsp))$ *that outputs a bit, indicating the validity of a transcript* (cmt, ch, rsp).
    Moreover, Ver *is defined via a deterministic algorithm* $\overline{\mathsf{Sim}}(pp, pk, ch, rsp)$ *that outputs a commitment* cmt', *and* $\mathsf{Ver}(pp, pk, (cmt, ch, rsp)) = 1$ *if and only if* $cmt' = cmt$.

Public-coin protocols. *The protocol* $\langle \mathcal{P}, \mathcal{V} \rangle$ *is public-coin if* ch *in the second move is randomly sampled from* $\mathcal{CH}$. *By default, in this work, we only consider identification schemes with public-coin protocols.*

Correctness. *For every* pp *and* $(sk, pk) \leftarrow \mathsf{Gen}(pp)$, *if* $\mathcal{P}$ *and* $\mathcal{V}$ *follow the 3-move protocol honestly, then* $\mathcal{V}$ *will accept with all but negligible probability.*

Besides correctness, we additionally require the following properties.

- ***One-wayness***. *The public parameter* pp *defines a one-way function* $f_{pp}(\cdot)$ *that maps an (extended) secret key* $sk \in \ddot{\mathcal{SK}}$ *to a public key* $pk \in \mathcal{PK}$. *Meanwhile, given* $pp \leftarrow \mathsf{Setup}(1^\lambda)$ *and* pk *where* $(sk, pk) \leftarrow \mathsf{Gen}(pp)$, *it is computationally infeasible to output* $sk' \in \ddot{\mathcal{SK}}$ *such that* $f_{pp}(sk') = pk$.
- κ***-min-entropy***. *For* $pp \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(sk, pk) \leftarrow \mathsf{Gen}(pp)$, *it holds that* $\mathbf{H}_\infty(cmt) \geq \kappa$, *where* cmt *is the first message in the protocol* $\langle \mathcal{P}, \mathcal{V} \rangle$.
- ϵ***-special soundness***. *For* $pp \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(sk, pk) \leftarrow \mathsf{Gen}(pp)$, *given two valid transcripts* (cmt, ch, rsp) *and* (cmt, ch', rsp') *such that both* ch *and* ch' *satisfy a uniform distribution over* $\mathcal{CH}$, *there exists an algorithm* $\mathsf{Extract}$ *that with probability at least* ϵ *extracts a secret key* $sk' \in \ddot{\mathcal{SK}}$ *of* pk *from* (ch, rsp) *and* (ch', rsp'). *Namely,* $\Pr[sk' \leftarrow \mathsf{Extract}(ch, rsp, ch', rsp') : f_{pp}(sk') = pk] \geq \epsilon$, *where the probability space is taken over the choice of* $pp, (sk, pk), ch$ *and* ch'.
- ***Simulatability***. *There exists an efficient simulator* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *s.t., for any* pp, $(sk, pk) \leftarrow \mathsf{Gen}(pp)$ *and* $ch \in \mathcal{CH}$, $\mathsf{Sim}(pp, pk, ch)$ *outputs a simulated transcript* (cmt, ch, rsp) *that is statistically close to a transcript outputted by an honest execution of* $\langle \mathcal{P}, \mathcal{V} \rangle$ *conditioned on the the challenge being* ch. *More precisely, the simulation is done by first sampling* rsp *according to some distribution, i.e.,* $rsp \leftarrow \mathsf{Sim}_1()$, *and then computing* cmt *via* $cmt \leftarrow \mathsf{Sim}_2(pp, pk, ch, rsp)$.
  If $\langle \mathcal{P}, \mathcal{V} \rangle$ *is public-coin, we simply omit the input* ch *by writing* $\mathsf{Sim}(pp, pk)$.
- Q***-pseudorandomness.*** *It is computationally infeasible to distinguish the following two distributions*

$$\begin{pmatrix} pp & pp_1 & pp_2 & ... & pp_Q \\ f_{pp}(sk) & f_{pp_1}(sk_1) & f_{pp_2}(sk) & ... & f_{pp_Q}(sk) \end{pmatrix} \text{ and } \begin{pmatrix} pp & pp_1 & pp_2 & ... & pp_Q \\ f_{pp}(sk) & u_1 & u_2 & ... & u_Q \end{pmatrix},$$

  where $pp, pp_1, ..., pp_Q$ *are independent and random parameters,* sk *is a randomly sampled secret key, and* $u_1, ..., u_Q$ *are random elements in the public key space* $\mathcal{PK}$.
  If Q *can be any polynomial of the security parameter, we simply call it pseudorandomness.*

Definition 14 (Homomorphism on coefficient space $\mathbb{F}$). *We say an identification scheme is homomorphic on coefficient space* $\mathbb{F}$, *if it satisfies the following properties.*

– *For any pp and valid key pair (sk, pk), any coefficient $a \in \mathbb{F}$, we define an operation $\cdot$ that maps pk to another public key $a \cdot pk \in \mathcal{PK}$.*
– *$a \cdot \rho \in \mathcal{R}$ for any $a \in \mathbb{F}$ and $\rho \in \mathcal{R}$, where $\mathcal{R}$ is the randomness space in* Commit.
– *For any pp and (sk, pk), let (cmt, ch, rsp) be a transcript where $cmt \leftarrow \mathsf{Commit}(pp; \rho)$ and $rsp \leftarrow \mathsf{Response}(sk, ch, \rho)$. There is a coefficient-verification algorithm $\mathsf{CoeVer}(pp, epk, (ecmt, ch, rsp), a)$ that outputs the same result as the verification algorithm $\mathsf{Ver}(pp, pk, (cmt, ch, rsp))$, where $epk := a \cdot pk$, and $ecmt \leftarrow \mathsf{Commit}(pp; a \cdot \rho)$. Moreover, there simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ in Def. 13 can be extended to output a simulated transcript $(ecmt, ch, rsp)$ that is statistically close to the output by an honest execution, where $rsp \leftarrow \mathsf{Sim}_1()$, and $ecmt \leftarrow \mathsf{Sim}_2(pp, epk, ch, a)$.*

6.2 Aggregatable Identification Schemes

Now, we propose the concept of aggregatable identification schemes, which serves as the main building block in our k-linkable ring signatures (k-LRS) with unconditional anonymity.

Definition 15 ($(k+1)$-Aggregatable identification schemes). *Let $k \in \mathbb{N}^+$. A $(k+1)$-aggregatable identification scheme is defined as an identification scheme in Def. 13 with the following additional properties.*

– ***Aggregation**. For $(k+1)$ valid key pairs $(sk_1, pk_1), ..., (sk_{k+1}, pk_{k+1})$ under $(k+1)$ different parameters $pp_1, ..., pp_{k+1}$, there exist two deterministic algorithms* AggrPK *and* AggrCMT *satisfying the following properties.*
  • *$\widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, ..., pk_{k+1})$. The algorithm* AggrPK *aggregates all public keys into a public key $\widetilde{pk}$ via operation "+", i.e., $\widetilde{pk} := \sum_{i \in [k+1]} pk_i$.*
  • *$\widetilde{cmt} \leftarrow \mathsf{AggrCMT}(cmt_1, ..., cmt_{k+1})$. The algorithm* AggrCMT *aggregates all commitments into a commitment $\widetilde{cmt}$, where $cmt_i \leftarrow \mathsf{Commit}(pp_i; \rho_i)$ for all $i \in [k+1]$.*
  • *$pp := (pp_1, ..., pp_{k+1})$ defines another 3-move protocol with key space $\mathcal{SK}^{k+1} \times \mathcal{PK}$, the extended secret key space $\ddot{\mathcal{SK}}^{k+1}$, commitment space $\mathcal{CMT}$, challenge space $\mathcal{CH}$, and response space $\mathcal{RSP}^{k+1}$, and $sk := (sk_1, ..., sk_{k+1})$ is a corresponding secret key under pp and $\widetilde{pk}$. Namely, $f_{pp}(sk) = \widetilde{pk}$.*
  • ***Correctness**. Let $rsp_i \leftarrow \mathsf{Response}(sk_i, ch, \rho_i)$ for all $i \in [k+1]$, and $rsp := (rsp_1, ..., rsp_{k+1})$. Then with overwhelming probability $(\widetilde{cmt}, ch, rsp)$ is a valid transcript, where the probability is taken over the choices of $\{\rho_i\}_{i \in [k+1]}$, and ch.*
  • *Moreover, it has one-wayness, min-entropy, special soundness, simulatability, and pseudorandomness, as defined in a canonical IS scheme.*
– ***Collision resistance of secret key**. Given pp, $sk = (sk_1, ..., sk_{k+1})$ generated from* Setup *and* Gen*, and the aggregated public key $\widetilde{pk}$, it is computationally infeasible to find two distinct secret keys $sk, sk' \in \ddot{\mathcal{SK}}^{k+1}$ such that $f_{pp}(sk) = f_{pp}(sk')$.*
– ***Uniformity**. Assume $pp = (pp_1, ..., pp_{k+1})$ is randomly sampled. Fixing any $sk_1, ..., sk_k$, if sk_{k+1} is distributed uniformly, then $\widetilde{pk}$ has a distribution statistically close to the uniform distribution over $\widetilde{\mathcal{PK}}$, where $\widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, ..., pk_{k+1})$, and $pk_i := f_{pp_i}(sk_i)$ for $i \in [k+1]$.*

Definition 16 (Downward compatible AIS). *Let* AIS *be a $(k+1)$-AIS with $k \in \mathbb{N}^+$. We say* AIS *is downward compatible if it is also a k'-AIS for any $k' \in \mathbb{N}$ and $k' < k$.*

Definition 17 (Homomorphism for AIS). *We say a k-AIS is homomorphic on coefficient space $\mathbb{F}$, if it additionally satisfies the following.*

– *For any coefficients $a_1, ..., a_k \in \mathbb{F}$, any k valid key pairs $(sk_1, pk_1), ..., (sk_k, pk_k)$ under k different parameters $pp_1, ..., pp_k$, let $(\widetilde{cmt}, ch, rsp = (rsp_1, ..., rsp_k))$ be a valid transcript for AIS, where $cmt_i := \mathsf{Commit}(pp_i; \rho_i)$, $rsp_i \leftarrow \mathsf{Response}(sk_i, ch, \rho_i)$ for $i \in [k]$, and $\widetilde{cmt} \leftarrow \mathsf{AggrCMT}(cmt_1, ..., cmt_k)$. Then there is a coefficient-verification algorithm $\mathsf{CoeVer}(pp, \widetilde{epk}, (\widetilde{ecmt}, ch, rsp), a_1, ..., a_k)$ that outputs the same result as the verification algorithm $\mathsf{Ver}(pp, \widetilde{pk}, (\widetilde{cmt}, ch, rsp))$, where $\widetilde{epk} \leftarrow \mathsf{AggrPK}(a_1 \cdot pk_1, ..., a_k \cdot pk_k)$, and $\widetilde{ecmt} \leftarrow \mathsf{AggrCMT}(\mathsf{Commit}(pp_1; a_1 \cdot \rho_1), ..., \mathsf{Commit}(pp_k; a_k \cdot \rho_k))$.*

Moreover, the simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *can be extended to output a simulated transcript* $(\widetilde{ecmt}, ch, rsp)$ *that is statistically close to the output by an honest execution, where* $rsp \leftarrow \mathsf{Sim}_1()$, *and* $\widetilde{ecmt} \leftarrow \mathsf{Sim}_2(pp, \widetilde{epk}, ch, a_1, ..., a_k)$.

– ***Infeasiblity of kernel***. *For randomly sampled* $\{pk_i\}_{i\in[k]}$, *it is computationally infeasible to find a group of efficients* $(a_1, ..., a_k) \in \mathbb{F}^k$ *such that* $\sum_{i\in[k]} a_i \cdot pk_i = 0_{\mathcal{PK}}$, *where* $0_{\mathcal{PK}}$ *is the unit elements in the group* $\mathcal{PK}$.

6.3 DL-based AIS

In this subsection we show instantiation of $(k+1)$-AIS based on the DL assumption. Let GGen be a group generation algorithm that outputs $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of prime order q with generator g.

Definition 18 (The DL assumption). *The discrete logarithm (DL) assumption states that, for any PPT adversary* $\mathcal{A}$, *its advantage*

$$\mathsf{Adv}^{dl}_{\mathbb{G},\mathcal{A}}(\lambda) := \Pr[x \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}(\mathbb{G}, q, g, g^x) = x]$$

is negligible over λ.

Definition 19 (The FindRep assumption [Bra93]). *The* $(k+1)$*-find representation (*$(k+1)$*-FindRep) assumption states that, for any PPT adversary* $\mathcal{A}$, *its advantage*

$$\mathsf{Adv}^{findrep}_{\mathbb{G},k+1,\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{c} For\ i \in [k+1] : \omega_i \xleftarrow{\$} \mathbb{Z}_q; g_i := g^{\omega_i} \\ ((x_1, ..., x_{k+1}), (x'_1, ..., x'_{k+1})) \leftarrow \mathcal{A}(\mathbb{G}, q, g_1, ..., g_{k+1}) \end{array} : \begin{array}{c} (x_1, ..., x_{k+1}) \neq (x'_1, ..., x'_{k+1}) \\ \wedge\ \prod_{i\in[k+1]} g_i^{x_i} = \prod_{i\in[k+1]} g_i^{x'_i} \end{array}\right]$$

is negligible over λ.

It is easy to see that the $(k+1)$-FindRep assumption implies k-FindRep assumption for all $k \geq 1$, and the 1-FindRep assumption is just the DL assumption. On the other hand, the DL assumption implies the k-FindRep assumption with a loss factor k.

The $(k+1)$-FindRep assumption also implies the following.

$$\Pr\left[\begin{array}{c} \text{For } i \in [k+1] : \omega_i \xleftarrow{\$} \mathbb{Z}_q; g_i := g^{\omega_i}; X \xleftarrow{\$} \mathbb{G} \\ (x_1, ..., x_{k+1}) \leftarrow \mathcal{A}(\mathbb{G}, q, g_1, ..., g_{k+1}, X) \end{array} : X = \prod_{i\in[k+1]} g_i^{x_i}\right] \leq negl(\lambda),$$

$$\Pr\left[\begin{array}{c} \text{For } i \in [k+1] : \omega_i \xleftarrow{\$} \mathbb{Z}_q; g_i := g^{\omega_i}; x_i \xleftarrow{\$} \mathbb{Z}_q; X := \prod_{i\in[k+1]} g_i^{x_i} \\ (x'_1, ..., x'_{k+1}) \leftarrow \mathcal{A}(\mathbb{G}, q, g_1, ..., g_{k+1}, X, x_1, ..., x_k) \end{array} : X = \prod_{i\in[k+1]} g_i^{x'_i}\right] \leq negl(\lambda),$$

$$\Pr\left[\begin{array}{c} \text{For } i \in [k+1] : \omega_i \xleftarrow{\$} \mathbb{Z}_q; g_i := g^{\omega_i} \\ (x_1, ..., x_{k+1}) \leftarrow \mathcal{A}(\mathbb{G}, q, g_1, ..., g_{k+1}) \end{array} : (x_1, ..., x_{k+1}) \neq \mathbf{0} \ \wedge\ 0_{\mathbb{G}} = \prod_{i\in[k+1]} g_i^{x_i}\right] \leq negl(\lambda).$$

Definition 20 (The DDH assumption). *The decisioal Diffie-Hellman (DDH) assumption states that, for any PPT adversary* $\mathcal{A}$, *its advantage*

$$\mathsf{Adv}^{ddh}_{\mathbb{G},\mathcal{A}}(\lambda) := \left|\Pr[x, y \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1] - \Pr[x, y, z \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1]\right|$$

is negligible over λ.

It has been proved that the DDH assumption implies the following with a tight reduction [EHK+17, GHKW16], for any $Q = poly(\lambda)$.

$\underline{\mathsf{Setup}(1^\lambda)}$:	$\underline{\mathsf{Gen}(pp)}$:
$(\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$	$x \xleftarrow{\$} \mathbb{Z}_q$; $X := g^x$
Return $pp := (\mathbb{G}, q, g)$	Return $(sk, pk) := (x, X)$.

$\underline{\langle \mathcal{P}, \mathcal{V} \rangle}$:

$\mathcal{P}(sk = x)$		$\mathcal{V}(pk = X)$
$r \xleftarrow{\$} \mathbb{Z}_q$	$\xrightarrow{cmt := g^r}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \mathbb{Z}_q$
$rsp := r - ch \cdot x$	$\xrightarrow{rsp}$	Return 1 if $cmt = g^{rsp} \cdot X^{ch}$?

Fig. 5. The DL-based identification scheme [Sch91].

$$\mathsf{Adv}^{ddh}_{\mathbb{G},\mathcal{A}}(\lambda) := \left| \begin{array}{l} \Pr\left[x, \omega_1, ..., \omega_Q \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}\left(\mathbb{G}, q, \begin{matrix} g & g^{\omega_1} & ... & g^{\omega_Q} \\ g^x & g^{\omega_1 x} & ... & g^{\omega_Q x} \end{matrix}\right) = 1\right] \\ - \Pr\left[x, \omega_1, ..., \omega_Q, u_1, ..., u_Q \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}\left(\mathbb{G}, q, \begin{matrix} g & g^{\omega_1} & ... & g^{\omega_Q} \\ g^x & g^{u_1} & ... & g^{u_Q} \end{matrix}\right) = 1\right] \end{array} \right| \leq negl(\lambda).$$

Now, we show the first $(k+1)$-AIS scheme from the DL assumption in Fig. 5. Here $pp = (\mathbb{G}, q, g)$ defines $\mathcal{SK} = \mathbb{Z}_q$, $\mathcal{PK} = \mathbb{G}$, and the extended secret key space $\ddot{\mathcal{SK}} = \mathcal{SK} = \mathbb{Z}_q$.

Theorem 8. *If the DDH assumption holds in $\mathbb{G}$ (which implies that the DL assumption holds in $\mathbb{G}$), then the scheme in Fig. 5 is a secure identification scheme. More precisely, it has correctness, one-wayness, $(\log q)$-min-entropy, $(1 - 1/q)$-special soundness, perfect simulatability, and pseudorandomness. Moreover, it has homomorphism on coefficient space $\mathbb{Z}_q^*$.*

Proof. We omit the proofs for the security of identification but just show its homomorphism. Namely, given $a \in \mathbb{Z}_q^*$, a valid transcript $(cmt = g^r, ch, rsp)$, $epk = X^a$ and $ecmt = g^{a \cdot r}$, the coefficient-verification algorithm $\mathsf{CoeVer}(pp, epk, (ecmt, ch, rsp), a)$ returns whether $ecmt = g^{rsp \cdot a} \cdot (epk)^{ch}$. □

Aggregation. Let $(\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$. For $i \in [k+1]$, $\omega_i \xleftarrow{\$} \mathbb{Z}_q$, and $pp_i := (\mathbb{G}, q, g_i := g^{\omega_i})$. Let $(sk_i = x_i, pk_i = X_i = g_i^{x_i})$ be $(k+1)$ valid key pairs under $\{pp_i\}_{i \in [k+1]}$, the algorithms AggrPK and $\mathsf{AggrCMT}$ are defined as follows.

- $\mathsf{AggrPK}(pk_1, ..., pk_{k+1})$ returns $\widetilde{pk} := \prod_{i \in [k+1]} pk_i = \prod_{i \in [k+1]} g_i^{x_i}$.
- Let $cmt_i := g_i^{r_i}$ for $i \in [k+1]$. $\mathsf{AggrCMT}(cmt_1, ..., cmt_{k+1})$ returns $\widetilde{cmt} := \prod_{i \in [k+1]} cmt_i = \prod_{i \in [k+1]} g_i^{r_i}$.
- $pp := (pp_1, ..., pp_{k+1})$ defines the 3-move identification protocol in Fig. 6.

$\underline{\langle \mathcal{P}, \mathcal{V} \rangle}$:

$\mathcal{P}(sk = (x_1, ..., x_{k+1}))$		$\mathcal{V}(\widetilde{pk})$
$r_1, ..., r_{k+1} \xleftarrow{\$} \mathbb{Z}_q$; $\widetilde{cmt} := \prod_{i \in [k+1]} g_i^{r_i}$	$\xrightarrow{\widetilde{cmt}}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \mathbb{Z}_q$
For $i \in [k+1]$: $rsp_i := r_i - ch \cdot x_i$; $rsp := (rsp_1, ..., rsp_k)$	$\xrightarrow{rsp}$	Return 1 if $\widetilde{cmt} = \prod_{i \in [k+1]} g_i^{rsp_i} \cdot \widetilde{pk}^{ch}$?

Fig. 6. The aggregation of the DL-based identification scheme.

Theorem 9. *If the DDH assumption (equivalently, the DL assumption and the $(k+1)$-FindRep assumption) holds in $\mathbb{G}$, then the scheme above is a secure $(k+1)$-AIS scheme. More precisely, it has correctness, one-wayness, $(\log q)$-min-entropy, $(1-1/q)$-special shoundness, perfect simulatability, and pseudorandomness. Moreover, it has downward compatibility and homomorphism on coefficient space $\mathbb{Z}_q^*$.*

Proof. The correctness is straightforward. Now, we show the aggregation forms a secure 3-move identification protocol.

One-wayness. Let $pp = (\mathbb{G}, q, g_1, ..., g_{k+1})$ and $sk = (x_1, ..., x_{k+1})$. Define $\widetilde{pk} = f_{pp}(sk) = \prod_{i\in[k+1]} g_i^{x_i}$. Meanwhile, for randomly sampled pp and sk, the aggregated public key $\widetilde{pk}$ distributes uniformly over $\mathbb{G}$. Under the $(k+1)$-FindRep assumption, given only pp and $\widetilde{pk}$, it is computationally infeasible to output sk' s.t. $f_{pp}(sk') = \widetilde{pk}$.

Min-entropy. If $r_1, ..., r_{k+1}$ are randomly sampled from $\mathbb{Z}_q$, then $\widetilde{cmt} = \prod_{i\in[k+1]} g_i^{r_i}$ is a uniform distribution over $\mathbb{G}$ and therefore it has a min-entropy $\log q$.

Special soundness. Let $(\widetilde{cmt}, ch, rsp = (rsp_1, ..., rsp_{k+1}))$ and $(\widetilde{cmt}, ch', rsp' = (rsp'_1, ..., rsp'_{k+1}))$ be two valid transcripts and $ch, ch' \xleftarrow{\$} \mathbb{Z}_q$. Namely,

$$\widetilde{cmt} = \prod_{i\in[k+1]} g_i^{rsp_i} \cdot \widetilde{pk}^{ch} = \prod_{i\in[k+1]} g_i^{rsp'_i} \cdot \widetilde{pk}^{ch'}.$$

With probability at least $1-1/q$ we have $ch \neq ch'$, and

$$\widetilde{pk} = \prod_{i\in[k+1]} g_i^{(rsp'_i - rsp_i)(ch-ch')^{-1}}.$$

Therefore, we can extract a secret key for $\widetilde{pk}$ under $pp = (\mathbb{G}, q, g_1, ..., g_{k+1})$, which breaks the $(k+1)$-FindRep assumption.

Perfect simulatability. Let ch be a random challenge. The simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ which takes as input $(pp, \widetilde{pk}, ch)$ and outputs a simulated transcript $(\widetilde{cmt}, ch, rsp = (rsp_1, ..., rsp_{k+1}))$ as follows.

1. $\mathsf{Sim}_1()$ samples $rsp_i \xleftarrow{\$} \mathbb{Z}_q$ for $i \in [k+1]$ and outputs $rsp = (rsp_1, ..., rsp_{k+1})$.
2. $\mathsf{Sim}_2(pp, \widetilde{pk}, ch, rsp)$ outputs $\widetilde{cmt} := \prod_{i\in[k+1]} g_i^{rsp_i} \cdot \widetilde{pk}^{ch}$.

It is easy to see that the simulated transcript distributes identically as the transcript from an honest execution of $\langle \mathcal{P}, \mathcal{V} \rangle$.

Pseudorandomness. Let $(g_1, ..., g_k), (h_{1,1}, ..., h_{1,k}), ..., (h_{Q,1}, ..., h_{Q,k})$ be $(Q+1)$ groups of generators randomly sampled over $\mathbb{G}$, and let $x_1, ..., x_k$ be a group of secret keys. According to the DDH assumption, for each $i \in [k]$, the following two distributions are indistinguishable,

$$\begin{pmatrix} g_i & h_{i,1} & ... & h_{i,Q} \\ g_i^{x_i} & h_{i,1}^{x_i} & ... & h_{i,Q}^{x_i} \end{pmatrix} \text{ and } \begin{pmatrix} g_i & h_{i,1} & ... & h_{i,Q} \\ g_i^{x_i} & u_1 & ... & u_Q \end{pmatrix},$$

where $u_1, ..., u_Q \xleftarrow{\$} \mathbb{G}$. The pseudorandomness of the $(k+1)$-AIS scheme follows consequently by the hybrid arguement.

Collision resistance of secret key. Under the $(k+1)$-FindRep assumption, given $(\mathbb{G}, q, g_1, ..., g_{k+1})$, it is infeasible to find $(x_1, ..., x_{k+1}) \neq (x'_1, ..., x'_{k+1})$ such that $\widetilde{pk} = \prod_{i\in[k+1]} g_i^{x_i} = \prod_{i\in[k+1]} g_i^{x'_i}$.

Uniformity. Fixed $x_1, ..., x_k \in \mathbb{Z}_q$, if $x_{k+1} \xleftarrow{\$} \mathbb{Z}_q$, then $\widetilde{pk} = \prod_{i\in[k]} g_i^{x_i} \cdot g_{k+1}^{x_{k+1}}$ distributes uniformly over $\mathbb{G}$.

Downward compatibility. This is straightforward.

Homomorphism on coefficient space $\mathbb{Z}_q^*$ and infeasiblilty of kernel. Consider a group of coefficients $a_1, ..., a_k \in \mathbb{Z}_q^*$. Let $(\widetilde{cmt}, ch, rsp = (rsp_1, ..., rsp_k))$ be a transcript with $\widetilde{cmt} = \prod_{i\in[k]} g_i^{r_i}$. Define $\widetilde{epk} = \prod_{i\in[k]} pk_i^{a_i}$, and $\widetilde{ecmt} = \prod_{i\in[k]} g_i^{a_i r_i}$. Then the coefficient-verification algorithm $\mathsf{CoeVer}(pp, \widetilde{epk}, (\widetilde{ecmt}, ch, rsp), a_1, ..., a_k)$ returns whether $\widetilde{ecmt} = \prod_{i\in[k]} g_i^{rsp_i a_i} \cdot \widetilde{epk}^{ch}$. Meanwhile, according to the k-FindRep assumption, given k randomly sampled public keys X_i, it is hard to find $(a_1, ..., a_k) \in (\mathbb{Z}_q^*)^k$ such that $0_{\mathbb{G}} = \prod_{i\in[k]} X_i^{a_i}$.

Moreover, given coefficients $a_1, ..., a_k \in \mathbb{Z}_q^*$ and $\widetilde{epk} = \prod_{i\in[k]} (g_i^{x_i})^{a_i}$, the extended simulator outputs $(\widetilde{ecmt}, ch, rsp = (rsp_1, ..., rsp_k))$ as follows.

1. $\mathsf{Sim}_1()$ samples $rsp_i \xleftarrow{\$} \mathbb{Z}_q$ for $i \in [k]$ and outputs $rsp = (rsp_1, ..., rsp_k)$.
2. $\mathsf{Sim}_2(pp, \widetilde{epk}, ch, rsp, a_1, ..., a_k)$ outputs $\widetilde{ecmt} := \prod_{i\in[k]} (g_i^{rsp_i})^{a_i} \cdot \widetilde{epk}^{ch}$.

□

6.4 k-LRS from Aggregatable Identification Schemes

Let N be the total number of users and $k \in \mathbb{N}^+$ be the upper bound of signing times per user with unlinkability. Let AIS be a downward campatible $(k+1)$-AIS with parameter space $\mathcal{PP}$, key space $\mathcal{SK} \times \mathcal{PK}$, and extended secret key space $\ddot{\mathcal{SK}} \supseteq \mathcal{SK}$, and $\mathcal{PP}$ defines the commitment-challenge-response space $\mathcal{CMT} \times \mathcal{CH} \times \mathcal{RSP}$ for the corresponding identification protocol. Furthermore, AIS has homomorphism on coefficient space $\mathbb{F}$. Let $\mathcal{H}_0 : \{0,1\}^* \to \mathcal{PP}^k$ and $\mathcal{H} : \{0,1\}^* \to \mathcal{CH}$ be two hash functions. Our k-LRS scheme LRS is constructed as follows.

- Setup: Invoke $pp_i \leftarrow \mathsf{AIS.Setup}(1^\lambda)$ for $i \in [k+1]$. Return the public parameter $pp := (pp_1, ..., pp_{k+1})$.
- Key Generation: Let $(sk_i, pk_i) \leftarrow \mathsf{AIS.Gen}(pp_i)$ for $i \in [k+1]$, and $\widetilde{pk} \leftarrow \mathsf{AIS.AggrPK}(pk_1, ..., pk_{k+1})$. Return the secret key $\mathsf{sk} := (sk_1, ..., sk_{k+1})$ and the public key $\mathsf{pk} := \widetilde{pk}$.
- Sign: Let $(\mathsf{pk}^{(1)}, ..., \mathsf{pk}^{(n)})$ be the public keys of n users in the ring. W.l.o.g., we assume the signer's index is 1. Parse $\mathsf{sk}^{(1)} = (sk_1^{(1)}, ..., sk_{k+1}^{(1)})$. To sign on message msg under event label e, the signer generates the signature as follows.
  1. Let $\mathcal{H}_0(\mathsf{e}) = epp = (epp_1, ..., epp_k) \in \mathcal{PP}^k$. For $i \in [k]$, compute $epk_i := f_{epp_i}(sk_i^{(1)})$.
  2. Sample $a_1, ..., a_k \xleftarrow{\$} \mathbb{F}$, and compute $\mathsf{epk} \leftarrow \mathsf{AIS.AggrPK}(a_1 \cdot epk_1, ..., a_k \cdot epk_k)$.
  3. For $i \in [k+1]$, sample randomness ρ_i at random. Compute $cmt_i^{(1)} \leftarrow \mathsf{AIS.Commit}(pp_i; \rho_i)$ for $i \in [k+1]$, and $ecmt_i^{(1)} \leftarrow \mathsf{AIS.Commit}(epp_i; a_i \cdot \rho_i)$ for $i \in [k]$. Aggregate these commitments by $\widetilde{cmt}^{(1)} \leftarrow \mathsf{AIS.AggrCMT}(cmt_1^{(1)}, ..., cmt_{k+1}^{(1)})$ and $\widetilde{ecmt}^{(1)} \leftarrow \mathsf{AIS.AggrCMT}(ecmt_1^{(1)}, ..., ecmt_k^{(1)})$.
  4. For $j = 2, 3, ..., n$: Let $ch^{(j)} := \mathcal{H}(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, \{a_i\}_{i\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e})$, and invoke the simulators to generate the transcripts. I.e.,
     $rsp^{(j)} = (rsp_1^{(j)}, ..., rsp_k^{(j)}, rsp_{k+1}^{(j)}) \leftarrow \mathsf{AIS.Sim}_1(ch^{(j)})$,
     $\widetilde{cmt}^{(j)} \leftarrow \mathsf{AIS.Sim}_2(pp, \mathsf{pk}^{(j)}, ch^{(j)}, (rsp_1^{(j)}, ..., rsp_{k+1}^{(j)}))$,
     $\widetilde{ecmt}^{(j)} \leftarrow \mathsf{AIS.Sim}_2(epp, \mathsf{epk}, ch^{(j)}, (rsp_1^{(j)}, ..., rsp_k^{(j)}), a_1, ..., a_k)$.
  5. Let $\mathcal{H}(\widetilde{cmt}^n, \widetilde{ecmt}^n, \{a_j\}_{j\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e}) = ch^{(1)}$.
     Compute $rsp^{(1)} \leftarrow \mathsf{AIS.Response}((sk_1^{(1)}, ..., sk_{k+1}^{(1)}), ch^{(1)}, (\rho_1, ..., \rho_{k+1}))$.
  6. Return $\sigma := (a_1, ..., a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$.
- Verify: For $j \in [n]$, compute

$$ch^{(j)} = \mathcal{H}(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, \{a_j\}_{j\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e}).$$

  Let $epp = \mathcal{H}_0(\mathsf{e})$. If for all $j \in [n]$,

- $\mathsf{AIS.Ver}(pp, \mathsf{pk}^{(j)}, \widetilde{cmt}^{(j)}, ch^{(j)}, rsp^{(j)}) = 1$, and
- $\mathsf{AIS.CoeVer}(epp, \mathsf{epk}, (\widetilde{ecmt}^{(j)}, ch^{(j)}, rsp^{(j)}_{|k}), a_1, ..., a_k) = 1$,

then output 1. Otherwise, output 0. Here $rsp^{(j)}_{|k}$ denotes the first k parts of $rsp^{(j)}$.

– Link: Let $\{\sigma_\ell = (a_{\ell,1}, ..., a_{\ell,k}, \mathsf{epk}_\ell, ...)\}_{\ell\in[k+1]}$ be $(k+1)$ different signatures for the same event e. If there is a solution of random variables $(epk_1, ..., epk_k) \in \mathcal{PK}^k$ for the following linear equation system, then return 1 (linked). Otherwise, return 0 (unlinked).

$$\begin{cases} a_{1,1} \cdot epk_1 + a_{1,2} \cdot epk_2 + ... + a_{1,k} \cdot epk_k = \mathsf{epk}_1, \\ a_{2,1} \cdot epk_1 + a_{2,2} \cdot epk_2 + ... + a_{2,k} \cdot epk_k = \mathsf{epk}_2, \\ ... \\ a_{k+1,1} \cdot epk_1 + a_{k+1,2} \cdot epk_2 + ... + a_{k+1,k} \cdot epk_k = \mathsf{epk}_{k+1}. \end{cases} \tag{7}$$

Correctness of verification. The correctness of verification follows from the correctness of AIS and its homomorphism.

In more details, let $\sigma = (a_1, ..., a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$ be a signature from user P_1 with $\mathsf{sk}^{(1)} = (sk_1^{(1)}, ..., sk_k^{(1)}, sk_{k+1}^{(1)})$ and $\mathsf{pk}^{(1)}$, where $\mathsf{pk}^{(1)} \leftarrow \mathsf{AIS.AggrPK}(pk_1^{(1)}, ..., pk_{k+1}^{(1)})$.

– Verification on user P_1 (the signer). Let $(\rho_1, ..., \rho_k, \rho_{k+1})$ be the randomness used in the commitment for user 1, and $rsp_i^{(1)} \leftarrow \mathsf{Response}(sk_i^{(1)}, ch^{(1)}, \rho_i^{(1)})$ for all $i \in [k+1]$. Note that $\widetilde{cmt}^{(1)} \leftarrow \mathsf{AggrCMT}(\mathsf{Commit}(pp_1; \rho_1), ..., \mathsf{Commit}(pp_{k+1}; \rho_{k+1}))$. According to the correctness of AIS, with overwhelming probability $(\widetilde{cmt}, ch, rsp^{(1)} = (rsp_1^{(1)}, ..., rsp_{k+1}^{(1)}))$ is a valid transcript under $pp = (pp_1, ..., pp_{k+1})$ and $\mathsf{pk}^{(1)}$, i.e., $\mathsf{AIS.Ver}(pp, \mathsf{pk}^{(1)}, \widetilde{cmt}^{(1)}, ch^{(1)}, rsp^{(1)}) = 1$.
On the other hand, the event-related parameter $epp = (epp_1, ..., epp_k)$ maps the first k secret keys $(sk_1^{(1)}, ..., sk_k^{(1)})$ into a group of *event-related public keys* $(epk_1, ..., epk_k)$. Due to the homomorphism of AIS, introducing the coefficients $a_1, ..., a_k$ does not affect the correctness of verification. Namely,

$$\mathsf{AIS.CoeVer}(epp, \mathsf{epk}, (\widetilde{ecmt}^{(1)}, ch^{(1)}, rsp^{(1)}_{|k}), a_1, ..., a_k)$$

will return 1 if

$$\mathsf{AIS.Ver}(epp, \mathsf{AggrPK}(epk_1, ..., epk_k), (\mathsf{AggrCMT}(\mathsf{Commit}(epp_1; \rho_1), ..., \mathsf{Commit}(epp_k; \rho_k)), ch^{(1)}, rsp^{(1)}_{|k}))$$

returns 1, which is guaranteed by the correctness of AIS. Recall that in AIS, the response is computed from the secret key, the challenge, and the randomness ρ, and it is independent of the public parameter.

– Verification on other users. According to the simulatability of AIS, the simulated transcript distributes statistically close to the transcript outputted from an honest execution of $\langle \mathcal{P}, \mathcal{V} \rangle$. Therefore, for all $j \in [n] \setminus \{1\}$, with overwhelming probability it holds that

$$\mathsf{AIS.Ver}(pp, \mathsf{pk}^{(j)}, \widetilde{cmt}^{(j)}, ch^{(j)}, rsp^{(j)}) = 1,$$

and

$$\mathsf{AIS.CoeVer}(epp, \mathsf{epk}, (ecmt^{(j)}, ch^{(j)}, rsp^{(j)}_{|k}), a_1, ..., a_k) = 1.$$

Correctness of linkability. Consider $k+1$ different signatures $\{\sigma_\ell = (a_{\ell,1}, ..., a_{\ell,k}, \mathsf{epk}_\ell, ...)\}_{\ell\in[k+1]}$ under the same event e. Let $\mathcal{H}_0(\mathsf{e}) = (epp_1, ..., epp_k)$.

– If all these signatures are signed by the same signer holding secret key $\mathsf{sk} = (sk_1, ..., sk_k, sk_{k+1})$, then its event-related public keys are $(epk_1, ..., epk_k) = (f_{epp_1}(sk_1), ..., f_{epp_k}(sk_k))$. According to the signing algorithm, for all $\ell \in [k+1]$, it holds that

$$\sum_{i\in[k]} a_{\ell,i} \cdot epk_i = \mathsf{epk}_\ell.$$

Therefore, $\mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1})$ will return 1.

– On the other hand, assume all these signatures are signed by two different users holding secret keys $\mathsf{sk}^{(1)} = (sk_1^{(1)}, ..., sk_k^{(1)}, sk_{k+1}^{(1)})$ and $\mathsf{sk}^{(2)} = (sk_1^{(2)}, ..., sk_k^{(2)}, sk_{k+1}^{(2)})$, respectively. W.l.o.g., we assume the first k signatures are signed using $\mathsf{sk}^{(1)}$ and the last signature σ_{k+1} is signed using $\mathsf{sk}^{(2)}$. With overwhelming probability $(a_{\ell,i})_{\ell,i\in[k]}$ forms an invertable matrix. And we can computed the shifted public keys $(epk_1^{(1)}, ..., epk_k^{(1)}) = (f_{epp_1}(sk_1^1), ..., f_{epp_k}(sk_k^{(1)}))$ of user 1, from $(a_{\ell,i})_{\ell,i\in[k]}$ and $(\mathsf{epk}_1, ..., \mathsf{epk}_k)$. Meanwhile, we have $\sum_{i\in[k]} a_{k+1,i} \cdot epk_i^{(2)} = \mathsf{epk}_{k+1}$, where $(epk_1^{(1)}, ..., epk_k^{(1)}) = (f_{epp_1}(sk_1^{(1)}), ..., f_{epp_k}(sk_k^{(1)}))$ are the event-related public keys of user 2. If $\mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1})$ returns 1, then

$$\sum_{i\in[k]} a_{k+1,i} \cdot epk_i^{(2)} = \mathsf{epk}_{k+1} = \sum_{i\in[k]} a_{k+1,i} \cdot epk_i^{(1)}.$$

In our scheme, *epp* is randomly sampled from $\mathcal{PP}^k$ since $\mathcal{H}_0(\cdot)$ is modeled as a random oracle, and $\mathsf{sk}^{(1)}$ and $\mathsf{sk}^{(2)}$ are sampled independently. Due to the infeasibility of kernel property, the above equation holds with a negligible probability, as otherwise we find a group of coefficients $(a_{k+1,1}, ..., a_{k+1,k})$ such that $\sum_{i\in[k]} a_{k+1,i} \cdot (epk_i^{(2)} - epk_i^{(1)}) = 0_{\mathcal{PK}}$. Namely, if $(\sigma_1, ..., \sigma_{k+1})$ are signed by at least two different users, then with overwhelming probability $\mathsf{Link}(\mathsf{e}, \sigma_1, ..., \sigma_{k+1})$ will return 0, which concludes the proof of correctness.

Theorem 10 (Security of k-LRS). *If* AIS *is a* $(k+1)$*-AIS with downward compatibility and homomorphism on* $\mathbb{F}$*, then the* k*-LRS scheme above is secure, i.e.t, it has unforgeability, unconditional or computational anonymity, linkability, and non-slanderability.*

We defer Appendix D.2 for the security proof.

Acknowledgments. We would like to thank the anonymous reviewers of EUROCRYPT 2025 and CRYPTO 2025 for their insightful feedback and constructive suggestions, especially for highlighting the flaw in the unconditional security argument and suggesting approaches to improve the efficiency of voting.

References

[ACST06] Au, M.H., Chow, S.S.M., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. In: Atzeni, A.S., Lioy, A. (eds.) Public Key Infrastructure, Third European PKI Workshop: Theory and Practice, EuroPKI 2006, Lecture Notes in Computer Science, vol. 4043, pp. 101–115. Springer (2006)

[Adi08] Adida, B.: Helios: Web-based open-audit voting. In: van Oorschot, P.C. (ed.) Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA, pp. 335–348. USENIX Association (2008)

[AOS02] Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: Zheng, Y. (ed.) Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science, vol. 2501, pp. 415–432. Springer (2002)

[AOZZ15] Alwen, J., Ostrovsky, R., Zhou, H.S., Zikas, V.: Incoercible multi-party computation and universally composable receipt-free voting. In: Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II 35, pp. 763–780. Springer (2015)

[BBG+22] Balla, D., Behrouz, P., Grontas, P., Pagourtzis, A., Spyrakou, M., Vrettos, G.: Designated-verifier linkable ring signatures with unconditional anonymity. In: Poulakis, D., Rahonis, G. (eds.) Algebraic Informatics - 9th International Conference, CAI 2022, Lecture Notes in Computer Science, vol. 13706, pp. 55–68. Springer (2022)

[BEHG20] Boneh, D., Eskandarian, S., Hanzlik, L., Greco, N.: Single secret leader election. In: Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, pp. 12–24 (2020)

[BEHM22] Bootle, J., Elkhiyaoui, K., Hesse, J., Manevich, Y.: Dualdory: logarithmic-verifier linkable ring signatures through preprocessing. In: European Symposium on Research in Computer Security, pp. 427–446. Springer (2022)

[BF78] Brams, S., Fishburn, P.: Approval Voting. American Political Science Review vol. 72(3), pp. 831–847 (1978)

[BGK+21] Behrouz, P., Grontas, P., Konstantakatos, V., Pagourtzis, A., Spyrakou, M.: Designated-verifier linkable ring signatures. In: Park, J.H., Seo, S. (eds.) Information Security and Cryptology - ICISC 2021 - 24th International Conference, Seoul, South Korea, December 1-3, 2021, Revised Selected Papers, Lecture Notes in Computer Science, vol. 13218, pp. 51–70. Springer (2021)

[BH18] Boyen, X., Haines, T.: Forward-secure linkable ring signatures from bilinear maps. Cryptogr. vol. 2(4), p. 35 (2018)

[BHM08] Backes, M., Hritcu, C., Maffei, M.: Automated verification of remote electronic voting protocols in the applied pi-calculus. In: Proceedings of the 21st IEEE Computer Security Foundations Symposium, CSF 2008, Pittsburgh, Pennsylvania, USA, 23-25 June 2008, pp. 195–209. IEEE Computer Society (2008)

[BKP20] Beullens, W., Katsumata, S., Pintore, F.: Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Lecture Notes in Computer Science, vol. 12492, pp. 464–492. Springer (2020)

[BL16] Bultel, X., Lafourcade, P.: k-times full traceable ring signature. In: 11th International Conference on Availability, Reliability and Security, ARES 2016, pp. 39–48. IEEE Computer Society (2016)

[BN06] Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, pp. 390–399. ACM (2006)

[BPW23] Boneh, D., Partap, A., Waters, B.: Accountable multi-signatures with constant size public keys. Cryptology ePrint Archive (2023)

[Bra93] Brands, S.: Untraceable off-line cash in wallets with observers (extended abstract). In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 302–318. Springer (1993)

[BT94a] Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, pp. 544–553 (1994)

[BT94b] Benaloh, J.C., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: Leighton, F.T., Goodrich, M.T. (eds.) Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada, pp. 544–553. ACM (1994)

[BY86] Benaloh, J.C., Yung, M.: Distributing the power of a government to enhance the privacy of voters. In: Proceedings of the fifth annual ACM symposium on Principles of distributed computing, pp. 52–62 (1986)

[CDS94] Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, Lecture Notes in Computer Science, vol. 839, pp. 174–187. Springer (1994)

[CEL+23] Chow, S.S., Egger, C., Lai, R.W., Ronge, V., Woo, I.K.: On sustainable ring-based anonymous systems. In: 2023 IEEE 36th Computer Security Foundations Symposium (CSF), pp. 568–583. IEEE (2023)

[CGS97] Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. European transactions on Telecommunications vol. 8(5), pp. 481–490 (1997)

[CH91] Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT '91, Lecture Notes in Computer Science, vol. 547, pp. 257–265. Springer (1991)

[Cha81] Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM vol. 24(2), pp. 84–90 (1981)

[CHK+06] Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., Meyerovich, M.: How to win the clonewars: efficient periodic n-times anonymous authentication. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) CCS 2006, pp. 201–210. ACM (2006)

[CHL05] Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Cramer, R. (ed.) EUROCRYPT 2005, vol. 3494, pp. 302–321. Springer (2005)

[CLW08] Chow, S.S.M., Liu, J.K., Wong, D.S.: Robust receipt-free election system with ballot secrecy and verifiability. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2008. The Internet Society (2008)

[CLZ23] Choi, W., Liu, X., Zikas, V.: Blockchain governance via sharp anonymous multisignatures. IACR Cryptol. ePrint Arch. p. 1881 (2023)

[CZZ+16] Chondros, N., Zhang, B., Zacharias, T., Diamantopoulos, P., Maneas, S., Patsonakis, C., Delis, A., Kiayias, A., Roussopoulos, M.: D-DEMOS: A distributed, end-to-end verifiable, internet voting system. In: 36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016, pp. 711–720. IEEE Computer Society (2016)

[DKNS04] Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology - EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, pp. 609–626. Springer (2004)

[DKR06] Delaune, S., Kremer, S., Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: 19th IEEE Computer Security Foundations Workshop, (CSFW-19 2006), 5-7 July 2006, Venice, Italy, pp. 28–42. IEEE Computer Society (2006)

[DKR10] Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols: A taster. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections, New Directions in Electronic Voting, Lecture Notes in Computer Science, vol. 6000, pp. 289–309. Springer (2010)

[DV09] Dallot, L., Vergnaud, D.: Provably secure code-based threshold ring signatures. In: Parker, M.G. (ed.) Cryptography and Coding 2009, Lecture Notes in Computer Science, vol. 5921, pp. 222–235. Springer (2009)

[EHK+17] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for diffie–hellman assumptions. Journal of cryptology vol. 30, pp. 242–288 (2017)

[FS86] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86, Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer (1986)

[FS07] Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) Public Key Cryptography - PKC 2007, Lecture Notes in Computer Science, vol. 4450, pp. 181–200. Springer (2007)

[GHKW16] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly cca-secure encryption without pairings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 1–27. Springer (2016)

[GMS21] Gersbach, H., Mamageishvili, A., Schneider, M.: Vote delegation and misbehavior. In: Caragiannis, I., Hansen, K.A. (eds.) SAGT 2021, Lecture Notes in Computer Science, vol. 12885, p. 411. Springer (2021)

[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A.V. (ed.) Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 218–229. ACM (1987)

[GPS25] Grontas, P., Pagourtzis, A., Spyrakou, M.: Voting with coercion resistance and everlasting privacy using linkable ring signatures. Cryptology ePrint Archive (2025)

[Gro03] Groth, J.: A verifiable secret shuffle of homomorphic encryptions. In: Desmedt, Y. (ed.) Public Key Cryptography - PKC 2003, Lecture Notes in Computer Science, vol. 2567, pp. 145–160. Springer (2003)

[Har25] Hara, K.: A linkable ring signature scheme with unconditional anonymity in the standard model. Theoretical Computer Science p. 115093 (2025)

[HC24] Hui, X., Chau, S.C.K.: Llring: logarithmic linkable ring signatures with transparent setup. In: European Symposium on Research in Computer Security, pp. 299–319. Springer (2024)

[HKSS22] Haque, A., Krenn, S., Slamanig, D., Striecks, C.: Logarithmic-size (linkable) threshold ring signatures in the plain model. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) Public-Key Cryptography - PKC 2022, Lecture Notes in Computer Science, vol. 13178, pp. 437–467. Springer (2022)

[HS00] Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 539–556. Springer (2000)

[HS12] Heather, J., Schneider, S.A.: A formal framework for modelling coercion resistance and receipt freeness. In: Giannakopoulou, D., Méry, D. (eds.) FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7436, pp. 217–231. Springer (2012)

[IKOS06] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography from anonymity. In: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings, pp. 239–248. IEEE Computer Society (2006)

[JCJ10] Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections, New Directions in Electronic Voting, Lecture Notes in Computer Science, vol. 6000, pp. 37–63. Springer (2010)

[JP10] Jonker, H., Pieters, W.: Anonymity in voting revisited. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections, New Directions in Electronic Voting, Lecture Notes in Computer Science, vol. 6000, pp. 216–230. Springer (2010)

[JV06] Jonker, H.L., de Vink, E.P.: Formalising receipt-freeness. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) Information Security, 9th International Conference, ISC 2006, Samos Island, Greece, August 30 - September 2, 2006, Proceedings, Lecture Notes in Computer Science, vol. 4176, pp. 476–488. Springer (2006)

[KAPS20] Khan, N., Ahmad, T., Patel, A., State, R.: Blockchain governance: An overview and prediction of optimal strategies using nash equilibrium. CoRR vol. abs/2003.09241 (2020)

[KL22] Kiayias, A., Lazos, P.: Sok: Blockchain governance. In: Herlihy, M., Narula, N. (eds.) AFT 2022, Cambridge, MA, USA, September 19-21, 2022, pp. 61–73. ACM (2022)

[KT09] Küsters, R., Truderung, T.: An epistemic approach to coercion-resistance for electronic voting protocols. In: 30th IEEE Symposium on Security and Privacy (SP 2009), 17-20 May 2009, Oakland, California, USA, pp. 251–266. IEEE Computer Society (2009)

[KTV11] Küsters, R., Truderung, T., Vogt, A.: Verifiability, privacy, and coercion-resistance: New insights from a case study. In: 32nd IEEE Symposium on Security and Privacy, SP 2011, 22-25 May 2011, Berkeley, California, USA, pp. 538–553. IEEE Computer Society (2011)

[KTV12] Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion resistance and its applications. Journal of Computer Security vol. 20(6), pp. 709–764 (2012)

[KZZ15a] Kiayias, A., Zacharias, T., Zhang, B.: DEMOS-2: scalable E2E verifiable elections without random oracles. In: Ray, I., Li, N., Kruegel, C. (eds.) Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015, pp. 352–363. ACM (2015)

[KZZ15b] Kiayias, A., Zacharias, T., Zhang, B.: End-to-end verifiable elections in the standard model. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9057, pp. 468–498. Springer (2015)

[LASZ14] Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Trans. Knowl. Data Eng. vol. 26(1), pp. 157–165 (2014)

[LAZ19] Lu, X., Au, M.H., Zhang, Z.: Raptor: A practical lattice-based (linkable) ring signature. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 2019, Lecture Notes in Computer Science, vol. 11464, pp. 110–130. Springer (2019)

[LJP10] Langer, L., Jonker, H., Pieters, W.: Anonymity and verifiability in voting: Understanding (un)linkability. In: Soriano, M., Qing, S., López, J. (eds.) Information and Communications Security - 12th International Conference, ICICS 2010, Barcelona, Spain, December 15-17, 2010. Proceedings, Lecture Notes in Computer Science, vol. 6476, pp. 296–310. Springer (2010)

[LJW+19] Lyu, J., Jiang, Z.L., Wang, X., Nong, Z., Au, M.H., Fang, J.: A secure decentralized trustless e-voting system based on smart contract. In: 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, 2019, pp. 570–577. IEEE (2019)

[LLXZ20] Liu, A., Lu, Y., Xia, L., Zikas, V.: How private are commonly-used voting rules? In: Adams, R.P., Gogate, V. (eds.) Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020, Proceedings of Machine Learning Research, vol. 124, pp. 629–638. AUAI Press (2020)

[LQT20] Lueks, W., Querejeta-Azurmendi, I., Troncoso, C.: Voteagain: A scalable coercion-resistant voting system. CoRR vol. abs/2005.11189 (2020)

[LRR+19] Lai, R.W., Ronge, V., Ruffing, T., Schröder, D., Thyagarajan, S.A.K., Wang, J.: Omniring: Scaling private payments without trusted setup. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 31–48 (2019)

[LW05] Liu, J.K., Wong, D.S.: Linkable ring signatures: Security models and new schemes. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) Computational Science and Its Applications - ICCSA 2005, Lecture Notes in Computer Science, vol. 3481, pp. 614–623. Springer (2005)

[LW06] Liu, J.K., Wong, D.S.: Enhanced security models and a generic construction approach for linkable ring signature. International Journal of Foundations of Computer Science vol. 17(06), pp. 1403–1422 (2006)

[LWW04] Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Lecture Notes in Computer Science, vol. 3108, pp. 325–335. Springer (2004)

[Lyu08] Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) Public Key Cryptography - PKC 2008, Lecture Notes in Computer Science, vol. 4939, pp. 162–179. Springer (2008)

[MH96] Michels, M., Horster, P.: Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 125–132. Springer (1996)

[Nef01] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Reiter, M.K., Samarati, P. (eds.) CCS 2001, pp. 116–125. ACM (2001)

[Oka97] Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In: Christianson, B., Crispo, B., Lomas, T.M.A., Roe, M. (eds.) Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings, Lecture Notes in Computer Science, vol. 1361, pp. 25–35. Springer (1997)

[PBD05] Peng, K., Boyd, C., Dawson, E.: Simple and efficient shuffling with provable correctness and ZK privacy. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005, Lecture Notes in Computer Science, vol. 3621, pp. 188–204. Springer (2005)

[RAVR21] Russo, A., Anta, A.F., Vasco, M.I.G., Romano, S.P.: Chirotonia: A scalable and secure e-voting framework based on blockchains and linkable ring signatures. In: Xiang, Y., Wang, Z., Wang, H., Niemi, V. (eds.) 2021 IEEE International Conference on Blockchain, Blockchain 2021, pp. 417–424. IEEE (2021)

[RST01] Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, vol. 2248, pp. 552–565. Springer (2001)

[SALY17] Sun, S., Au, M.H., Liu, J.K., Yuen, T.H.: Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security - ESORICS 2017, Lecture Notes in Computer Science, vol. 10493, pp. 456–474. Springer (2017)

[Sch91] Schnorr, C.: Efficient signature generation by smart cards. J. Cryptol. vol. 4(3), pp. 161–174 (1991)

[Sho06] Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press (2006)

[SK95] Sako, K., Kilian, J.: Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth. In: Guillou, L.C., Quisquater, J. (eds.) Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques, Saint-Malo, France, May 21-25, 1995, Proceeding, Lecture Notes in Computer Science, vol. 921, pp. 393–403. Springer (1995)

[TFS04] Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication (extended abstract). In: Lee, P.J. (ed.) Advances in Cryptology - ASIACRYPT 2004, Lecture Notes in Computer Science, vol. 3329, pp. 308–322. Springer (2004)

[TS06] Teranishi, I., Sako, K.: k-times anonymous authentication with a constant proving cost. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) Public Key Cryptography - PKC 2006, Lecture Notes in Computer Science, vol. 3958, pp. 525–542. Springer (2006)

[TW05] Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: Deng, R.H., Bao, F., Pang, H., Zhou, J. (eds.) Information Security Practice and Experience, First International Conference, ISPEC 2005, Lecture Notes in Computer Science, vol. 3439, pp. 48–60. Springer (2005)

[TWC+04] Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004, Lecture Notes in Computer Science, vol. 3348, pp. 384–398. Springer (2004)

[VH21] Venugopalan, S., Homoliak, I.: Always on voting: A framework for repetitive voting on the blockchain. CoRR vol. abs/2107.10571 (2021)

[XLAZ24] Xue, Y., Lu, X., Au, M.H., Zhang, C.: Efficient linkable ring signatures: new framework and post-quantum instantiations. In: European Symposium on Research in Computer Security, pp. 435–456. Springer (2024)

[Yao82] Yao, A.C.: Protocols for secure computations (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science, pp. 160–164. IEEE Computer Society (1982)

[YNKM24] Yu, A., Nguyen, H.H., Kate, A., Maji, H.K.: Unconditional security using (random) anonymous bulletin board. IACR Cryptol. ePrint Arch. p. 101 (2024)

[YSL+20] Yuen, T.H., Sun, S., Liu, J.K., Au, M.H., Esgin, M.F., Zhang, Q., Gu, D.: Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security - 24th International Conference, FC 2020, Lecture Notes in Computer Science, vol. 12059, pp. 464–483. Springer (2020)

[YWM+22] Ye, Q., Wang, M., Meng, H., Xia, F., Yan, X.: Efficient linkable ring signature scheme over ntru lattice with unconditional anonymity. Computational Intelligence and Neuroscience vol. 2022(1), p. 8431874 (2022)

[ZK02] Zhang, F., Kim, K.: Id-based blind signature and ring signature from pairings. In: Zheng, Y. (ed.) Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science, vol. 2501, pp. 533–547. Springer (2002)

A Limitation of Liu et al.'s LRS Scheme

In this section we present a concrete attack showing that the LRS scheme in [LASZ14] does not achieve linkability (cf. Def. 4).

Let $\mathbb{G}$ be a cyclic group of order q, and g_1, g_2 are two random generators. The LRS scheme in [LASZ14] is shown as follows. Here $H_0 : \{0,1\}^* \to \mathbb{G}$, $H : \{0,1\}^* \to \mathbb{Z}_q^*$ are two hash functions which are modeled as random oracles.

$\underline{\mathsf{Setup}(1^\lambda)}$: Return $pp := (\mathbb{G}, q, g_1, g_2)$ $\underline{\mathsf{Gen}(pp)}$: $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$; $X := g_1^{x_1} g_2^{x_2}$ Return $(sk, pk) := ((x_1, x_2), X)$ $\underline{\mathsf{Ver}(R, PK_R, \mathsf{e}, \mathsf{msg}, \sigma)}$: $g_\mathsf{e} := H_0(\mathsf{e})$ $cmt' := g_1^{z_1} g_2^{z_2} \prod_{j\in[n]} (X^{(j)})^{c^{(j)}}$ $ecmt' := g_\mathsf{e}^{z_1} \prod_{j\in[n]} (pid)^{c^{(j)}}$ If $\sum_{j\in[n]} c^{(j)} = H(PK_R, pid, cmt', ecmt', \mathsf{e}, \mathsf{msg})$: return 1 Otherwise: reutnr 0	$\underline{\mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})}$: Parse $sk^{(\delta)} = (x_1, x_2)$, $PK_R = (X^{(1)}, X^{(2)}, ..., X^{(n)})$ $g_\mathsf{e} := H_0(\mathsf{e})$; $pid := g_\mathsf{e}^{x_1}$ $r_1, r_2, c^{(1)}, ..., c^{(\delta-1)}, c^{(\delta+1)}, ..., c^{(n)} \xleftarrow{\$} \mathbb{Z}_q$ $cmt := g_1^{r_1} g_2^{r_2} \prod_{j\in[n], j\neq\delta} (X^{(j)})^{c^{(j)}}$ $ecmt := g_\mathsf{e}^{r_1} \prod_{j\in[n], j\neq\delta} (pid)^{(c^{(j)})}$ $c_0 := H(PK_R, pid, cmt, ecmt, \mathsf{e}, \mathsf{msg})$ $c^{(\delta)} := c_0 - \sum_{j\in[n], j\neq\delta} c^{(j)}$ $z_1 := r_1 - c^{(\delta)} x_1$; $z_2 := r_2 - c^{(\delta)} x_2$ Return $\sigma := (pid, z_1, z_2, c^{(1)}, ..., c^{(n)})$ $\underline{\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2)}$: Parse $\sigma_1 = (pid_1, ...)$ and $\sigma_2 = (pid_2, ...)$ If $pid_1 = pid_2$: return 1 Otherwise: return 0

Fig. 7. The LRS scheme in [LASZ14].

Theorem 11. *The LRS scheme in Fig. 7 does not achieve linkability defined in Def. 4.*

Proof. To prove this theorem, we show that an adversary knowing two secret keys can generate three signatures which are pairwise unlinked.

Consider a ring of three signers with public key list $PK_R = (X^{(1)}, X^{(2)}, X^{(3)})$, and the first two secret keys $sk^{(1)} = (x_1^{(1)}, x_2^{(1)})$ and $sk^{(2)} = (x_1^{(2)}, x_2^{(2)})$ are known to the adversary $\mathcal{A}$. Fixed event e and $g_\mathsf{e} := H_0(\mathsf{e})$, the adversary $\mathcal{A}$ can first generate two unlinked signatures using $sk^{(1)}$ and $sk^{(2)}$, respectively. Now $\mathcal{A}$ aims to forge a signature $\sigma = (pid, z_1, z_2, c^{(1)}, ..., c^{(n)})$ such that $pid \neq g_\mathsf{e}^{x_1^{(1)}}$ and $pid \neq g_\mathsf{e}^{x_1^{(2)}}$.

$\mathcal{A}$ forges as follows.

1. Sample $r_e, r_1, r_2, c^{(3)} \xleftarrow{\$} \mathbb{Z}_q$, and $\tilde{x} \xleftarrow{\$} \mathbb{Z}_q \setminus \{x_1^{(1)}, x_1^{(2)}\}$.
2. Compute $pid := g_\mathsf{e}^{\tilde{x}}$, $cmt := g_1^{r_1} g_2^{r_2} (X^{(3)})^{c^{(3)}}$, and $ecmt := g_\mathsf{e}^{r_e} (pid)^{c^{(3)}}$.
3. Compute $c_0 := H(PK_R, pid, cmt, ecmt, \mathsf{e}, \mathsf{msg})$.
4. Compute $c^{(1)}, c^{(2)}, z_1, z_2$ satisfying the following four equations (with the knowledge of $(x_1^{(1)}, x_2^{(1)})$ and $(x_1^{(2)}, x_2^{(2)})$):
$$\begin{cases} c^{(1)} + c^{(2)} = c_0 - c^{(3)} \\ r_1 = z_1 + c^{(1)} x_1^{(1)} + c^{(2)} x_1^{(2)} \\ r_2 = z_2 + c^{(1)} x_2^{(1)} + c^{(2)} x_2^{(2)} \\ r_e = z_1 + \tilde{x}(c^{(1)} + c^{(2)}) \end{cases}$$
5. Return $\sigma = (pid, z_1, z_2, c^{(1)}, c^{(2)}, c^{(3)})$.

First, the four equations above imply that

$$cmt = g_1^{z_1} g_2^{z_2} (X^{(1)})^{c^{(1)}} (X^{(2)})^{c^{(2)}} (X^{(3)})^{c^{(3)}} \text{ and } ecmt = g_{\mathsf{e}}^{z_1} (g_{\mathsf{e}}^{\tilde{x}})^{c^{(1)}+c^{(2)}+c^{(3)}}.$$

Namely, σ is a valid forgery. Furthermore, σ is unlinked to the signatures generated from $sk^{(1)}$ and $sk^{(2)}$, since $pid \neq g_{\mathsf{e}}^{x_1^{(1)}}$ and $pid \neq g_{\mathsf{e}}^{x_1^{(2)}}$. Therefore, $\mathcal{A}$ successfully breaks the linkability of the scheme. □

B The Insecurity of Bultel and Lafourcade's Scheme

In this section we show that Bultel and Lafourcade's scheme [BL16] does not achieve traceability as the authors claimed.

We first quickly review the TRS scheme in [BL16]. Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_t$ be groups of prime order q, and g_1 is a generator of $\mathbb{G}_1$, g_2 is a generator of $\mathbb{G}_2$, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ is a non-degenerate bilinear map. Two hash functions are defined as $H_0 : \{0,1\}^* \to \mathbb{G}_1$, and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$.

The scheme involves a non-interactive zero-knowledge proof (NIZK) scheme to show that $(T_1, T_2, T_3, T_4, T_5)$ in the signature are well-formed, and the corresponding $(x^{(j)}, x_i^{(j)})$ is a secret key of some public key in PK_R. The authors cite the 1-out-of-n proof skill in [CDS94]. For simplicity, we omit the details here, and merge the "Match" algorithm and the "Trace" algorithm in [BL16] into one algorithm Trace here. We also omit the details of how to extend Trace to detect all signatures from one actual signer, since it is out of the insecurity identified below.

$\underline{\mathsf{Setup}(1^\lambda)}$:
Return $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, q, g_1, g_2, e, H_0, H_1)$

$\underline{\mathsf{Gen}(pp, k)}$:
$x, x_1, ..., x_k \xleftarrow{\$} \mathbb{Z}_q^*$
$X := g_1^x$; $X_1 := g_1^{x_1}, ..., X_k := g_1^{x_k}$
Return $(sk, pk) = ((x, x_1, ..., x_k), (X, X_1, ..., X_k))$

$\underline{\mathsf{Sign}(sk^{(j)}, PK_R, \mathsf{e}, \mathsf{msg}, i)}$: // User j invokes the i-th signing
Parse $sk^{(j)} = (x^{(j)}, x_1^{(j)}, ..., x_k^{(j)})$
$r \xleftarrow{\$} \mathbb{Z}_q^*$
$A := H_0(\mathsf{e}, 0)$; $B := H_0(\mathsf{e}, 1)$; $C := H_0(\mathsf{e}, 2)$; $W := H_0(\mathsf{e}, 3)$
$u := H_1(\mathsf{e}, \mathsf{msg}, 0, g_2^r)$; $v := H_1(\mathsf{e}, \mathsf{msg}, 1, g_2^r)$
$T_1 := A^{x_i^{(j)}}$; $T_2 := B^{x_i^{(j)}} \cdot g_1^{u \cdot x^{(j)}}$;
$T_3 := C^{x_i^{(j)}} \cdot W^{v \cdot x^{(j)}}$; $T_4 := g_2^r$
$T_5 := e(W, T_4)^{(x^{(j)})}$
$\pi \leftarrow \mathsf{NIZK.Prove}(T_1, T_2, T_3, T_4, T_5, PK_R)$
Return $\sigma := (T_1, T_2, T_3, T_4, T_5, \pi)$

$\underline{\mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma)}$:
Parse $\sigma = (T_1, T_2, T_3, T_4, T_5, \pi)$
Compute A, B, C, W, u, v as Sign
Return $\mathsf{NIZK.Ver}((T_1, T_2, T_3, T_4, T_5, PK_R), \pi)$

$\underline{\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2)}$:
Parse $\sigma_1 = (T_{1,1}, T_{1,2}, T_{1,3}, T_{1,4}, T_{1,5}, \pi_1)$
Parse $\sigma_2 = (T_{2,1}, T_{2,2}, T_{3,3}, T_{4,4}, T_{5,5}, \pi_1)$
If $T_{1,1} = T_{2,1}$: return 1
Otherwise: return 0

$\underline{\mathsf{Trace}(\mathsf{e}, \mathsf{msg}_1, \sigma_1, \mathsf{msg}_2, \sigma_2, \mathsf{msg}, \sigma)}$:
If $\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2) = 0$: return $\perp$
Parse $\sigma_1 = (T_{1,1}, T_{1,2}, T_{1,3}, T_{1,4}, T_{1,5}, \pi_1)$
Parse $\sigma_2 = (T_{2,1}, T_{2,2}, T_{3,3}, T_{4,4}, T_{5,5}, \pi_1)$
$u_1 := H_1(\mathsf{e}, \mathsf{msg}_1, 0, T_{1,4})$; $v_1 := H_1(\mathsf{e}, \mathsf{msg}_1, 1, T_{1,4})$
$u_2 := H_1(\mathsf{e}, \mathsf{msg}_2, 0, T_{2,4})$; $v_1 := H_1(\mathsf{e}, \mathsf{msg}_2, 1, T_{2,4})$
$id := (T_{1,2}/T_{2,2})^{(u_1-u_2)^{-1}}$
Return id // return an identity of the signer

Fig. 8. The TRS scheme in [BL16].

Correctness of Trace. The authors argued the correctness of Trace as follows.

$$id = \left(\frac{T_{1,2}}{T_{2,2}}\right)^{(u_1-u_2)^{-1}} = \left(\frac{B^{x_i^{(j)}} \cdot g_1^{u \cdot x^{(j)}}}{B^{x_i^{(j)}} \cdot g_1^{u \cdot x^{(j)}}}\right)^{(u_1-u_2)^{-1}} = g_1^{x^{(j)}},$$

where for $b \in \{0,1\}$, $u_b = H_1(\mathsf{e}, \mathsf{msg}_b, 0, g_2^{r_b})$, and r_b is the randomness used in signing. Given $g_1^{x^{(j)}}$, a part of the public key, the identity of the signer is exposed.

On the insecurity of Bultel's scheme. The correctness of trace heavily relies on $u_1 \neq u_2$. In case $H_1(\cdot)$ is modeled as a random oracle, this is statistically equal to $r_1 \neq r_2$. However, a malicious signer, who signs twice with the same count index $i \in [k]$, can always take the same randomness of r when signing, and hence make the trace algorithm fail. We emphasize that this would not result in a repeated signature, since there is randomness in generating a NIZK proof.

As a result, an adversary accessing to a secret key of a signer, is able to generate $k+1$ signatures w.r.t. the same event e and make the trace algorithm fail. Therefore, $\mathcal{A}$ breaks the securtity of traceability defined in [BL16].

C Impossiblity of Traceable Ring Signatures with Unconditional Anonymity

In this part we show that in traceable ring signatures [FS07] (an extension of LRS, of which the link algorithm will additionally output a public key when returning "linked"), it is impossible to achieve unconditional anonymity.

We first recall the definition of traceable ring signatures.

Definition 21 (Traceable ring signatures [FS07]). *A traceable ring signature (TRS) scheme is defined the same as a 1-LRS scheme (i.e., $k = 1$) in Def. 1, except that the algorithm* Link *is replaced with the following algorithm* Trace.

- $\{0,1\} \times (\mathcal{PK} \cup \{\perp\}) \leftarrow \mathsf{Trace}(\mathsf{e}, \sigma_1, \sigma_2)$. *The trace algorithm takes as input an event label* e *and two signatures σ_1, σ_2, and outputs a bit as well as a public key or a failure symbol $\perp$.*

Correctness of Trace. *Let $N \in \mathbb{N}^+$. For any $R_1, R_2 \subseteq [N]$ and $\delta_1 \in R_1, \delta_2 \in R_2$, any event* e *and messages* $\mathsf{msg}_1, \mathsf{msg}_2$, $\sigma_1 \leftarrow \mathsf{Sign}(PK_{R_1}, sk^{(\delta_1)}, \mathsf{e}, \mathsf{msg}_1); \sigma_2 \leftarrow \mathsf{Sign}(PK_{R_2}, sk^{(\delta_2)}, \mathsf{e}, \mathsf{msg}_2)$, *it holds with overwhelming probability that*

$$\mathsf{Trace}(\mathsf{e}, \sigma_1, \sigma_2) = \begin{cases} (0, \perp) & \text{if } \delta_1 \neq \delta_2, \\ (1, \perp) & \text{if } \delta_1 = \delta_2, \mathsf{msg}_1 = \mathsf{msg}_2, \\ (1, pk^{(\delta_1)}) & \text{if } \delta_1 = \delta_2, \mathsf{msg}_1 \neq \mathsf{msg}_2, \end{cases}$$

where $pp \leftarrow \mathsf{Setup}(1^\lambda, k = 1)$, $(sk^{(1)}, pk^{(1)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp)$, $\sigma_1 \leftarrow \mathsf{Sign}(PK_{R_1}, sk^{(\delta_1)}, \mathsf{e}, \mathsf{msg}_1)$, $\sigma_2 \leftarrow \mathsf{Sign}(PK_{R_2}, sk^{(\delta_2)}, \mathsf{e}, \mathsf{msg}_2)$, and the probability is taken over the choice of $\mathsf{Setup}, \mathsf{Gen}$, *and* Sign.

Theorem 12. *Traceable ring signatures cannot achieve unconditional anonymity (cf. Def. 3).*

Proof. We prove this theorem via a simple example. Consider a ring of two signers $R = \{\delta_0, \delta_1\}$. Let b be the secret bit in the unconditional anonymity experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k=1}(\lambda)$. After a query $\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}, \mathsf{msg}, \mathsf{msg}'')$ for random e and $\mathsf{msg}, \mathsf{msg}''$, the adversary $\mathcal{A}$ obtain two signatures $(\sigma^*, \cdot)$ where σ^* is a signature for msg from δ_b (we omit the second signature returned from the experiment). Now, $\mathcal{A}$'s goal is to decide the value of b.

From the argument in [LASZ14], we know for unconditionally traceable/link ring signatures, there must be more than one secret key corresponding to a public key. W.l.o.g., we assume that for a public key pk, there exists at most $\eta \in \mathbb{N}^+$ corresponding secret keys $sk_1, ..., sk_\eta$, and the key generation algorithm first samples a secret key randomly, and then computes the corresponding public key via some one-way function defined by pp.

In $\mathsf{Exp}^{anony}_{\mathcal{A},k=1}(\lambda)$, let $sk^{(0)}$, $sk^{(1)}$ be the secret keys of δ_0 and δ_1, respectively, and $pk^{(0)}$ and $pk^{(1)}$ be the corresponding public keys. Given $pk^{(0)}$, the all-powerful adversary searches all η possible secret keys $\{sk_1^{(0)}, ..., sk_\eta^{(0)}\}$ of $pk^{(0)}$, and then signs message $\mathsf{msg}' \neq \mathsf{msg}$ under event e with $sk_i^{(0)}$ $(i \in [\eta])$ to obtain a signature σ_i. If there exists $i \in [\eta]$ such that $\mathsf{Trace}(\mathsf{e}, \sigma^*, \sigma_i) = (1, pk^{(0)})$, then $\mathcal{A}$ outputs $b' = 0$. Otherwise $\mathcal{A}$ outputs $b' = 1$.

Now, we analyze the advantage of $\mathcal{A}$. First consider the case $b = 0$, i.e., σ^* is signed using δ_0's secret key $sk^{(0)}$. In this case, there must exists $s \in [\eta]$ such that $sk^{(0)} = sk_s^{(0)}$, and subsequently $\mathsf{Trace}(\mathsf{e}, \sigma^*, \sigma_s) = (1, pk^{(0)})$ according to the correctness of trace. Therefore, $\mathcal{A}$ will output $b' = 0$ with overwhelming probability.

Then we analyze the case $b = 1$. For all $i \in [\eta]$, $sk_i^{(0)}$ is a secret key corresponding to $pk^{(0)}$ and σ_i is a signature signed using $sk_i^{(0)}$. Recall that σ^* is signed using δ_1's secret key $sk^{(1)}$. Therefore, $\mathsf{Trace}(\mathsf{e}, \sigma^*, \sigma_i)$ will return $(0, \perp)$ due to the correctness of trace, and $\mathcal{A}$ will output $b' = 1$ with overwhelming probability.

Namely, $\mathcal{A}$'s advantage is almost $1/2$, showing that it is not unconditionally anonymous at all. □

Remark 11. In our k-LRS scheme, as well as Liu *et al.*'s LRS scheme [LASZ14], only a part of the secret key is revealed through the signature, and the public key remains uniformly distributed as long as the unrevealed part of the secret key is uniformly distributed. Consequently, for an information-theoretic adversary, the knowledge of the partial secret key (derived from the signature(s)) provides no advantage in tracing the corresponding public key since any two distinct public keys are equally likely to share the revealed portion as a partial secret key.

D Deferred Material in Construction

In this section we provide the SIS-based construction of $(k+1)$-AIS and the proof of Theorem 7.

D.1 SIS-based AIS

Definition 22 (The SIS assumption). *Let $n = n(\lambda), m, q$ be positive integers and β be a positive real. The short integer solution (SIS) assumption states that for any PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{sis}_{[n,m,q,\beta],\mathcal{A}}(\lambda) := \Pr[\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}; \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}) : \mathbf{A}\mathbf{x} = 0^n \wedge \mathbf{x} \neq 0^m \wedge ||\mathbf{x}|| \leq \beta]$$

is negligible in λ.

Lemma 1 (Leftover hash lemma [Lyu08]). *Let X be a subset of $\mathbb{Z}_q^m$. Then for all but a $2^{\frac{n\log q - \log|X|}{4}}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we have*

$$\Delta(\mathbf{A}\mathbf{x}, \mathbf{u}) \leq 2^{\frac{n\log q - \log|X|}{4}},$$

where $\mathbf{x} \xleftarrow{\$} X$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.

We recall the SIS-based identification scheme in [Lyu08] in Fig. 9. Here $pp = \mathbf{A}$ defines $\mathcal{SK} = \{0,1\}^m$, $\mathcal{PK} = \mathbb{Z}_q^n$, and the extended secret key space $\ddot{\mathcal{SK}} = \{\mathbf{x} \in \mathbb{Z}_q^m \mid ||\mathbf{x}|| \leq 10m^{1.5}\}$. SAFE is defined as $\{1, ..., 10m-1\}^m$. For the purpose of aggregation, in this work, we change the sampling of $\mathbf{r}^{[\![\iota]\!]}$ ($\iota \in [t]$) from $\{0, 1, ..., 10m-1\}^m$ instead of $\{0, 1, ..., 5m-1\}^m$ in [Lyu08]. This change leads to a scheme with a smaller correctness error, but on the other hand, it requires a stricter parameter due to the hardness of the underlying SIS problem.

Theorem 13. *Assume $m \geq 10$, $m = Q\lceil n \log n\rceil$ for some constant Q, and the $[n, m, q, \beta]$-SIS assumption holds for $\beta = 20m^{1.5}$, then the scheme in Fig. 9 is a secure identification scheme. More precisely, it has correctness error less that $2^{-t/14}$, one-wayness, $(tn\log q)$-min-entropy, $(1 - 2^{-t/6})$-special soundness, simulatability, and $Q/4$-pseudorandomness. Moreover, it has homomorphism on coefficient space $\mathbb{Z}_q^*$.*

Proof. The proof roughly follows the proof in [Lyu08].

Correctness. For $\iota \in [t]$, unless $c^{[\![\iota]\!]} = 1$ and $\mathbf{r}^{[\![\iota]\!]} + \mathbf{x} \notin \mathsf{SAFE}$, the response $\mathbf{z}^{[\![\iota]\!]}$ will always pass the verification and hence $d^{[\![\iota]\!]} = 1$. Therefore,

$$\begin{aligned}\Pr[d^{[\![\iota]\!]} = 1] &\geq \Pr[c^{[\![\iota]\!]} = 0] + \Pr[c^{[\![\iota]\!]} = 1]\Pr[\mathbf{r}^{[\![\iota]\!]} + \mathbf{x} \in \mathsf{SAFE} \mid c^{[\![\iota]\!]} = 1] \\ &= 1/2 + 1/2 \cdot (1 - 1/10m)^m \\ &\geq 0.95, \text{ for } m \geq 10.\end{aligned}$$

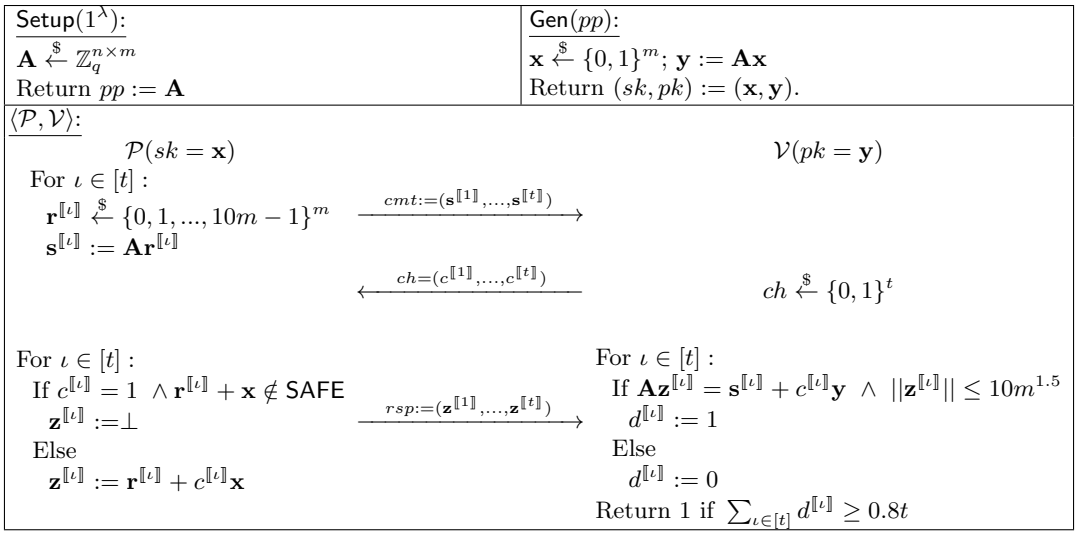

Fig. 9. The SIS-based identification scheme [Lyu08]

Let $sum = \sum_{\iota\in[t]} d^{[\![\iota]\!]}$. By Chernoff bound, we have

$$\text{correctness error} = \Pr[sum < 0.8t] = \Pr[sum < (0.95 - 0.15)t] \le e^{-2t(0.15)^2} \le 2^{-t/20}.$$

One-wayness. Let $pp = \mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $sk = \mathbf{x} \in \ddot{\mathcal{SK}} = \{\mathbf{x} \in \mathbb{Z}_q^m \mid ||\mathbf{x}|| \le 10m^{1.5}\}$. Define $f_{pp}(sk) = \mathbf{A}\mathbf{x}$. If an adversary $\mathcal{A}$ can break the one-wayness, then we can construct a reduction algorithm that breaks the SIS assumption as follows.

Given $\mathbf{A}$, the reduction algorithm first samples $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$, and then passes the public parameter and the public key $(\mathbf{A}, \mathbf{A}\mathbf{x})$ to $\mathcal{A}$. According to Lemma 8 in [Lyu08], with overwhelming probability there exists a distinct $\mathbf{x}' \in \{0,1\}^m \subset \ddot{\mathcal{SK}}$ such that $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$. And if $\mathcal{A}$ returns $\mathbf{x}' \neq \mathbf{x}$, which happens with probability at least $1/2$, then $(\mathbf{x} - \mathbf{x}')$ is a solution to the SIS($\mathbf{A}$) problem.

Min-entropy. According to Lemma 1, $\mathbf{A}\mathbf{r}^{[\![\iota]\!]}$ is statistically close to uniform distribution over $\mathbb{Z}_q^n$ if $\mathbf{r}^{[\![\iota]\!]}$ distributes uniformly over $\mathbb{Z}_q^m$. Therefore, the identification scheme has a min-entropy about $tn \log q$.

Special soundness. Let $((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, ..., c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$ and $((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]'}, ..., c^{[\![t]\!]'}), (\mathbf{z}^{[\![1]\!]'}, ..., \mathbf{z}^{[\![t]\!]'}))$ be two valid transcripts. First, if there exists $\iota^* \in [t]$ s.t. $c^{[\![\iota^*]\!]} \neq c^{[\![\iota^*]\!]'}$ and $\mathbf{z}^{[\![\iota^*]\!]}, \mathbf{z}^{[\![\iota^*]\!]'} \neq \perp$, then from the equation (w.l.o.g. we assume $c^{[\![\iota^*]\!]} - c^{[\![\iota^*]\!]'} = 1$)

$$\mathbf{A}(\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}) = \mathbf{y}$$

we can extract a solution $(\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'})$ to the SIS($\mathbf{A}$) problem, and $||\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}|| \le ||(\mathbf{z}^{[\![\iota^*]\!]})|| + ||\mathbf{z}^{[\![\iota^*]\!]'}|| \le 20m^{1.5}$.

Next we argue that the above case happens with probability at least $(1 - 2^{-t/6})$. Recall that $c^{[\![\iota]\!]}, c^{[\![\iota]\!]'} \xleftarrow{\$} \{0,1\}$ for $\iota \in [t]$. If $\sum_{\iota\in[t]} |c^{[\![\iota]\!]} - c^{[\![\iota]\!]'}| > 0.2$, then by the pigeonhole principle there must exists $\iota^* \in [t]$ s.t. $c^{[\![\iota^*]\!]} \neq c^{[\![\iota^*]\!]'}$ and $\mathbf{z}^{[\![\iota^*]\!]}, \mathbf{z}^{[\![\iota^*]\!]'} \neq \perp$. Due to the Chernoff bound, we have

$$\Pr[\sum_{\iota\in[t]} |c^{[\![\iota]\!]} - c^{[\![\iota]\!]'}| < 0.2t] \le e^{-2t(0.3)^2} < 2^{-t/6}.$$

Simulatability. [Lyu08] does not present a simulator for the zero-knowledge property (instead, [Lyu08] shows the scheme is witness indistinguishable). Here we present an efficient simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ as follows.

1. Let $ch = (c^{[\![1]\!]}, ..., c^{[\![t]\!]}) \in \{0,1\}^t$ be the challenge. For $\iota \in [t]$ and $\zeta \in [m]$, sample $b^{[\![\iota]\!]}[\zeta] \leftarrow \chi$, where χ is a biased binary distribution with 0-probability equal to $1/10m$.
2. For $\iota \in [t]$:
   (a) if $c^{[\![\iota]\!]} = 1$ and there exists $\zeta \in [m]$ s.t. $b^{[\![\iota]\!]}[\zeta] = 0$, then $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} := \perp$, and $\mathsf{Sim}_2(pp, pk, ch, rsp)$ outputs $\mathbf{s}^{[\![\iota]\!]} := \mathbf{A}\mathbf{r}^{[\![\iota]\!]}$, where $\mathbf{r}^{[\![\iota]\!]} := (\mathbf{r}^{[\![\iota]\!]}[1], ..., \mathbf{r}^{[\![\iota]\!]}[m])$, and $\mathbf{r}^{[\![\iota]\!]}[\zeta] := 0$ if $b^{[\![\iota]\!]}[\zeta] = 0$ and $\mathbf{r}^{[\![\iota]\!]}[\zeta] \xleftarrow{\$} \{1, ..., 10m-1\}$ if $b^{[\![\iota]\!]}[\zeta] = 1$ (that is, we always assume $\mathbf{x}[\zeta] = 0$ when simulating);
   (b) else if $c^{[\![\iota]\!]} = 1$, $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} \xleftarrow{\$} \{1, ..., 10m-1\}^m$ (the SAFE range), and $\mathsf{Sim}_2(pp, pk, ch, rsp)$ outputs $\mathbf{s}^{[\![\iota]\!]} := \mathbf{A}\mathbf{z}^{[\![\iota]\!]} - \mathbf{y}$;
   (c) else if $c^{[\![\iota]\!]} = 0$, $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} := \mathbf{r}^{[\![\iota]\!]}$, and $\mathsf{Sim}_2(pp, pk, ch, rsp)$ outputs $\mathbf{s}^{[\![\iota]\!]} := \mathbf{A}\mathbf{z}^{[\![\iota]\!]}$ with $\mathbf{r}^{[\![\iota]\!]} \xleftarrow{\$} \{0, ..., 10m-1\}^m$.
3. The simulated transcript is $(cmt = (\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), ch, rsp = (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$.

Analysis of Sim: First, the probability that $\mathbf{z}^{[\![\iota]\!]} = \perp$ in the simulation is equal to that in a real execution. For the ι-th execution ($\iota \in [t]$):

– Case (a): If the ζ-th element of $\mathbf{r}^{[\![\iota]\!]}[\zeta] + \mathbf{x}[\zeta]$ ($\zeta \in [m]$) is UNSAFE, then $\mathbf{r}^{[\![\iota]\!]}[\zeta] = 0$ if $\mathbf{x}[\zeta] = 0$, and $\mathbf{r}^{[\![\iota]\!]}[\zeta] = 10m-1$ if $\mathbf{x}[\ell] = 1$. If the ζ-th element ($\zeta \in [m]$) is SAFE, then $\mathbf{r}^{[\![\iota]\!]}[\zeta]$ is uniformly distributed in $\{1, ..., 10m-1\}$ if $\mathbf{x}[\zeta] = 0$ or $\{0, ..., 10m-2\}$ if $\mathbf{x}[\zeta] = 1$. With high probability, there are very few unsafe points among total m choices, which means that $\mathbf{r} \in Y \subseteq \{0, ..., 10m-1\}^m$ with $|Y|$ very large. Then, according to Lemma 1, $\mathbf{A}\mathbf{r}$ is statistically close to uniform vector $\mathbf{u} \in \mathbb{Z}_q^n$. That is, in the case of UNSAFE, the simulated transcript is statistically indistinguishable.
– Case (b): In the SAFE case, for all $\zeta \in [m]$, $\mathbf{r}^{[\![\iota]\!]}[\zeta] + \mathbf{x}[\zeta]$ is a random distribution over $\{1, ..., 10m-1\}$, no matter $\mathbf{x}[\zeta] = 0$ or $\mathbf{x}[\zeta] = 1$. Therefore, the simulation is perfect.
– Case (c): the simulation is perfect.

$Q/4$-pseudorandomness. Let $\mathbf{A}, \mathbf{A}_1, ..., \mathbf{A}_Q \in \mathbb{Z}_q^{n \times m}$ be two random matrices, and let $\mathbf{x} \in \{0,1\}^m$ be a secret key sampled uniformly. By applying Lemma 1 on the matrix $\widetilde{\mathbf{A}} := (\mathbf{A}^\top || \mathbf{A}_1{}^\top || ... || \mathbf{A}_Q{}^\top)^\top \in \mathbb{Z}_q^{Qn \times m}$, $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{A}}x)$ is statistically close to $(\widetilde{\mathbf{A}}, \mathbf{v})$, where $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^{Qn}$, which implies that the following two distributions are indistinguishable:

$$\begin{pmatrix} \mathbf{A} & \mathbf{A}\mathbf{x} \\ \mathbf{A}_1 & \mathbf{A}_1\mathbf{x} \\ \vdots & \vdots \\ \mathbf{A}_Q & \mathbf{A}_Q\mathbf{x} \end{pmatrix} \text{ and } \begin{pmatrix} \mathbf{A} & \mathbf{A}\mathbf{x} \\ \mathbf{A}_1 & \mathbf{u}_1 \\ \vdots & \vdots \\ \mathbf{A}_Q & \mathbf{u}_Q \end{pmatrix},$$

where $\mathbf{u}_1, ..., \mathbf{u}_Q \xleftarrow{\$} \mathbb{Z}_q^n$.[18]

Homomorphism on coefficient space $\mathbb{Z}_q^*$. Given $a \in \mathbb{Z}_q^*$ and a transcript $(cmt, ch, rsp) = ((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, ..., c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$ where $(\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}) = (\mathbf{A}\mathbf{r}^{[\![1]\!]}, ..., \mathbf{A}\mathbf{r}^{[\![t]\!]})$, the coefficient-verification algorithm
$\mathsf{CoeVer}(pp, epk, (ecmt, ch, rsp), a)$ works as follows. Here $epk = a \cdot pk = a\mathbf{y} \in \mathbb{Z}_q^n$, and $ecmt = (a\mathbf{s}^{[\![1]\!]}, ..., a\mathbf{s}^{[\![t]\!]}) = (\mathbf{A}(a\mathbf{r}^{[\![1]\!]}), ..., \mathbf{A}(a\mathbf{r}^{[\![t]\!]}))$.

1. For $\iota \in [t]$: if $a\mathbf{A}\mathbf{z}^{[\![\iota]\!]} = (a\mathbf{s}^{[\![\iota]\!]}) + c^{[\![\iota]\!]}(a\mathbf{y})$ and $||\mathbf{z}^{[\![\iota]\!]}|| \leq 10m^{1.5}$, then set $d^{[\![\iota]\!]} := 1$; otherwise set $d^{[\![\iota]\!]} := 0$.
2. Return 1 if $\sum_{i \in [t]} d^{[\![\iota]\!]} \geq 0.8t$ and 0 otherwise.

Given coefficient $a \in \mathbb{Z}_q^*$ and $epk = a \cdot pk = a\mathbf{y}$, the simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ above can be extended to output a simulated coefficient-transcript $(ecmt, ch, rsp)$ as follows.

[18] The pseudorandomness of the SIS-based scheme does not rely on any hardness assumption. However, it holds only for a constant Q, different from the DDH-based construction where Q can be any polynomial size. This results to a shortcoming of the SIS-based k-LRS scheme that the total number of events one can sign is bounded.

1. Let $ch = (c^{[\![1]\!]}, ..., c^{[\![t]\!]}) \in \{0,1\}^t$ be the challenge. For $\iota \in [t]$ and $\zeta \in [m]$, sample $b^{[\![\iota]\!]}[\zeta] \leftarrow \chi$, where χ is a biased binary distribution with 0-probability equal to $1/10m$.
2. For $\iota \in [t]$:
   (a) if $c^{[\![\iota]\!]} = 1$ and there exists $\zeta \in [m]$ s.t. $b^{[\![\iota]\!]}[\zeta] = 0$, then $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} := \perp$, and $\mathsf{Sim}_2(pp, pk, ch, rsp, a)$ outputs $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{A}\mathbf{r}^{[\![\iota]\!]}$, where $\mathbf{r}^{[\![\iota]\!]} := (\mathbf{r}^{[\![\iota]\!]}[1], ..., \mathbf{r}^{[\![\iota]\!]}[m])$, and $\mathbf{r}^{[\![\iota]\!]}[\zeta] := 0$ if $b^{[\![\iota]\!]}[\zeta] = 0$ and $\mathbf{r}^{[\![\iota]\!]}[\zeta] \xleftarrow{\$} \{1, ..., 10m-1\}$ if $b^{[\![\iota]\!]}[\zeta] = 1$;
   (b) else if $c^{[\![\iota]\!]} = 1$, $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} \xleftarrow{\$} \{1, ..., 10m-1\}^m$ (the SAFE range), and $\mathsf{Sim}_2(pp, pk, ch, rsp, a)$ outputs $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{A}\mathbf{z}^{[\![\iota]\!]} - (a\mathbf{y})$;
   (c) else if $c^{[\![\iota]\!]} = 0$, $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} := \mathbf{r}^{[\![\iota]\!]}$, and $\mathsf{Sim}_2(pp, pk, ch, rsp)$ outputs $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{A}\mathbf{z}^{[\![\iota]\!]}$ with $\mathbf{r}^{[\![\iota]\!]} \xleftarrow{\$} \{0, ..., 10m-1\}^m$.
3. The simulated transcript is $(ecmt = (a\mathbf{s}^{[\![1]\!]}, ..., a\mathbf{s}^{[\![t]\!]}), ch, rsp = (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$.

□

Aggregation. Define $\mathsf{SAFE} = \{0, 1, ..., 10m-1\}^{km+m}$. For $i \in [k+1]$, let $pp_i := \mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. Define $pp := (pp_1, ..., pp_{k+1}) = \mathbf{A} = (\mathbf{A}_1 || ... || \mathbf{A}_{k+1}) \in \mathbb{Z}_q^{n \times (km+m)}$. For $(k+1)$ valid key pairs $(sk_i = \mathbf{x}_i, pk_i = \mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i)$ under pp_i, the algorithms AggrPK and $\mathsf{AggrCMT}$ are defined as follows.

- $\mathsf{AggrPK}(pk_1, ..., pk_{k+1})$ returns $\widetilde{pk} := \sum_{i \in [k+1]} pk_i = \mathbf{y} = \sum_{i \in [k+1]} \mathbf{y}_i$.
- For $i \in [k+1]$, let $cmt_i := (\mathbf{A}_i\mathbf{r}_i^{[\![1]\!]}, ..., \mathbf{A}_i\mathbf{r}_i^{[\![t]\!]})$ with $\mathbf{r}_i^{[\![\iota]\!]} \xleftarrow{\$} \{0, 1, ..., 10m-1\}^m$ for $\iota \in [t]$. $\mathsf{AggrCMT}(cmt_1, ..., cmt_{k+1})$ returns $\widetilde{cmt} := (\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}) = (\sum_{i \in [k+1]} \mathbf{A}_i\mathbf{r}_i^{[\![1]\!]}, ..., \sum_{i \in [k+1]} \mathbf{A}_i\mathbf{r}_i^{[\![t]\!]})$.
- $pp := (pp_1, ..., pp_{k+1})$ defines the 3-move identification protocol in Fig. 10.

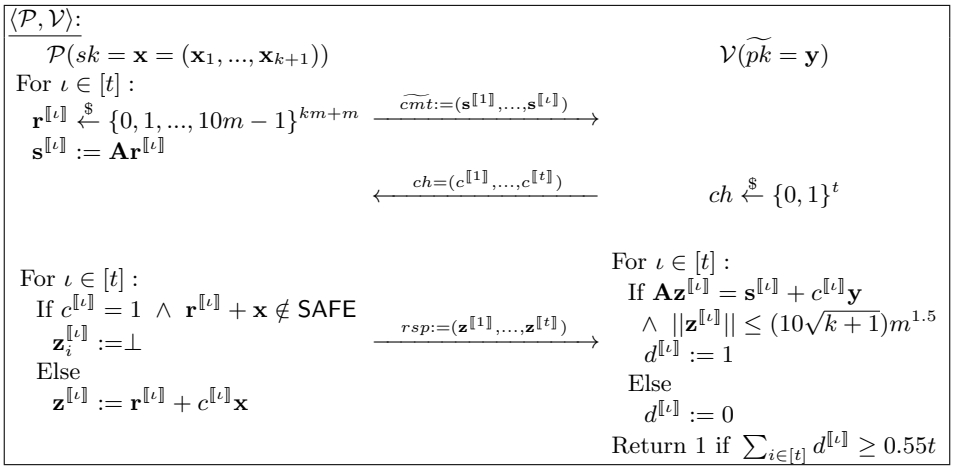

Fig. 10. The aggregation of the SIS-based identification scheme

In fact, the aggregated protocol is the same as before if we view $(k+1)m$ in the aggregation as m in the original one and view $(\mathbf{A}_1||\mathbf{A}_2||...||\mathbf{A}_{k+1}) \in \mathbb{Z}_q^{n \times (km+m)}$ as the parameter $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

Theorem 14. *Assume $k \leq 4$, $m \geq 10$, $k \ll n$, $m = Q\lceil n \log n \rceil$ for constant Q, and the $[n, m, q, \beta]$-SIS assumption holds for $\beta = 20\sqrt{k+1}m^{1.5}$, then the scheme above is a secure $(k+1)$-AIS scheme. More precisely, it has correctness error less than $2^{-t/400}$, one-wayness, $(tn \log q)$-min-entropy, $(1 - 2^{-t/200})$-special soundness, simulatability, and $Q/4$-pseudorandomness. Moreover, it has downward compatibility and homomorphism on coefficient space $\mathbb{Z}_q^*$.*

Proof. **Correctness**. For $\iota \in [t]$, unless $c^{[\![\iota]\!]} = 1$ and $\mathbf{r}^{[\![\iota]\!]} + \mathbf{x} \notin \mathsf{SAFE}$, the response $\mathbf{z}^{[\![\iota]\!]}$ will always pass the verification and hence $d^{[\![\iota]\!]} = 1$. Therefore,

$$\begin{aligned}\Pr[d^{[\![\iota]\!]} = 1] &\geq \Pr[c^{[\![\iota]\!]} = 0] + \Pr[c^{[\![\iota]\!]} = 1] \Pr[\mathbf{r}^{[\![\iota]\!]} + \mathbf{x} \in \mathsf{SAFE} \mid c^{[\![\iota]\!]} = 1] \\ &= 1/2 + 1/2 \cdot (1 - 1/10m)^{km+m} \\ &\geq 0.6, \text{ for } m \geq 10, k \leq 4.\end{aligned}$$

Let $sum = \sum_{\iota \in [t]} d^{[\![\iota]\!]}$. By Chernoff bound, we have

$$\text{correctness error} = \Pr[sum < 0.55t] = \Pr[sum < (0.6 - 0.05)t] \leq e^{-2t(0.05)^2} \leq 2^{-t/400}.$$

One-wayness. Let $pp = \mathbf{A} \in \mathbb{Z}_q^{n \times (km+m)}$ and $sk = \mathbf{x} \in \ddot{\mathcal{SK}} = \{\mathbf{x} \in \mathbb{Z}_q^{km+m} \mid ||\mathbf{x}|| \leq (10\sqrt{k+1})m^{1.5}\}$. Define $f_{pp}(sk) = \mathbf{A}\mathbf{x}$. If an adversary $\mathcal{A}$ can break the one-wayness, then we can construct a reduction algorithm that breaks the SIS assumption as follows.

Given $\mathbf{A}$, the reduction algorithm first samples $\mathbf{x} \xleftarrow{\$} \{0,1\}^{km+m}$, and then passes the public parameter and the public key $(\mathbf{A}, \mathbf{A}\mathbf{x})$ to $\mathcal{A}$. According to Lemma 8 in [Lyu08], with overwhelming probability there exists a distinct $\mathbf{x}' \in \{0,1\}^{km+m} \subset \ddot{\mathcal{SK}}$ such that $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$. And if $\mathcal{A}$ returns $\mathbf{x}' \neq \mathbf{x}$, which happens with probability at least $1/2$, then $(\mathbf{x} - \mathbf{x}')$ is a solution to the SIS($\mathbf{A}$) problem.

Min-entropy. Note that $\widetilde{cmt} := (\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}) = (\sum_{i \in [k+1]} \mathbf{A}_i \mathbf{r}_i^{[\![1]\!]}, ..., \sum_{i \in [k+1]} \mathbf{A}_i \mathbf{r}_i^{[\![t]\!]})$, where $\mathbf{r}_i^{[\![\iota]\!]} \leftarrow \{0, 1, ..., 5m-1\}^m$ for all $i \in [k+1]$ and $\iota \in [t]$. According to Lemma 1, $\mathbf{A}_i \mathbf{r}_i^{[\![\iota]\!]}$ is statistically close to uniform vector $\mathbf{u} \in \mathbb{Z}_q^n$. Therefore, $\widetilde{cmt}$ has a min-entropy about $tn \log q$.

Special soundness. Let $((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, ..., c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$ and $((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]'}, ..., c^{[\![t]\!]'}), (\mathbf{z}^{[\![1]\!]'}, ..., \mathbf{z}^{[\![t]\!]'}))$ be two valid transcripts. First, if there exists $\iota^* \in [t]$ s.t. $c^{[\![\iota^*]\!]} \neq c^{[\![\iota^*]\!]'}$ and $\mathbf{z}^{[\![\iota^*]\!]}, \mathbf{z}^{[\![\iota^*]\!]'} \neq \perp$, then from the equation (w.l.o.g. we assume $c^{[\![\iota^*]\!]} - c^{[\![\iota^*]\!]'} = 1$)

$$\mathbf{A}(\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}) = \mathbf{y}$$

we can extract a solution $(\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'})$ to the SIS problem, and $||\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}|| \leq ||(\mathbf{z}^{[\![\iota^*]\!]})|| + ||\mathbf{z}^{[\![\iota^*]\!]'}|| \leq (20\sqrt{k+1})m^{1.5}$.

Next we argue that the above case happens with probability at least $(1 - 2^{-t/200})$. Recall that $c^{[\![\iota]\!]}, c^{[\![\iota]\!]'} \xleftarrow{\$} \{0,1\}$ for $\iota \in [t]$. If $\sum_{\iota \in [t]} |c^{[\![\iota]\!]} - c^{[\![\iota]\!]'}| > 0.45t$, then by the pigeonhole principle there must exists $\iota^* \in [t]$ s.t. $c^{[\![\iota^*]\!]} \neq c^{[\![\iota^*]\!]'}$ and $\mathbf{z}^{[\![\iota^*]\!]}, \mathbf{z}^{[\![\iota^*]\!]'} \neq \perp$. Due to the Chernoff bound, we have

$$\Pr[\sum_{\iota \in [t]} |c^{[\![\iota]\!]} - c^{[\![\iota]\!]'}| < 0.45t] \leq e^{-2t(0.05)^2} < 2^{-t/200}.$$

Simulatability. The simulator is similar to that in the original IS scheme, and we omit it here.

$Q/4$-pseudorandomness. Similar to the analysis in the original IS scheme, the $Q/4$-pseudorandomness holds or each secret key $\mathbf{x}_i$ ($i \in [k]$), due to Lemma 1.[19]

Collision resistance of secret key. Given randomly sampled $\mathbf{A} = (\mathbf{A}_1 || ... || \mathbf{A}_{k+1}) \in \mathbb{Z}_q^{km+m}$, if the adversary finds two distinct extended secret keys $\mathbf{x} = (\mathbf{x}_1; ...; \mathbf{x}_{k=1}), \mathbf{x}' = (\mathbf{x}'_1; ...; \mathbf{x}'_{k+1}) \in \mathbb{Z}_q^{km+m}$ s.t. $||\mathbf{x}|| \leq (10\sqrt{k+1})m^{1.5}$ and $||\mathbf{x}'|| \leq (10\sqrt{k+1})m^{1.5}$, then $\Delta\mathbf{x} := \mathbf{x} - \mathbf{x}'$ is a short solution for the SIS($\mathbf{A}$) problem, and $||\Delta\mathbf{x}|| \leq (20\sqrt{k+1})m^{1.5} + \sqrt{km+m}$.

Uniformity. Let $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ for all $i \in [k+1]$. Then according to Lemma 1, the function $f_{\mathbf{A}_i}(\cdot) : \{0,1\}^m \times \mathbb{Z}_q^n$ defined as $f_{\mathbf{A}_i}(\mathbf{x}) := \mathbf{A}_i \mathbf{x}$ is a randomness extractor. Therefore, fixed any $\mathbf{x}_1, ..., \mathbf{x}_k$, if $sk_{k+1} = \mathbf{x}_{k+1}$

[19] As discussed before, the pseudorandomness holds only for a constant Q.

distributes uniformly over $\{0,1\}^m$, then $\mathbf{y} := \sum_{i\in[k]} \mathbf{A}_i\mathbf{x}_i + \mathbf{A}_{k+1}\mathbf{x}_{k+1}$ is statistically close to a uniform vector over $\mathbb{Z}_q^n$.

Downward compatibility. This is straightforward.

Homomorphism on coefficient space $\mathbb{Z}_q^*$ and infeasibility of kernel. Given a group of coefficients $a_1, ..., a_k \in \mathbb{Z}_q^*$, a transcript $(cmt, ch, rsp) = ((\mathbf{s}^{[\![1]\!]}, ..., \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, ..., c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, ..., \mathbf{z}^{[\![t]\!]}))$ where $\mathbf{s}^{[\![\iota]\!]} = \sum_{i\in[k]} \mathbf{s}_i^{[\![\iota]\!]} = \sum_{i\in[k]} \mathbf{A}_i\mathbf{r}_i^{[\![\iota]\!]}$ and $\mathbf{z}^{[\![\iota]\!]} = (\mathbf{z}_1^{[\![\iota]\!]}, ..., \mathbf{z}_k^{[\![\iota]\!]})$ for $\iota \in [t]$, the coefficient-verification algorithm $\mathsf{CoeVer}(pp, \widetilde{epk}, (\widetilde{ecmt}, ch, rsp), a_1, ..., a_k)$ works as follows. Here

$$\widetilde{epk} = \sum_{i\in[k]} a_i(\mathbf{A}_i\mathbf{x}_i),$$

$$\widetilde{ecmt} = (\sum_{i\in[k]} a_i\mathbf{s}_i^{[\![1]\!]}, ..., \sum_{i\in[k]} a_i\mathbf{s}_i^{[\![t]\!]}) = (\sum_{i\in[k]} \mathbf{A}_i(a_i\mathbf{r}_i^{[\![1]\!]}), ..., \sum_{i\in[k]} \mathbf{A}_i(a_i\mathbf{r}_i^{[\![t]\!]})).$$

1. For $\iota \in [t]$: if $\sum_{i\in[k]} a_i\mathbf{A}_i\mathbf{z}_i^{[\![\iota]\!]} = \sum_{i\in[k]}(a_i\mathbf{s}_i^{[\![\iota]\!]}) + c^{[\![\iota]\!]}\widetilde{epk}$ and $||\mathbf{z}^{[\![\iota]\!]}|| \leq 10m^{1.5}$, then set $d^{[\![\iota]\!]} := 1$; otherwise set $d^{[\![\iota]\!]} := 0$.
2. Return 1 if $\sum_{i\in[t]} d^{[\![\iota]\!]} \geq 0.8t$ and 0 otherwise.

Meanwhile, given k randomly sampled public keys $\mathbf{y}_1, ..., \mathbf{y}_k \in \mathbb{Z}_q^n$, if $k \ll n$, then with high probability $\mathbf{y}_1, ..., \mathbf{y}_k$ are linear independent, which means that there exits no $a_1, ..., a_k \in \mathbb{Z}_q^*$ such that $\sum_{i\in[k]} a_i\mathbf{y}_i = 0^n$. □

Remark 12 (On the choice of k). From the argument, we can see that k cannot be too large as, otherwise, the security bound will be very loose, resulting in an SIS assumption with stricter parameters. In this construction, we assume $k \leq 4$, which covers most cases of applications in the voting system.

D.2 Security of k-LRS

Theorem 10 (Security of k-LRS). *If AIS is a $(k+1)$-AIS with downward compatibility and homomorphism on $\mathbb{F}$, then the k-LRS scheme above is secure, i.e.t, it has unforgeability, unconditional or computational anonymity, linkability, and non-slanderability.*

Proof. **Unforgeability**. We prove the unforgeability of k-LRS via a series of hybrid games $\mathsf{G}_0, ..., \mathsf{G}_3$. We argue that every two adjacent games are indistinguishable, and in the last game G_3, the adversary's advantage is negligible.

Game G_0. This is the original unforgeability experiment between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$.

Game G_1. In this game, the challenge $\mathcal{C}$ makes a guess $\overline{\mathcal{S}}_{corr}$ on the set of corrupted users $\mathcal{S}_{corr}$ at the beginning of the experiment. If $\overline{\mathcal{S}}_{corr} \neq \mathcal{S}_{corr}$ at the end, then $\mathcal{C}$ aborts and outputs $\perp$.

According to the complexity argument, if $\mathcal{A}$ has a non-negligible advantage in G_0, then $\mathcal{A}$ has a non-negligible advantage in G_1.

Game G_2. In this game, upon receiving a query $\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})$ s.t. $\delta \notin \overline{\mathcal{S}}_{corr}$, instead of returning a signature signed by $\mathsf{sk}^{(\delta)} = (sk_1^{(\delta)}, ..., sk_k^{(\delta)}, sk_{k+1}^{(\delta)})$, the challenger $\mathcal{C}$ works as follows.

1. Let $\mathcal{H}_0(\mathsf{e}) = epp = (epp_1, ..., epp_k)$. For $i \in [k]$, compute $epk_i := f_{epp_i}(sk_i^{(\delta)})$.
2. $a_1, ..., a_k \xleftarrow{\$} \mathbb{F}$, and compute $\mathsf{epk} := \mathsf{AggrPK}(a_1 \cdot epk_1, ..., a_k \cdot epk_k)$.
3. Randomly sample a challenge $ch^{(\delta)} \leftarrow \mathcal{CH}$, and invoke the simulators to generate the two transcripts $(\widetilde{cmt}^{(\delta)}, ch^{(\delta)}, rsp^{(\delta)})$ and $(\widetilde{ecmt}^{(\delta)}, ch^{(\delta)}, rsp_{|k}^{(\delta)})$.

4. Let $(\mathsf{pk}^{(1)}, ..., \mathsf{pk}^{(n)})$ be the public keys of n users in R. For $j = \delta+1, ...n, ..., \delta-1$, let $ch^{(j)} = \mathcal{H}(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, \{a_i\}_{i\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e})$. Invoke the simulators to generate transcripts $(\widetilde{cmt}^{(j)}, ch^{(j)}, rsp^{(j)})$ and $(\widetilde{ecmt}^{(j)}, ch^{(j)}, rsp^{(j)}_{|k})$.
5. Program $\mathcal{H}$ such that $\mathcal{H}(\widetilde{cmt}^{\delta-1}, \widetilde{ecmt}^{\delta-1}, \{a_i\}_{i\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e}) = ch^{(\delta)}$, and return $\sigma := (a_1, ..., a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$. If $\mathcal{H}$ has already been defined on this input, then $\mathcal{C}$ aborts the experiment and outputs $\perp$.

If the reprogram in Step 5 does not fail, then SIGN in G_2 performs statistically close to that in G_1, due to the simulatability of AIS. Meanwhile, thanks to the min-entropy of AIS, with overwhelming probability $\mathcal{H}(\widetilde{cmt}^{\delta-1}, ...)$ has never been queried before, either by the adversary $\mathcal{A}$ or by the challenger $\mathcal{C}$. Therefore, G_1 and G_2 are indistinguishable.

Note that in G_2, for an uncorrupted user δ, its secret key $\mathsf{sk}^{(\delta)} = (sk_1^{(\delta)}, ..., sk_k^{(\delta)}, sk_{k+1}^{(\delta)})$ is used only for generating the event-related public keys. Furthermore, only the first k parts of $\mathsf{sk}^{(\delta)}$ are required.

Game G_3. In this game, $\mathcal{C}$ samples $(\hat{sk}_1^{(\delta)}, ..., \hat{sk}_k^{(\delta)})$ for every $\delta \notin \overline{\mathcal{S}}_{corr}$ independently at the beginning. Upon receiving a query $\text{SIGN}(R, \delta, \mathsf{e}, \mathsf{msg})$, $\mathcal{C}$ uses $(\hat{sk}_1^{(\delta)}, ..., \hat{sk}_k^{(\delta)})$ instead of the original secret key $(sk_1^{(\delta)}, ..., sk_k^{(\delta)})$ of user δ to generate the shifted public keys.

G_3 and G_2 are indistinguishable due to the following equations.

$$
\begin{aligned}
&\left(\left(pp_1, ..., pp_{k+1}, sk_1, ..., sk_k, \widetilde{pk}\right) \middle| \begin{array}{l}\text{For } i \in [k+1] : pp_i \leftarrow \mathsf{Setup}(1^\lambda), (sk_i, pk_i) \leftarrow \mathsf{Gen}(pp_i) \\ \widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, ..., pk_{k+1})\end{array}\right) \\
\approx &\left(\left(pp_1, ..., pp_{k+1}, sk_1, ..., sk_k, \widetilde{pk}\right) \middle| \begin{array}{l}\text{For } i \in [k+1] : pp_i \leftarrow \mathsf{Setup}(1^\lambda), (sk_i, pk_i) \leftarrow \mathsf{Gen}(pp_i) \\ \widetilde{pk} \xleftarrow{\$} \mathcal{PK}\end{array}\right) \\
\equiv &\left(\left(pp_1, ..., pp_{k+1}, \hat{sk}_1, ..., \hat{sk}_k, \widetilde{pk}\right) \middle| \begin{array}{l}\text{For } i \in [k+1] : pp_i \leftarrow \mathsf{Setup}(1^\lambda), (\hat{sk}_i, \hat{pk}_i) \leftarrow \mathsf{Gen}(pp_i) \\ \widetilde{pk} \xleftarrow{\$} \mathcal{PK}\end{array}\right) \\
\approx &\left(\left(pp_1, ..., pp_{k+1}, \hat{sk}_1, ..., \hat{sk}_k, \widetilde{pk}\right) \middle| \begin{array}{l}\text{For } i \in [k+1] : pp_i \leftarrow \mathsf{Setup}(1^\lambda) \\ \quad (sk_i, pk_i), (\hat{sk}_i, \hat{pk}_i) \leftarrow \mathsf{Gen}(pp_i) \\ \widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, ..., pk_{k+1})\end{array}\right)
\end{aligned}
$$

Here "$\approx$" in line 2 and line 4 are due to the uniformity of AIS.

Now we argue that $\mathcal{A}$'s advantage in G_3 is negligible, as otherwise, there exists an extractor that extracts an extended secret key $\ddot{\mathsf{sk}}^{(\delta)} = (\ddot{sk}_1^{(\delta)}, ..., \ddot{sk}_{k+1}^{(\delta)}) \in \ddot{\mathcal{SK}}^{k+1}$ of some uncorrupted user δ, given only pp and $\mathsf{pk}^{(\delta)}$.

More precisely, let $(R^*, \mathsf{e}^*, \mathsf{msg}^*, \sigma^*)$ be $\mathcal{A}$'s final forgery and $\sigma^* = (a_1^*, ..., a_k^*, \mathsf{epk}^*, \{\widetilde{cmt}^{*(j)}, \widetilde{ecmt}^{*(j)}\}_{j\in R^*}, \{rsp^{*(j)}\}_{j\in R^*})$. Let

$$ch^{*(j)} = \mathcal{H}(\widetilde{cmt}^{*j}, \widetilde{ecmt}^{*(j)}, \{a_i^*\}_{i\in[k]}, \mathsf{epk}^*, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[R^*]}, \mathsf{msg}^*, \mathsf{e}^*)$$

for all $j \in R^*$. For $\mathcal{A}$ to win in G_3, it must have queried $ch^{*(j)}$ for all $j \in R^*$. Therefore, $\mathcal{C}$ can locate a user $\delta \in R^*$ such that $\mathcal{A}$ queries $ch^{*(\delta)}$ at last. Note that at the time when $\mathcal{A}$ querying $ch^{*(\delta)}$, the commitments $\widetilde{cmt}^{*(\delta)}$ and $\widetilde{ecmt}^{*(\delta)}$ are fixed in the hash list maintained by $\mathcal{C}$, and $(\widetilde{cmt}^{*(\delta)}, ch^{*(\delta)}, rsp^{*(\delta)})$ is a valid transcript under pp and $\mathsf{pk}^{(\delta)}$.

Let θ be the user preceding δ in R^*. Now we rewind the running of G_3 to the position when $\mathcal{A}$ queries $\mathcal{H}(\widetilde{cmt}^{*(\theta)}, \widetilde{ecmt}^{*(\theta)}, \{a_i^*\}_{i\in[k]}, \mathsf{epk}^*, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[R^*]}, \mathsf{msg}^*, \mathsf{e}^*)$, and return an independent and randomly sampled $ch^{*(\delta)'}$ to $\mathcal{A}$. According to the forking lemma [BN06], with non-negligible probability $\mathcal{A}$ will return

another valid transcript $(\widetilde{cmt}^{*(\delta)}, ch^{*(\delta)'}, rsp^{*(\delta)'})$ under pp and $\mathsf{pk}^{(\delta)}$. Meanwhile, both $ch^{*(\delta)}$ and $ch^{*(\delta)'}$ are sampled from a uniform distribution. According to the special soundness of $(k+1)$-AIS, there is an extractor that extracts an extended secret key $\ddot{\mathsf{sk}}^{(\delta)}$ from these two valid transcripts, given only pp and $\mathsf{pk}^{(\delta)}$, which breaks the one-wayness of AIS.

Unconditional Anonymity. We use two games, G_0 and G_1 to prove the unconditional anonymity, where G_0 is just the original unconditional anonymity experiment.

Game G_1. In this game, upon receiving a query $\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})$ or a query $\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}, \mathsf{msg}_0, \mathsf{msg}_1)$, $\mathcal{C}$ uses the simulator to generate valid transcripts for all users in R, just like G_2 in the proof of unforgeability. Following the same argument as before, we know G_1 and G_0 are statistically close.

Recall that for each secret key, the adversary can obtain at most k signatures from either signing oracle $\textsc{Sign}$ or challenge oracle $\textsc{Chall}$. In G_1, the only information $\mathcal{A}$ gets is two groups of partial secret keys $(sk_1^{(0)}, ..., sk_k^{(0)})$ (used for signing msg_0) and $(sk_1^{(1)}, ..., sk_k^{(1)})$ (used for signing msg_1). However, due to the uniformity of $(k+1)$-AIS, fixed the first k parts of the secret key, $\widetilde{pk} \leftarrow \mathsf{AggrPK}(f_{pp_1}(sk_1^{(b)}), ..., f_{pp_k}(sk_k^{(b)}), f_{pp_{k+1}}(sk_{k+1}^{(b)}))$ distributes uniformly as long as $sk_{k+1}^{(b)}$ distributes uniformly ($b \in \{0,1\}$). Therefore, the probability $(sk_1^{(0)}, ..., sk_k^{(0)})$ is linked to $\mathsf{pk}^{(\delta_0)}$ and the probability $(sk_1^{(1)}, ..., sk_k^{(1)})$ is linked to $\mathsf{pk}^{(\delta_1)}$ are the same, and unconditional anonymity holds as a result.

Computational Anonymity. We use three games, G_0, G_1, and G_2 to prove the computational anonymity. Recall that the adversary now can query signing oracle $\textsc{Sign}$ and challenge oracle $\textsc{Chall}$ multiple times for different events, as long as the signing time for each event is bounded by k.

Game G_0. This is just the original computational anonymity experiment between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$.

Game G_1. In this game, upon receiving a query $\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})$ or a query $\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}, \mathsf{msg}_0, \mathsf{msg}_1)$, $\mathcal{C}$ uses the simulator to generate valid transcripts for all users in R, just like G_2 in the proof of unforgeability. Following the same argument as before, we know G_1 and G_0 are statistically close.

Game G_2. In this game, whenever a signature of $\delta \in [N]$ on event e is generated, instead of using the secret key $(sk_1^{(\delta)}, ..., sk_k^{(\delta)})$, the challenger $\mathcal{C}$ independently samples a group of public keys $(epk_{\mathsf{e},1}^{\delta}, ..., epk_{\mathsf{e},k}^{(\delta)})$, and uses them to generate the event-related public key $\mathsf{epk} := \mathsf{AggrPK}(a_1 \cdot epk_{\mathsf{e},1}^{(\delta)}, ..., a_k \cdot epk_{\mathsf{e},k}^{(\delta)})$. Meanwhile, $\mathcal{C}$ records $(epk_{\mathsf{e},1}^{\delta}, ..., epk_{\mathsf{e},k}^{(\delta)})$ to ensure that the same key is used for the same user δ w.r.t. the same event e.

G_2 and G_1 are indistinguishable due to the pseudorandomness of $(k+1)$-AIS. Furthermore, it is easy to see that in G_2, the experiment when $b = 0$ is identical to the experiment when $b = 1$, which finishes the proof of the computational anonymity.

Linkability. Let $\mathcal{S}_{corr}$ be the set of users corrupted by $\mathcal{A}$. W.l.o.g., we assume $\mathcal{A}$ first uses the corrupted secret keys to generate $k \cdot |\mathcal{S}_{corr}|$ different signatures under some event e^*, and $\mathcal{H}_0(\mathsf{e}^*) = (epp_1, ..., epp_k)$. Now, $\mathcal{A}$'s goal is to generate a valid forgery $\sigma^* = (a_1, ..., a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$ such that $\mathsf{epk} \neq \sum_{i\in[k]} a_i \cdot f_{epp_i}(sk_i^{(j)})$ for any corrupted user $j \in \mathcal{S}_{corr}$. Otherwise, the algorithm Link will return 1 and $\mathcal{A}$ fails in $\mathsf{Exp}_{\mathcal{A},k}^{link}(\lambda)$.

Recall that $\mathcal{H}(\cdot)$ is modeled as a random oracle. Following the same argument in the proof of unforgeability, there exists a δ such that $\mathcal{A}$ queries $ch^{(\delta)}$ at last among all $ch^{(j)}$ in the ring, and $(\widetilde{cmt}^{(\delta)}, ch^{(\delta)}, rsp^{(\delta)})$ is a valid transcript under pp and $\mathsf{pk}^{(\delta)}$. The challenge $\mathcal{C}$ rewinds the simulation of $\mathsf{Exp}_{\mathcal{A},k}^{link}(\lambda)$ and returns an independent and randomly sampled $ch^{(\delta)'}$. Then again according to the forking lemma [BN06], with non-negligible probability $\mathcal{A}$ will return another valid transcript $(\widetilde{cmt}^{(\delta)}, ch^{(\delta)'}, rsp^{(\delta)'})$ in the forgery. By the special soundness of $(k+1)$-AIS, we can extract an extended secret key $\ddot{\mathsf{sk}}^{(\delta)}$, given only pp and $\mathsf{pk}^{(\delta)}$.

Let $\mathsf{sk}^{(\delta)}$ be the secret key of user δ sampled at the beginning of $\mathsf{Exp}^{link}_{\mathcal{A},k}(\lambda)$. If $\ddot{\mathsf{sk}}^{(\delta)} = \mathsf{sk}^{(\delta)}$, then $\mathcal{A}$ fails in $\mathsf{Exp}^{link}_{\mathcal{A},k}(\lambda)$. And if $\ddot{\mathsf{sk}}^{(\delta)} \neq \mathsf{sk}^{(\delta)}$, then we can construct a reduction algorithm to break the collision resistance of the secret key for $(k+1)$-AIS. This concludes the proof of linkability.

Non-slanderability. Let $(\mathsf{e}^*, \delta^*, \{(R_i^*, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[k+1]})$ be $\mathcal{A}$'s final output in $\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k}(\lambda)$, where the first k signatures are signed using $\mathsf{sk}^{(\delta^*)}$, and the last one

$$\sigma_{k+1}^* = (a_1, ..., a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[R_{k+1}^*]}, \{rsp^{(j)}\}_{j\in[R_{k+1}^*]})$$

is forged by $\mathcal{A}$. Similarly, since $\mathcal{H}(\cdot)$ works as a random oracle, we can locate a user δ' such that $\mathcal{A}$ queries $ch^{(\delta')}$ at last among all users in R_{k+1}^*. We analyze the following three cases.

- Case 1. $\delta' \notin \mathcal{S}_{corr}$. Since δ' has not been corrupted, $\mathcal{A}$ breaks the unforgeability in this case, which happens with negligible.
- Case 2. $\delta' \in \mathcal{S}_{corr}$. Let $\mathsf{sk}^{(\delta')} = (sk_1^{(\delta')}, ..., sk_k^{(\delta')}, sk_{k+1}^{(\delta')})$ be the secret key of $\mathsf{pk}^{(\delta')}$. Similar to the argument above, by rewinding we can extract an extended secret key $\ddot{\mathsf{sk}}^{(\delta')} = (\ddot{sk}_1^{(\delta')}, ..., \ddot{sk}_k^{(\delta')}, \ddot{sk}_{k+1}^{(\delta')})$ for $\mathsf{pk}^{(\delta')}$.
  - Case 2.1. $\ddot{\mathsf{sk}}^{(\delta')} \neq \mathsf{sk}^{(\delta')}$. In this case, we can construct a reduction algorithm that breaks the collision resistance of the secret key for $(k+1)$-AIS.
  - Case 2.2 $\ddot{\mathsf{sk}}^{(\delta')} = \mathsf{sk}^{(\delta')}$. Let $\mathcal{H}_0(\mathsf{e}^*) = (epp_1, ..., epp_k)$. From the k signatures signed by $\mathsf{sk}^{(\delta^*)} = (sk_1^{(\delta^*)}, ..., sk_k^{(\delta^*)}, sk_{k+1}^{(\delta^*)})$, the event-related public keys

    $$(epk_1^{(\delta^*)}, ..., epk_k^{(\delta^*)}) = (f_{epp_1}(sk_1^{(\delta^*)}), ..., f_{epp_k}(sk_k^{(\delta^*)}))$$

    of δ^* are totally exposed. At the same time, the event-related public keys $(epk_1^{(\delta')}, ..., epk_k^{(\delta')}) = (f_{epp_1}(sk_1^{(\delta')}), ..., f_{epp_k}(sk_k^{(\delta')}))$ of δ' are fixed. Namely, the adversary $\mathcal{A}$ has to find a group of coefficients $(a_1, ..., a_k)$ such that $\sum_{i\in[k]} a_i \cdot epk_i^{(\delta^*)} = \sum_{i\in[k]} a_i \cdot epk_i^{(\delta')}$, i.e., $\sum_{i\in[k]} a_i \cdot (epk_i^{(\delta^*)} - epk_i^{(\delta')}) = 0_{\mathcal{PK}}$. Recall that both $epk_i^{(\delta^*)}$ and $epk_i^{(\delta')}$ are distributed uniformly over $\mathcal{PK}$ (by the uniformity of AIS). According to the infeasibility of kernel property, this is infeasible for $\mathcal{A}$.

□

E Concrete Schemes

In this section we show the concrete schemes of k-LRS from the DL/DDH assumption and the SIS assumption, respectively.

E.1 Concrete Scheme from the DL/DDH Assumption

Let $\mathbb{G}$ is a cyclic group of order q with generator g, and $H_0 : \{0,1\}^* \to \mathbb{G}$, $H : \{0,1\}^* \to \mathbb{Z}_q$ are two hash functions. The DL-based k-LRS scheme is presented as follows.

- Setup: For $i \in [k+1]$, $\omega_i \xleftarrow{\$} \mathbb{Z}_q^*$, $g_i := g^{\omega_i}$. Return $\mathsf{pp} := (\mathbb{G}, q, g_1, ..., g_{k+1})$.
- Key Generation: For $i \in [k+1]$, $x_i \xleftarrow{\$} \mathbb{Z}_q$, $X_i := \prod_{i\in[k+1]} g_i^{x_i}$, $X := \prod_{i\in[k+1]} X_i$. Return $(sk, pk) := ((x_1, ..., x_{k+1}), X)$.
- Sign: Let $PK_R = (X^{(1)}, ..., X^{(n)})$ be the set of public keys in the ring. W.l.o.g., we assume the signer's index is 1. To sign a message msg under event label e, the signer does as follows.
  1. Sample $a_1, ..., a_k \xleftarrow{\$} \mathbb{Z}_q^*$, compute $H_0(\mathsf{e}) = (e_1, ..., e_k) \in \mathbb{G}^k$, and $\widetilde{X} := \prod_{i\in[k]} e_i^{a_i x_i^{(1)}}$.

2. $r_1, ..., r_{k+1} \xleftarrow{\$} \mathbb{Z}_q$, compute $\widetilde{cmt}^{(1)} := \prod_{i\in[k+1]} g_i^{r_i}$, $\widetilde{ecmt}^{(1)} := \prod_{i\in[k]} e_i^{a_i r_i}$.
3. For $j = 2, ..., n$:
   (a) $\mathbb{Z}_q \ni ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, ..., a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$.
   (b) $z_1^{(j)}, ..., z_{k+1}^{(j)} \xleftarrow{\$} \mathbb{Z}_q$.
   (c) $\widetilde{cmt}^{(j)} := \prod_{i\in[k+1]} g_i^{z_i^{(j)}} \cdot (X^{(j)})^{ch^{(j)}}$, $\widetilde{ecmt}^{(j)} := \prod_{i\in[k]} e_i^{a_i z_i^{(j)}} \cdot \widetilde{X}^{ch^{(j)}}$.
4. $ch^{(1)} := H(\widetilde{cmt}^{(n)}, \widetilde{ecmt}^{(n)}, a_1, ..., a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$.
5. For $i \in [k+1]$: $z_i^{(1)} := r_i - x_i^{(1)} \cdot ch^{(1)}$.
6. Return $\sigma := \left(a_1, ..., a_k, \widetilde{X}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}, z_1^{(j)}, ..., z_{k+1}^{(j)}\}_{j\in[n]}\right)$.

– Verification: Let $PK_R = (X^{(1)}, ..., X^{(n)})$. For $j \in [n]$, compute $ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, ..., a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$ (user n is also referred to as user 0). If

$$\widetilde{cmt}^{(j)} = \prod_{i\in[k+1]} g_i^{z_i^{(j)}} (X^{(j)})^{ch^{(j)}} \ \wedge \ \widetilde{ecmt}^{(j)} := \prod_{i\in[k]} e_i^{a_i z_i^{(j)}} \cdot \widetilde{X}^{ch^{(j)}},$$

holds for all $j \in [n]$, return 1; otherwise return 0.

– Link: Let $\left(\sigma_\iota = (a_{\iota,1}, ..., a_{\iota,k}, \widetilde{X}_\iota, \{\widetilde{cmt}_\iota^{(j)}, \widetilde{ecmt}_\iota^{(j)}, z_{\iota,1}^{(j)}, ..., z_{\iota,k+1}^{(j)}\})\right)_{\iota\in[k+1]}$ be $k+1$ signatures for the same event e (note that different σ_ι may contain different rings). Let $H_0(\mathsf{e}) = (e_1, ..., e_k)$. If there exists a solution $(epk_1, ..., epk_k) \in \mathbb{G}^k$ for the following linear equation system

$$\begin{cases} epk_1^{a_{1,1}} \cdot epk_2^{a_{1,2}} \cdot ... \cdot epk_k^{a_{1,k}} = \widetilde{X}_1, \\ epk_1^{a_{2,1}} \cdot epk_2^{a_{2,2}} \cdot ... \cdot epk_k^{a_{2,k}} = \widetilde{X}_2, \\ ... \\ epk_1^{a_{k+1,1}} \cdot epk_2^{a_{k+1,2}} \cdot ... \cdot epk_k^{a_{k+1,k}} = \widetilde{X}_{k+1}, \end{cases}$$

then return 1 (linked). Otherwise, return 0 (unlinked).

E.2 Concrete Scheme from the SIS Assumption

Let $H_0 : \{0,1\}^* \to (\mathbb{Z}_q^{n\times m})^k$ and $H : \{0,1\}^* \to \{0,1\}^t$ be two hash functions. Define $\mathsf{SAFE} := \{0, 1, ..., 10m-1\}^{km+m}$. The SIS-based k-LRS is presented as follows.

– Setup: For $i \in [k+1]$, sample $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Return $pp := \mathbf{A} = (\mathbf{A}_1||...||\mathbf{A}_{k+1})$.
– Key Generation: For $i \in [k+1]$, $\mathbf{x}_i \xleftarrow{\$} \{0,1\}^m$. Set $\mathbf{x}^\top := (\mathbf{x}_1^\top||...||\mathbf{x}_{k+1}^\top)$ and $\mathbf{y} := \sum_{i\in[k+1]} \mathbf{A}_i\mathbf{x}_i$. Return $(sk, pk) := (\mathbf{x}, \mathbf{y})$.
– Sign: Let $PK_R = (\mathbf{y}^{(1)}, ..., \mathbf{y}^{(n)})$ be the set of public keys in the ring[20]. W.l.o.g., we assume the signer's index is 1. To sign a message msg under event label e, the signer does as follows.
   1. Sample $a_1, ..., a_k \xleftarrow{\$} \mathbb{Z}_q^*$, compute $H_0(\mathsf{e}) = (\mathbf{A}_{\mathsf{e},1}, ..., \mathbf{A}_{\mathsf{e},k}) \in (\mathbb{Z}_q^{n\times m})^k$, and $\tilde{\mathbf{y}} := \sum_{i\in[k]} a_i \mathbf{A}_{\mathsf{e},i}\mathbf{x}_i^{(1)}$.
   2. For $\iota = 1, ..., t$: (the underlying identification protocol repeats t times)
      (a) $\mathbf{r}_{\iota,1}, ..., \mathbf{r}_{\iota,k+1} \xleftarrow{\$} \{0, 1, ..., 10m-1\}^m$, and $\mathbf{r}_\iota^\top := (\mathbf{r}_{\iota,1}^\top||...||\mathbf{r}_{\iota,k+1}^\top)$.
      (b) $\widetilde{cmt}_\iota^{(1)} := \sum_{i\in[k+1]} \mathbf{A}_i\mathbf{r}_{\iota,i} = \mathbf{A}\mathbf{r}_\iota$, $\widetilde{ecmt}_\iota^{(1)} := \sum_{i\in[k]} a_i\mathbf{A}_{\mathsf{e},i}\mathbf{r}_{\iota,i}$.
      (c) $\widetilde{cmt}^{(1)} := (\widetilde{cmt}_1^{(1)}, ..., \widetilde{cmt}_t^{(1)})$, and $\widetilde{ecmt}^{(1)} := (\widetilde{ecmt}_1^{(1)}, ..., \widetilde{cmt}_t^{(1)})$.
   3. For $j = 2, ..., n$:
      (a) $\{0,1\}^t \ni ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, ..., a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$.

[20] Here we slightly abuse the notion n to keep consistency with earlier sections.

(b) For $\iota \in [t]$ and $\zeta \in [km+m]$, sample $b_\iota[\zeta] \leftarrow \chi$, where χ is a biased binary distribution with 0-probability equal to $1/10m$.

(c) For $\iota \in [t]$, sample $\mathbf{r}_{\iota,1}, ..., \mathbf{r}_{\iota,k+1} \xleftarrow{\$} \{0,1,...,10m-1\}^m$, and $\mathbf{r}_\iota^\top := (\mathbf{r}_{\iota,1}^\top || ... || \mathbf{r}_{\iota,k+1}^\top)$.

i. if $ch_\iota^{(j)} = 1$ and there exists $\zeta \in [km+m]$ such that $b_\iota[\zeta] = 0$, then define $\mathbf{z}_\iota = \perp$, $\mathbf{s}_\iota := \sum_{i\in[k+1]} \mathbf{A}_i \mathbf{r}_{\iota,i} = \mathbf{A}\mathbf{r}_\iota$, and $\mathbf{t}_\iota := \sum_{i\in[k]} a_i \mathbf{A}_{\mathsf{e},i} \mathbf{r}_{\iota,i}$.

ii. else if $ch_\iota^{(j)} = 1$, sample $\mathbf{z}_{\iota,1}, ..., \mathbf{z}_{\iota,k+1} \xleftarrow{\$} \{0,1,...,10m-1\}^m$, and compute $\mathbf{s}_\iota := \sum_{i\in[k+1]} \mathbf{A}_i \mathbf{z}_{\iota,i} - \mathbf{y}^{(j)}$, and $\mathbf{t}_\iota := \sum_{i\in[k]} a_i \mathbf{A}_{\mathsf{e},i} \mathbf{z}_{\iota,i} - \tilde{\mathbf{y}}$.

iii. else if $ch_\iota^{(j)} = 0$, define $\mathbf{z}_\iota := \mathbf{r}_\iota$, $\mathbf{s}_\iota := \mathbf{A}\mathbf{z}_\iota$ and $\mathbf{t}_\iota := \sum_{i\in[k]} a_i \mathbf{A}_{\mathsf{e},i} \mathbf{z}_{\iota,i}$, where $\mathbf{z}_\iota^\top = (\mathbf{z}_{\iota,1}^\top || ... || \mathbf{z}_{\iota,k+1}^\top)$.

(d) Define $\widetilde{cmt}^{(j)} := (\mathbf{s}_1^{(j)}, ..., \mathbf{s}_\iota^{(j)}, ..., \mathbf{s}_t^{(j)})$ and $\widetilde{ecmt}^{(j)} := (\mathbf{t}_1^{(j)}, ..., \mathbf{t}_\iota^{(j)}, ..., \mathbf{t}_t^{(j)})$.

4. $ch^{(1)} := H(\widetilde{cmt}^{(n)}, \widetilde{ecmt}^{(n)}, a_1, ..., a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$.
5. For $\iota \in [t]$:
   (a) if $ch_\iota^{(1)} = 1$ and $\mathbf{x}^{(1)} + \mathbf{r}_\iota \notin \mathsf{SAFE}$, set $\mathbf{z}_\iota^{(1)} := \perp$.
   (b) otherwise, $\mathbf{z}_\iota^{(1)} := \mathbf{r}_\iota + ch_\iota^{(1)} \cdot \mathbf{x}^{(1)}$.
6. Return $\sigma := \left(a_1, ..., a_k, \tilde{\mathbf{y}}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}, rsp^{(j)} := \{\mathbf{z}_\iota\}_{\iota\in[t]}\}_{j\in[n]}\right)$

– Verification: Let $PK_R = (\mathbf{y}^{(1)}, ..., \mathbf{y}^{(n)})$ be the set of public keys. For $j = 1, ..., n$, compute $ch^{(j)} := H(\widetilde{cmt}^{(i-1)}, \widetilde{ecmt}^{(i-1)}, a_1, ..., a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$ (user n is also referred as user 0).
1. For $j \in [n]$, define $sum^{(j)} = 0$.
2. For $j \in [n]$ and $\iota \in [t]$, if

$$\mathbf{A}\mathbf{z}_\iota^{(j)} = \widetilde{cmt}^{(j)} + ch_\iota^{(j)} \cdot \mathbf{y}^{(j)} \ \wedge \ \sum_{i\in[k]} a_i \mathbf{A}_{\mathsf{e},i} \mathbf{z}_{\iota,i}^{(j)} = \widetilde{ecmt}^{(j)} + ch_\iota^{(j)} \cdot \tilde{\mathbf{y}},$$

and $||\mathbf{z}_\iota^{(j)}|| \le (10\sqrt{k+1})m^{1.5}$, then $sum^{(j)} := sum^{(j)} + 1$.

3. If $sum^{(j)} \ge 0.55t$ for all $j \in [n]$, then return 1, otherwise, return 0.

– Link: Let $\{\sigma_\ell = (a_{\ell,1}, ..., a_{\ell,k}, \tilde{\mathbf{y}}_\ell, ...)\}_{\ell\in[k+1]}$ be $(k+1)$ signatures for the same event e. Let $H_0(\mathsf{e}) = (\mathbf{A}_{\mathsf{e},1}, ..., \mathbf{A}_{\mathsf{e},k})$. If there exists a solution $(epk_1, ..., epk_k) \in (\mathbb{Z}_q^n)^k$ for the following linear equation system,

$$\begin{cases} a_{1,1} \cdot epk_1 + a_{1,2} \cdot epk_2 + ... + a_{1,k} \cdot epk_k = \tilde{\mathbf{y}}_1, \\ a_{2,1} \cdot epk_1 + a_{2,2} \cdot epk_2 + ... + a_{2,k} \cdot epk_k = \tilde{\mathbf{y}}_2, \\ ... \\ a_{k+1,1} \cdot epk_1 + a_{k+1,2} \cdot epk_2 + ... + a_{k+1,k} \cdot epk_k = \tilde{\mathbf{y}}_{k+1}, \end{cases}$$

then return 1 (linked). Otherwise, return 0 (unlinked).

Remark 13. The SIS-based instantiation above achieves unforgeability but not strong unforgeability, since given a valid signature, an adversary can easily obtain another valid signature by switching some $\mathbf{z}$ (the response in AIS) to $\perp$.

Remark 14 (On the parameter of the SIS-based construction). The SIS-based construction above supports only a small parameter k (e.g., $k \le 4$), as larger values would result in an unacceptably high correctness error. Meanwhile, each signer can issue signatures for at most a constant number Q of different events (e.g., $Q \le 10$) to ensure anonymity.

F Multi-Event LRS: Definition and Generic Construction from k-LRS

In this section we formally define multi-event LRS (MLRS) and its security requirements, and show a generic construction from k-LRS.

F.1 Defnition of Multi-Event LRS

Definition 23 ($(k_1,...,k_M)$-MLRS). *Let $M, k_1, ..., k_M \in \mathbb{N}^+$. A $(k_1,...,k_M)$-multi-event linkable ring signature ($(k_1,...,k_M)$-MLRS) scheme consists of the following five algorithms. Namely,*
$(k_1,...,k_M)$-$\mathsf{MLRS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Link})$.

- $pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M)$*. The setup algorithm takes as input the security parameter 1^λ, the upper bounds of event index M and unlinked signing bounds for different events $k_1, ..., k_M$, and outputs a public parameter pp.*
- $(sk, pk) \leftarrow \mathsf{Gen}(pp)$*: The key generation algorithm takes as input pp, and outputs a pair of secret and public keys (pk, sk). W.l.o.g., we assume pp is contained in pk.*
- $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})$*: The signing algorithm takes as input a group of public keys PK_R, the secret key $sk^{(\delta)}$ of user $\delta \in R$, event labels $\mathsf{e}_1, ..., \mathsf{e}_M$ and a message msg, and outputs a signature σ.*
- $1/0 \leftarrow \mathsf{Ver}(PK_R, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}, \sigma)$*: The verification algorithm Ver takes as input R, PK_R, $\mathsf{e}_1, ..., \mathsf{e}_M$, msg and σ, and outputs a bit indicating the validity of σ.*
  We say that a signature σ (w.r.t. events $\mathsf{e}_1, ..., \mathsf{e}_M$, message msg and ring R) is valid (resp., invalid), if $\mathsf{Ver}(PK_R, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}, \sigma) = 1$ (resp., $\mathsf{Ver}(PK_R, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}, \sigma) = 0$).
- $1/0 \leftarrow \mathsf{Link}(\pi, \mathsf{e}_\pi, \sigma_1, ..., \sigma_{k_\pi+1})$*: The link algorithm that takes as input an event index $\pi \in [M]$, an event label e_π and $(k_\pi + 1)$ signatures $\sigma_1, ..., \sigma_{k_\pi+1}$, and outputs a bit, where 1 indicates that these $(k_\pi + 1)$ signatures are linked to one actual signer, and 0 indicates unlinked.*

Correctness.

1. *Let $M, N, k_1, ..., k_M \in \mathbb{N}^+$. For any $R \subseteq [N]$ and $\delta \in R$, any events $(\mathsf{e}_1, ..., \mathsf{e}_M)$ and message msg, it holds that*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M) \\ (sk^{(1)}, pk^{(1)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp) : \mathsf{Ver}(PK_R, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}, \sigma) = 0 \\ \sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}) \end{array}\right] \leq negl(\lambda).$$

2. *Let $M, N, k_1, ..., k_M \in \mathbb{N}^+$. For any $\pi \in [M]$, any $R_i \subseteq [N]$ and $\delta_i \in R_i$ ($i \in [k_\pi + 1]$), any event e_π and messages $\mathsf{msg}_1, ..., \mathsf{msg}_{k_\pi+1}$, it holds that*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M) \\ (sk^{(1)}, pk^{(1)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp) : \mathsf{Link}(\pi, \mathsf{e}_\pi, \sigma_1, ..., \sigma_{k_\pi+1}) = 1 \\ \sigma_i \leftarrow \mathsf{Sign}(PK_{R_i}, sk^{(\delta_i)}, ..., \mathsf{e}_\pi, ..., \mathsf{msg}_i) \end{array}\right] = 1, \quad \text{if } \delta_1 = \delta_2 = ... = \delta_{k_\pi+1};$$

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M) \\ (sk^{(1)}, pk^{(i)}), ..., (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp) : \mathsf{Link}(\pi, \mathsf{e}_\pi, \sigma_1, ..., \sigma_{k_\pi+1}) = 1 \\ \sigma_i \leftarrow \mathsf{Sign}(PK_{R_i}, sk^{(\delta_i)}, ..., \mathsf{e}_\pi, ..., \mathsf{msg}_i) \end{array}\right] \leq negl(\lambda), \quad \text{otherwise.}$$

Remark 15. It is naturally required that for two different event indexes $\pi_1 \neq \pi_2$, the two corresponding event spaces are disjoint. That is, an event label e under index π_1 cannot be used as a label under index π_2. This can be easily achieved by adding the index as a prefix of the event string.

As for security, we require unforgeability, (unconditional) anonymity, linkability, and non-slanderability for $(k_1, ..., k_M)$-MLRS.

Unforgeability. The adversary cannot forge a signature for a new message on behalf of a ring R in which it is not included.

(Unconditional) anonymity. For any event index π and the corresponding signing bound k_π, given at most k_π ring signatures, the adversary (even it is computationally unbounded) cannot detect which signer actually signed them.

Linkability. The adversary who corrupts at most s signers cannot output $(s \cdot k_\pi + 1)$ different signatures such that, every $(k_\pi + 1)$ of them are unlinked under the same event e_π.

Non-slanderability. After seeing k_π signatures from one honest signer under some event e_π, the adversary cannot generate a valid forgery that makes these $k_\pi + 1$ signature linked.

More formally, we define the security properties via the following experiments.

Definition 24 (Unforgeability of $(k_1, ..., k_M)$-MLRS). *Let* MLRS *be a $(k_1, ..., k_M)$-MLRS scheme. We say that* MLRS *has unforgeability, if,* $\mathsf{Adv}^{unforg}_{\mathcal{A},k_1,...,k_M}(\lambda) := \Pr[\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,...,k_M}(\lambda) = 1] \leq negl(\lambda)$ *holds for all PPT adversary $\mathcal{A}$, where the experiment $\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,...,k_M}(\lambda)$ is defined in Fig. 11.*

$\underline{\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,...,k_M}(\lambda)}$:	$\underline{\textsc{Sign}(R, \delta, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})}$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M)$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	$\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}, \sigma)\}$
$(R^*, \mathsf{e}^*_1, ..., \mathsf{e}^*_M, \mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	Return σ
If $R^* \subseteq [N] \ \wedge \ R^* \cap \mathcal{S}_{corr} = \varnothing \ \wedge \ (R^*, \mathsf{e}^*_1, ..., \mathsf{e}^*_M, \mathsf{msg}^*, \cdot) \notin \mathcal{S}$ $\wedge \ \mathsf{Ver}(PK_{R^*}, \mathsf{e}^*_1, ..., \mathsf{e}^*_M, \mathsf{msg}^*, \sigma^*) = 1$: output 1	$\underline{\textsc{Corr}(j)}$:
Otherwise: output 0	$\mathcal{S}_{corr} := \{j\}$
	Return $sk^{(j)}$

Fig. 11. The unforgeability experiment of $(k_1, ..., k_M)$-MLRS.

Definition 25 ((Unconditional) anonymity of $(k_1, ..., k_M)$-MLRS). *Let* MLRS *be a $(k_1, ..., k_M)$-MLRS scheme. We say that* MLRS *has anonymity, if,* $\mathsf{Adv}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda) := |\Pr[\mathsf{Exp}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda) = 1] - 1/2| \leq negl(\lambda)$ *holds for all PPT adversary $\mathcal{A}$, where $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda)$ is defined in Fig. 12.*

If $\mathsf{Adv}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda) \leq negl(\lambda)$ even for all powerful adversary $\mathcal{A}$, then we say MLRS *has unconditional anonymity.*

$\underline{\mathsf{Exp}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda)}$:	$\underline{\textsc{Sign}(R, \delta, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})}$:
$b \xleftarrow{\$} \{0, 1\}$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M)$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})$
$\mathcal{S}_{pair} := \varnothing$	For $\iota \in [M]$: $++ \mathcal{L}[\delta, \mathsf{e}_\iota]$
For $j \in [N]$:	$++ \mathcal{T}[\delta]$
$(sk^{(j)}, pk^{(i)}) \leftarrow \mathsf{Gen}(pp)$	Return σ
$\mathcal{T}[j] := 0$	
For all δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$	$\underline{\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}_0, \mathsf{msg}_1)}$:
$b' \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Chall}(b,\cdots)}(\{pk^{(j)}\}_{j\in[N]})$	If $R \nsubseteq [N] \ \vee \ \delta_0 \notin R \ \vee \ \delta_1 \notin R$: return $\perp$
	If $\mathcal{T}[\delta_0] \geq \min_\iota(k_\iota) \vee \mathcal{T}[\delta_1] \geq \min_\iota(k_\iota)$: return $\perp$
If $b = b'$: output 1	For $\iota \in [M]$: if $\mathcal{L}[\delta_0, \mathsf{e}_\iota] \geq k_\iota \ \vee \ \mathcal{L}[\delta_1, \mathsf{e}_\iota] \geq k_\iota$: return $\perp$
Otherwise: output 0	For $\iota \in [M]$: $++ \mathcal{L}[\delta_0, \mathsf{e}_\iota]$, $++ \mathcal{L}[\delta_1, \mathsf{e}_\iota]$
	$\sigma_0 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_0 \oplus b)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}_0)$
	$\sigma_1 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_1 \oplus b)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}_1)$
	Return (σ_0, σ_1)

Fig. 12. The anonymity experiment of $(k_1, ..., k_M)$-MLRS, where the hightlighted codes are only shown in the unconditional anonymity experiment.

Definition 26 (Linkability of $(k_1,.,,,.k_M)$-MLRS). *Let* MLRS *be a $(k_1,...,k_M)$-MLRS scheme. We say that* MLRS *has unforgeability, if,* $\mathsf{Adv}^{link}_{\mathcal{A},k_1,...,k_M}(\lambda) := \Pr[\mathsf{Exp}^{link}_{\mathcal{A},k_1,...,k_M}(\lambda)=1] \leq negl(\lambda)$ *holds for all PPT adversary $\mathcal{A}$, where the experiment* $\mathsf{Exp}^{link}_{\mathcal{A},k_1,...,k_M}(\lambda)$ *is defined in Fig. 13.*

$\underline{\mathsf{Exp}^{link}_{\mathcal{A},k_1,...,k_M}(\lambda)}$:	$\underline{\textsc{Sign}(R,\delta,\mathsf{e}_1,...,\mathsf{e}_M,\mathsf{msg})}$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M)$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	For $\iota \in [M]$: if $\mathcal{L}[\delta,\mathsf{e}_\iota] \geq k_\iota$: return $\perp$
For all δ, e: $\mathcal{L}[\delta,\mathsf{e}] := 0$	For $\iota \in [M]$: $++\,\mathcal{L}[\delta,\mathsf{e}_\iota]$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})$
$(\pi^*, \mathsf{e}^*, \{(R_i^*, \mathsf{msg}_i^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \sigma_i^*)\}_{i\in[\ell]}) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^j\}_{j\in[N]})$	$\mathcal{S} := \mathcal{S} \cup \{(R,\mathsf{e}_1,...,\mathsf{e}_M,\mathsf{msg},\sigma)\}$
	Return σ
If $(\forall i \in [\ell] : (R_i^*, \mathsf{e}_{i,1}, ..., \mathsf{e}_{i,M}, \mathsf{msg}_i^*, \cdot) \notin \mathcal{S}$	
$\quad \wedge \ \mathsf{Ver}(PK_{R_i^*}, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \mathsf{msg}_i^*, \sigma_i^*) = 1 \ \wedge \ \mathsf{e}_{i,\pi^*} = \mathsf{e}^*)$	$\underline{\textsc{Corr}(j)}$:
$\wedge \ (\forall G \subseteq [\ell], \lvert G\rvert = k_{\pi^*}+1 : \mathsf{Link}(\pi^*, \mathsf{e}^*_{\pi^*}, \{\sigma_i^*\}_{i\in G}) = 0)$	$\mathcal{S}_{corr} := \mathcal{S}_{corr} \cup \{j\}$
$\wedge \ \lvert\mathcal{S}_{corr}\rvert \leq \lfloor(\ell-1)/k_{\pi^*}\rfloor$: output 1	Return $sk^{(j)}$
Otherwise: output 0	

Fig. 13. The linkability experiment of $(k_1,...,k_M)$-MLRS.

Definition 27 (Non-slanderability of $(k_1,.,,,.k_M)$-MLRS). *Let* MLRS *be a $(k_1,...,k_M)$-MLRS scheme. We say* MLRS *has unforgeability, if* $\mathsf{Adv}^{non\text{-}sl}_{\mathcal{A},k_1,...,k_M}(\lambda) := \Pr[\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,...,k_M}(\lambda)=1] \leq negl(\lambda)$ *holds for all PPT adversary $\mathcal{A}$, where the experiment* $\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,...,k_M}(\lambda)$ *is defined in Fig. 14.*

$\underline{\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,...,k_M}(\lambda)}$:	$\underline{\textsc{Sign}(R,\delta,\mathsf{e}_1,...,\mathsf{e}_M,\mathsf{msg})}$:
$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, ..., k_M)$	If $R \nsubseteq [N] \ \vee \ \delta \notin R$: return $\perp$
For $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$	For $\iota \in [M]$: if $\mathcal{L}[\delta,\mathsf{e}_\iota] \geq k_\iota$: return $\perp$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$	For $\iota \in [M]$: $++\,\mathcal{L}[\delta,\mathsf{e}_\iota]$
For all δ, e: $\mathcal{L}[\delta,\mathsf{e}] := 0$	$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg})$
$(\pi^*, \mathsf{e}^*, \delta^*, \{(R_i^*, \mathsf{msg}_i^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \sigma_i^*)\}_{i\in[k_{\pi^*}+1]})$	$\mathcal{S} := \mathcal{S} \cup \{(R,\delta,\mathsf{e}_1,...,\mathsf{e}_M,\mathsf{msg},\sigma)\}$
$\leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$	Return σ
If $\{(R_i^*, \delta^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[k_{\pi^*}]} \subseteq \mathcal{S}$	
$\wedge \ (R^*_{k_{\pi^*}+1}, \delta^*, \{\mathsf{e}_{k_{\pi^*}+1,\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_{k_{\pi^*}+1}, \cdot) \notin \mathcal{S}$	
$\wedge \ (\forall i \in [k_{\pi^*}+1] : \mathsf{e}_{i,\pi^*} = \mathsf{e}^*)$	
$\wedge \ \mathsf{Ver}(PK_{R^*_{k_{\pi^*}+1}}, \{\mathsf{e}_{k_{\pi^*}+1,\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_{k_{\pi^*}+1}, \sigma^*_{k_{\pi^*}+1}) = 1$	$\underline{\textsc{Corr}(j)}$:
$\wedge \ \mathsf{Link}(\pi^*, \mathsf{e}^*_{\pi^*}, \{\sigma_i^*\}_{i\in[k_{\pi^*}+1]}) = 1 \ \wedge \ \delta^* \notin \mathcal{S}_{corr}$:	$\mathcal{S}_{corr} := \{j\}$
output 1	Return $sk^{(j)}$
Otherwise: output 0	

Fig. 14. The non-slanderability experiment of $(k_1,...,k_M)$-MLRS.

F.2 Generic Construction from k-LRS

Let N be the total number of users, M be the total number of event indexes, and $k_1,...,k_M$ be the upper bounds of signing times w.r.t. different event indexes. Let k_1-LRS, ..., k_M-LRS be a series of LRS schemes. Our MLRS scheme $(k_1,...,k_M)$-MLRS is constructed as follows.

– Setup: For all $i \in [M]$: $pp_i \leftarrow k_i\text{-LRS.Setup}(1^\lambda, k_i)$. Return the public parameter $pp := (pp_1, ..., pp_M)$.

– Key Generation: For all $i \in [M]$: $(sk_i, pk_i) \leftarrow k_i\text{-}\mathsf{LRS.Gen}(pp_i)$. Return the secret key $\mathsf{sk} = (sk_1, ..., sk_M)$ and $\mathsf{pk} = (pk_1, ..., pk_M)$.
– Sign: Let $(\mathsf{pk}^{(1)}, ..., \mathsf{pk}^{(n)})$ be the public keys of n users in the ring, and assume the signer's index is 1. To sign the message msg under event $(\mathsf{e}_1, ..., \mathsf{e}_M)$, the signer does as follows.
  1. Let $\overline{\mathsf{msg}} := (R||\mathsf{msg}||\mathsf{e}_1||...||\mathsf{e}_M)$.
  2. Parse $\mathsf{sk}^{(1)} = (sk_1, ..., sk_M)$. For $i \in [M]$, use sk_i to sign $\overline{\mathsf{msg}}$ under event e_i, i.e., $\sigma_i \leftarrow k_i\text{-}\mathsf{LRS.Sign}(PK_R, sk_i, \mathsf{e}_i, \overline{\mathsf{msg}})$.
  3. Return $\sigma := (\sigma_1, ..., \sigma_M)$.
– Verify: For all $i \in [M]$, let $PK_{R,i}$ be the assembly of the i-parts of public keys for users in R. If $k_i\text{-}\mathsf{LRS.Ver}(PK_{R,i}, \mathsf{e}_i, \overline{\mathsf{msg}}, \sigma_i) = 1$ for all $i \in [M]$, then return 1. Otherwise, return 0.
– Link: Let $\{\sigma_i = (\sigma_{i,1}, ..., \sigma_{i,M})\}_{i \in [k_\pi + 1]}$ be $(k_\pi + 1)$ signatures whose corresponding π-th event labels are e_π. Return $k_\pi\text{-}\mathsf{LRS.Link}(\mathsf{e}_\pi, \sigma_{1,\pi}, ..., \sigma_{k_\pi+1,\pi})$.

Correctness. The correctness of $(k_1, ..., k_M)$-MLRS follows directly from the correctness (of both verification and linkability) of the underlying LRS schemes k_1-LRS, ..., k_M-LRS.

Theorem 15. *If k_1-LRS, ..., k_M-LRS are secure LRS schemes (i.e., they have unforgeability, unconditional (resp., computational) anonymity, linkability, and non-slanderability), then the MLRS scheme $(k_1, ..., k_M)$-MLRS constructed above is secure, i.e., it has unforgeability, unconditional (resp., computational) anonymity, linkability, and non-slanderability.*

Proof. The proof is straightforward and we describe the sketch here.

Unforgeability. Let $(R^*, \mathsf{e}_1^*, ..., \mathsf{e}_M^*, \mathsf{msg}^*, \sigma^*)$ be $\mathcal{A}$'s final forgery and $\sigma^* = (\sigma_1^*, ..., \sigma_M^*)$. Recall $\mathcal{A}$ has never queried the signing oracle with $(R^*, \mathsf{e}_1^*, ..., \mathsf{e}_M^*, \mathsf{msg}^*)$, as otherwise $\mathcal{A}$ fails in the unforgeability experiment. Therefore,

$$\overline{\mathsf{msg}}^* := (R^*||\mathsf{msg}^*||\mathsf{e}_1^*||...||\mathsf{e}_M^*)$$

has never occurred in the simulation of the signing oracle, and $(R^*, \mathsf{e}_1^*, \overline{\mathsf{msg}}^*, \sigma_1^*)$ is a valid message-signature pair for k_1-LRS, which means that $\mathcal{A}$ breaks the unforgeability of k_1-LRS (the same case for $i = 2, ..., M$).

Unconditional Anonymity. We use a series of hybrid games $\mathsf{G}_0, ..., \mathsf{G}_M$ to prove the unconditional anonymity.

Game G_i $(0 \le i \le M)$. In this game, upon receiving $\textsc{Chall}(R, \delta_0, \delta_1, \mathsf{e}_1, ..., \mathsf{e}_M, \mathsf{msg}_0, \mathsf{msg}_1)$, the challenger $\mathcal{C}$ first sets $\overline{\mathsf{msg}}_0 := (R||\mathsf{msg}_0||\mathsf{e}_1||...||\mathsf{e}_M)$ and $\overline{\mathsf{msg}}_1 := (R||\mathsf{msg}_1||\mathsf{e}_1||...||\mathsf{e}_M)$, and then does the followings.

– For $j \le i$, it signs $\overline{\mathsf{msg}}_0$ using $\delta^{(1)}$'s j-th secret key to get $\sigma_{j,0}$, and signs $\overline{\mathsf{msg}}_1$ using $\delta^{(0)}$'s j-th secret key to get $\sigma_{j,1}$.
– For $i < j \le M$, it signs $\overline{\mathsf{msg}}_0$ using $\delta^{(0)}$'s j-th secret key to get $\sigma_{j,0}$, and signs $\overline{\mathsf{msg}}_1$ using $\delta^{(1)}$'s j-th secret key to get $\sigma_{j,1}$.
– Finally it returns $(\sigma_0, \sigma_1) := ((\sigma_{0,1}, ..., \sigma_{0,M}), (\sigma_{1,1}, ..., \sigma_{1,M}))$.

It is easy to see that G_0 is just the experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda)$ with $b = 0$, and G_M is just the experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,...,k_M}(\lambda)$ with $b = 1$. Meanwhile, the adjacent two games G_i and G_{i+1} $(0 \le i < M)$ are statistically indistinguishable due to the unconditional anonymity of k_i-LRS. Then the anonymity of $(k_1, ..., k_M)$-MLRS holds as a result.

Computational Anonymity. The proof of computational anonymity is similar as above and we omit here.

Linkability. Let $(\pi^*, \mathsf{e}^*, \{(R_i^*, \mathsf{msg}_i^*, \{\mathsf{e}_{i,\iota}\}_{\iota \in [M]}, \sigma_i^*)\}_{i \in [\ell]})$ be $\mathcal{A}$'s final output in $\mathsf{Exp}^{link}_{\mathcal{A},k_1,...,k_M}(\lambda)$, and $\sigma_i^* = (\sigma_{i,1}^*, ..., \sigma_{i,M}^*)$. Recall that all these signatures share the same π^*-th event label e^*. We focus on the π^*-th parts of all signatures, i.e., $\{\sigma_{i,\pi^*}^*\}_{i \in [\ell]}$. If $\mathcal{A}$ wins, then we have

1. $\forall i \in [\ell]$: $k_{\pi^*}\text{-}\mathsf{LRS.Ver}(PK_{R_i^*,\pi^*}, \mathsf{e}_{i,\pi^*}, \overline{\mathsf{msg}}_i^*, \sigma_{i,\pi^*}^*) = 1$, where

$$\overline{\mathsf{msg}}_i^* := (R_i^*||\mathsf{msg}_i^*||\mathsf{e}_{i,1}||...||\mathsf{e}_{i,M}).$$

2. $\forall G \subseteq [\ell]$ s.t. $|G| = k_{\pi^*} + 1$: $k_{\pi^*} - \mathsf{LRS.Link}(\mathsf{e}^*, \{\sigma^*_{i,\pi^*}\}_{i\in G}) = 0$.
3. $|\mathcal{S}_{corr}| \leq \lfloor (\ell - 1)/k_{\pi^*} \rfloor$.

Namely, $\mathcal{A}$ breaks the linkability of k_{π^*}-LRS.

Non-slanderability. The arguement is similar as above. Let

$$(\pi^*, \mathsf{e}^*, \delta^*, \{(R^*_i, \mathsf{msg}^*_i, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \sigma^*_i)\}_{i\in[k_{\pi^*}+1]})$$

be $\mathcal{A}$'s final output in $\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$, and $\sigma^*_i = (\sigma^*_{i,1}, ..., \sigma^*_{i,M})$. Recall that all these signatures share the same π^*-th event label e^*. We focus on the π^*-th parts of all signatures, i.e., $\{\sigma^*_{i,\pi^*}\}_{i\in[\ell]}$. If $\mathcal{A}$ wins, then we have

1. The first k_{π^*} signatures come from the signing oracle, i.e.,

$$\{(R^*_i, \delta^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_i, \sigma^*_i)\}_{i\in[k_{\pi^*}]} \subseteq \mathcal{S}.$$

2. The last signature is outside of $\mathcal{S}$, i.e.,

$$(R^*_{k_{\pi^*}+1}, \delta^*, \{\mathsf{e}_{k_{\pi^*},\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_{k_{\pi^*}+1}, \sigma^*_{k_{\pi^*}+1}) \notin \mathcal{S}.$$

3. k_{π^*}-$\mathsf{LRS.Ver}(PK_{R^*_{k_{\pi^*}+1},\pi^*}, \mathsf{e}^*, \overline{\mathsf{msg}}^*_{k_{\pi^*}+1}, \sigma^*_{k_{\pi^*}+1,\pi^*}) = 1$, where

$$\overline{\mathsf{msg}}^*_{k_{\pi^*}+1} := (R^*_{k_{\pi^*}+1} || \mathsf{msg}^*_{k_{\pi^*}+1} || \mathsf{e}_{k_{\pi^*}+1,1} || ... || \mathsf{e}_{k_{\pi^*}+1,M}).$$

4. k_{π^*}-$\mathsf{LRS.Link}(\mathsf{e}^*, \{\sigma_{i,\pi^*}\}_{i\in[k_{\pi^*}+1]}) = 1$.

Namely, $\mathcal{A}$ breaks the non-slanderability of k_{π^*}-LRS. □

G Discussions

G.1 On the Necessity of Stateless k-LRS and Linking All $k+1$ Signatures

Although statelessness is not a necessary, but rather a flourish property in many (signature-based) voting cases, it is still desirable in the case where there are many phases of a voting event. Additionally, statelessness is the golden and standard property in the syntax of signatures (including k-LRS), and it is natural and desirable to make a k-LRS scheme stateless. This is not only for voting, but also for all other applications of k-LRS.

The trivial solution from "using LRS k times" (see the technical overview in Sec. 1.3) cannot achieve k-linkability (linking all $k+1$ signatures) as ours. In voting, there are two main benefits of this property. First, this prevents the tally inconsistency caused from canceling just part of the vote. Suppose we allow voters to overvote but remove only the excessive part of their signatures. In such cases, say $k = 2$, if three signatures are linked, what signature should be chosen to be removed? This might require a more complex, error-prone structure. Second, this creates an incentive for voters to follow the rule because otherwise they will be penalized.

Detailed Comparion with [BL16] . As we mentioned in the related work, both [BL16] and our work achieve k-linkability, and the stateful scheme in [BL16] can be viewed as an extension of the "using LRS k times" paradigm. Regardless of the insecurity of [BL16] (Appendix B), our scheme is superior to [BL16] in: (1) statelessness; (2) public key size ($O(1)$ v.s. $O(k)$), but worse than [BL16] in: (3) signing/verification time ($O(kn)$ v.s. $O(n)$).

G.2 Other Applications of k-LRS

We discuss additional applications of k-LRS as follows.

Trial Browsing of Contents. In trial browsing of contents [TFS04], a server aims to allow users anonymously browse content such as movies and music on trial. At the same time, the server wants to bound the number of times a user can access the service, avoiding a moocher using it excessively. Let k be the bound predetermined by the server. Then this problem can be solved via a k-LRS scheme as follows. The server maintains a log which contains all queries from anonymous users. Every time a user requests a service from the server, it generates a k-LRS signature for this request under the event label the identity of the server, and then sends the message-event-signature tuple to the server. Via the link algorithm, the server can check within the log whether there are already exist k signatures that make these $(k+1)$ signatures linked, i.e., whether the user has already used all its k trials. At the same time, the identity of the user is hidden due to the anonymity of k-LRS.

K-times Anonymous Authentication. In the classical "client-server" scenario, the client needs to authenticate itself to the server anonymously. At the same time, the server might want to limit the number of times each client can access it. For example, a basic member of a music website can download up to 10 tracks of music every month. To acheive this goal, the client signs its request every time using a k-LRS scheme, hiding its real identity from a group of clients. The server, who maintains a log for every request and checks the validity via the link algorithm, can guarantee that no request from a client will get accepted if it exceeds the limit.

Anonymous Veto. An anonymous veto [BL16] scheme allows members of a group to anonymously express up to k vetoes. For instance, in a decision-making process, each participant can anonymously veto up to k options without revealing their identity. This ensures that individuals can express their objections without fear of retribution or bias. Similar to the application in voting, this problem can be solved perfectly via a k-LRS scheme.

G.3 Further Discussion on AIS

We formalize the concept of $(k+1)$-aggregatable identification schemes (AIS) to refine the core technique used in constructing k-LRS, and provide two instantiations based on the DL assumption (or the DDH assumption) and the SIS assumption, respectively. Except k-LRS, AIS may also contribute to the construction of threshold ring signatures, multi-signatures, and related schemes. In a recent work [BPW23], Boneh, Partap, and Waters introduced the concept of accountable multi-signatures with constant size public keys, where a single public signature aggregation key pkc is generated during setup and used for verification. AIS appears to have potential applications in constructing accountable multi-signatures, which we leave for our future work.

H Anonymous Bulletin Board

We recall the definition of Anonymous Bulletin Boards [IKOS06, YNKM24] in this section.

Definition 28 (Anonymous Bulletin Boards). *Let $P^{(1)}, ..., P^{(n)}$ be n parties. An anonymous bulletin board ABB among $P^{(1)}, ..., P^{(n)}$ is a functionality such that,*

- *in the writing phase, ABB takes a set of messages $X^{(j)} := \{x_1^{(j)}, ..., x_{N_j}^{(j)}\}$ as inputs from every party $P^{(j)}$ ($j \in [n]$), and*
- *in the reading phase, ABB outputs a set of messages $\Gamma := \{y_1, y_2, ..., y_{\sum_j N_j}\}$, where Γ is obtained by applying a random permutation $S_{\sum_j N_j}$ on the set $\bigcup X^{(j)}$.*

We can also define ABBs such that all elements in Γ are sorted according to a canonical ordering. A crucial property for ABB is that, by examining Γ alone, it should be impossible to determine which elements in Γ orginated from $X^{(j)}$, for any $j \in [n]$.

I Mode 3 Borda Count from MLRS

In this section we consider Mode 3 for the Borda count (a.k.a. Borda voting), where there are q candidates $C_1 ..., C_q$, and each voter is eligible to cast a sort of those q candidates as their votes. More formally, $\mathcal{E} \subseteq [q]^{*q}$. This can be regarded as the case where each voter assigns (up to q times) points to candidates based on their ranking.

Canonical Borda count forbids voters from undervoting. However, as shown in Theorem 7, undervote identification can not be guaranteed simultaneously with privacy. Therefore, in this paper we consider the variant of Borda voting where undervote is allowed (for consistency we still call it the Borda voting). In the following, we show how to achieve Mode 3 by $(1, 1)$-MLRS (cf. Appendix F for formal definitions): the vote of a voter u on candidate C_i and ranking $j_i \in [q]$ is in the form of

$$(C_i||j_i||(str_i, \sigma_i))) \,,$$

and σ_i is a $(1, 1)$-MLRS signature w.r.t. message ("VOTE:" || C_i || j_i || str_i) and dual-event label ("CANDIDATE: C_i", "RANKING: j_i").

We omit the other details, including the preparing, voting, and counting phases, since they are almost identical to the other modes and trivial. We can employ the same methods as in Mode 2 to achieve the "remove all" property by requiring each vote to include signatures for all q candidates, with a higher weight assigned to the intended candidate.

Similar as Mode 2 voting, here the above Mode 3 voting achieves computationally privacy rather than unconditional privacy, since the secret key of the underlying $(1, 1)$-MLRS is used to sign signatures under different event labels and therefore the total signing time may exceed the bound.

J Relaxed Mode 2 Voting and Simpler Constructions from k-LRS and MLRS

In this section we consider two relaxed variants of Mode 2 voting, and show simpler constructions from k-LRS and MLRS.

Variant 1 (Multi-voting without restriction). Each voter has at most k votes, and they can cast any number of votes to any candidate or abstain. We design a protocol from k-LRS as follows.

Assume a voter wants to vote on a candidate C_i ($i \in [q]$). The voter signs the message ("VOTE:" || C_i || str") under event label ("VOTE") on behalf of $\mathcal{V}$ (the assemble of all voters) and gets a k-LRS signature, where str is a randomly sampled string. Voters can vote multiple times for any candidate, as long as its total number of voting does not exceed k.

Now a vote on C_i is in the form of

$$(C_i||str||\sigma) \,.$$

Then the voter upload all the message-event-signature tuples to the anonymous bulletin board.

It is easy to see that the protocol is correct, and the (unconditionally) privacy is achieved due to the (unconditional) anonymity of k-LRS.

Variant 2 (Multi-voting with a maximum number on each candidate). Each voter has at most $k = k_1$ votes, and they can cast up to k_2 votes to any candidate or abstain. If a dishonest voter allocates more than k_2 votes to one candidate, then these repeated votes will be discarded (but other votes of this dishonest voter remain). We design a Variant 2 voting system from (k_1, k_2)-MLS scheme as follows.

Assume a voter wants to vote on some candidate C_i ($i \in [q]$). The voter generates a random string str and signs the message ("VOTE:" || C_i || str") under event labels (("VOTE"), ("CANDIDATE:" || C_i)) on behalf of $\mathcal{V}$, and gets a signature σ. The voter can repeat this up to k times.

Now a vote on C_i is in the form of

$$(C_i||str||\sigma).$$

Finally, it uploads all the message-event-signature tuples to the anonymous bulletin board.

If a malicious voter votes mote than k_1 times or votes more than k_2 times on the same candidate, then all signatures contained in the votes will be linked on the first event label or the second event label, an hence be discarded, ensuring the correctness of the protocol. Moreover, privacy is achieved due to the anonymity of (k_1, k_2)-MLRS.

Achieving Mode 2 from MLRS. We can also design a Mode 2 voting system from (w, t)-MLRS, where the i-th vote of a voter u on candidate C_{c_i} is in the form of

$$\begin{aligned}&\big(C_{c_i}||(str_{i,1}, \sigma_{i,1})||...||(str_{i,x}, \sigma_{i,x})||(str'_{i,1}, \sigma'_{i,1})||... \\ &...||(str'_{i,c_i-1}, \sigma'_{c_i-1})||(str'_{i,c_i+1}, \sigma'_{c_i+1})||...||(str'_{i,q}, \sigma'_{i,q})\big)\,,\end{aligned}$$

and

1. For $j \in [x]$, $\sigma_{i,j}$ is a (w, t)-MLRS signature w.r.t. message ("VOTE:" $||$ C_{c_i} $||$ $str_{i,j}$) and dual-event label ("VOTE", "CANDIDATE: C_{c_i}").
2. For $j \in [q] \setminus \{c_i\}$, $\sigma'_{i,j}$ is a (w, t)-MLRS signature w.r.t. message ("VOTE:" $||$ C_{c_i} $||$ $str'_{i,j}$) and dual-event label ("VOTE", "CANDIDATE: C_j").

However, in this case we have to set $(q + x - 1)k \leq w$, resulting a very large w. Therefore, we take the double-layer pattern here in order to get high efficiency.

K Deferred Proofs

K.1 Proofs of Theorems for Voting Modes

Proof (of Theorem 4 (Security of Mode 1 Voting Variant 1)). Recall that the protocol is non-interactive, and therefore, malicious voters cannot influence the result except by uploading their votes to the ABB. Obviously, the votes should satisfy the specific form defined by the rule; otherwise, they will be discarded immediately during the counting phase.

We present the following two facts to show that the counting results are correct.

1. All votes from honest voters (who vote/sign no more than k times) will remain. This is guaranteed by the randomness of str and the non-slanderability of k-LRS and LRS. Recall that if two message-event-signature tuples contain the same random string, then one of them will be discarded. While for honestly generated votes, this happens with negligible probability. Moreover, according to the non-slanderability of both k-LRS and LRS, even after seeing at most k votes from an honest voter, an attacker still cannot forge a new vote (with a non-repeated random string) that makes either $\hat{\sigma}$ or σ' in those votes or linked.
2. All votes from dishonest voters (who vote/sign more than k times) will be identified and discarded. Recall that a vote is in the form of $(str||\hat{\sigma}||\sigma')$. If a malicious voter votes mote than k times, then by the pigeonhole principle, there are at least one pair of votes whose σ' will be linked, due to the linkability of LRS. Furthermore, due to the linkability of k-LRS, if the malicious signer signs more than k times in total, then all signatures $\hat{\sigma}$ contained in the votes will be linked.[21] Moreover, thanks to the non-slanderability, malicious voters cannot invalidate votes from honest voters, as analyzed above.

The voting system has anonymity due to the anonymity of k-LRS and LRS.[22] Furthermore, the s message-event-signature tuples from a specific voter is confused among all ℓ (valid) tuples after the permutation by ABB, and no information is revealed except that the candidate gets exactly a score of ℓ, and hence the privacy is guaranteed. □

[21] Of course a voter v might sign twice using the same secret key sk'_i of LRS but the total votes v casted does not exceet k. In this case, v does not break the voting rule and hence there is no need to discard v's votes.

[22] This variant achieves computational anonymity only due to the computational anonymity of the underlying LRS scheme.

Proof (of Theorem 6 (Security of Mode 2 Voting)). The protocol is non-interactive, and therefore malicious voters cannot influence the result except by uploading their votes to the ABB. To show that the counting results are correct, we claim the following two facts.

1. All voters from honest voters will remain. This is guaranteed by the randomness of *str* and the non-slanderability property of k-LRS and t-LRS, and the fact that $k+k_2(x-1) \leq t$ (recall that $k_1 = k$). Here non-slanderability ensures that malicious voters cannot invalidate votes from honest voters by uploading any forge message-event-signature tuples and make them to be linked with honestly generated signatures.
2. All votes from dishonest voters (who vote/sign more than k times or vote/sign more than k_2 times on the same candidate) will be identified and discarded. First, due to the linkability of k-LRS, no dishonest voter can vote more than k times, as otherwise there would be $(k+1)$ votes that contain linked outer signature σ_i. Second, if one votes on the same candidate, say C, at least k_2+1 times, then there should be at least $(k_2+1)x$ inner signatures $\sigma_{i,j}$ under event label ("CANDIDATE:" $||$ C). According to the linkability of t-LRS and the fact that $(k_2+1)x > t$, all these signatures are linked, and consequently votes for other candidates (which contains at least one inner signature under event label ("CANDIDATE:" $||$ C)) from this dishonest voter can be identified and hence discarded.

The voting system has (computational) anonymity due to the anonymity of k-LRS and t-LRS. Furthermore, the message-event-signature tuples from a specific voter is confused among all (valid) tuples after the permutation by ABB. Therefore, no information is revealed except that the vote itself is valid, and hence the privacy is guaranteed. □

K.2 Proof of the Negative Result on Privacy

Proof (of Theorem 7). We prove the theorem via a counterexample. Consider a 3-voting specification where there are six candidates A, B, C, D, E, F and two voters, and each voter is expected to select exactly 3 candidates.

A distribution of a non-overvoted profile[23] can be denoted as

$$P = \begin{pmatrix} \mathsf{vote}_1^{(1)}, \mathsf{vote}_2^{(1)}, \mathsf{vote}_3^{(1)} \\ \mathsf{vote}_1^{(2)}, \mathsf{vote}_2^{(2)}, \mathsf{vote}_3^{(2)} \end{pmatrix},$$

where the first row contains the three votes of the first voter, and the second row contains the three votes of the second voter, and for $i \in \{1,2,3\}$, $j \in \{1,2\}$, $\mathsf{vote}_i^{(j)} \in \{A, B, C, D, E, F\}$ or $\mathsf{vote}_i^{(j)} = \perp$ (i.e., the j-th voter abstains from casting their i-th vote).

In this work, we consider ABB-based non-interactive setting, and each vote $\mathsf{vote}_i^{(j)}$ uploaded to ABB is attached with a signature π (or more generally, a certificate or a proof), showing the validity of $\mathsf{vote}_i^{(j)}$.

According to privacy, it holds that

$$\begin{pmatrix} (A,\pi_A), (B,\pi_B), (C,\pi_C) \\ (D,\pi_D), (E,\pi_E), (F,\pi_F) \end{pmatrix} \overset{ind.}{\approx} \begin{pmatrix} (A,\pi_A), (E,\pi_E), (C,\pi_C) \\ (D,\pi_D), (B,\pi_B), (F,\pi_F) \end{pmatrix}.$$

Namely, from the state of ABB, no adversary can distinguish the case where the first voter votes for A, B, C, and the second voter votes for D, E, F, from the case where the first voter votes for A, E, C, and the second voter votes for D, B, F.

The equation above implies that ([Sho06], Theorem 8.32)

$$\begin{pmatrix} (A,\pi_A), (B,\pi_B), & \perp \\ (D,\pi_D), (E,\pi_E), & (F,\pi_F) \end{pmatrix} \overset{ind.}{\approx} \begin{pmatrix} (A,\pi_A), (E,\pi_E), & \perp \\ (D,\pi_D), (B,\pi_B), & (F,\pi_F) \end{pmatrix}.$$

[23] If the profile is overvoted, we first modify it to a non-overvoted profile by identifying all overvotes and replacing them with empty strings in accordance with the overvote identification.

On the other hand, according to the undervote identification, if the first voter abstains from casting their third vote, then all their previous votes would be identified. Namely, if the certificated profile (from ABB) is

$$\begin{pmatrix} (A,\pi_A),\ (B,\pi_B), & \perp \\ (D,\pi_D),\ (E,\pi_E),\ (F,\pi_F) \end{pmatrix},$$

then $((A,\pi_A),(B,\pi_B))$ can be identified and hence removed, leaving a profile (D,E,F) left.

Similarly, if the certificated profile is

$$\begin{pmatrix} (A,\pi_A),\ (E,\pi_E), & \perp \\ (D,\pi_D),\ (B,\pi_B),\ (F,\pi_F) \end{pmatrix},$$

then $((A,\pi_A),(E,\pi_E))$ can be identified and hence removed, and the profile left is (D,B,F), which is clearly distinguishable from (D,E,F). That is, the distributions

$$\begin{pmatrix} (A,\pi_A),\ (B,\pi_B), & \perp \\ (D,\pi_D),\ (E,\pi_E),\ (F,\pi_F) \end{pmatrix} \text{ and } \begin{pmatrix} (A,\pi_A),\ (E,\pi_E), & \perp \\ (D,\pi_D),\ (B,\pi_B),\ (F,\pi_F) \end{pmatrix}$$

are distinguishable, which leads to a conflict. □