K-Linkable Ring Signatures and Applications in Generalized Voting

Wonseok Choi[1⋆], Xiangyu Liu[2⋆⋆], Yanxin Pang[3], Lirong Xia[4], and Vassilis Zikas[3]

[1] DGIST
[2] CISPA
[3] Georgia Tech
[4] Rutgers University
wonseok@dgist.ac.kr, xiangyu1994liu@gmail.com, yanxin@gatech.edu, lirong.xia@rutgers.edu, vzikas@gatech.edu

Abstract. *Linkable ring signatures* (LRS) allow a user to sign anonymously on behalf of a ring at most once: two signatures from the same signer are publicly identified, i.e., linked. Linkability makes LRS suitable for classical, *plurality* voting protocols—each voter casts at most one vote and the candidate with the most votes wins.

Several voting and governance 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 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.

We introduce *k-Linkable Ring Signatures* (k-LRS), an extension of standard (1-)linkable signatures, which simultaneously achieves 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 no one can 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) achieving *unconditional anonymity,* which we believe is important for voting situations which keep state indefinitely., e.g., blockchain governance mechanisms. We, then, show how k-LRS can be applied to a broad range of voting rules, including *score voting*, *multi-voting*, and *Borda*. Our voting protocols are non-interactive—each voter just posts a message on a bulletin board—which highlights the potential of k-LRS in blockchain-governance scenarios. Finally, we initiate the study of computational hardness of linking, i.e., exposing violations of k-linkability. Our study shed light to trade-offs between privacy, statefulness, and linkability of k-LRS, and as a result to their usefulness (and limitations) for digital governance.

Note on May 28. This is an updated preprint, but further revisions are ongoing according to reviewer suggestions.

⋆ Partial work was done while the author was at Purdue University and Georgia Tech.
⋆⋆ Partial work was done while the author was at Purdue University and Georgia Tech.

Table of Contents

K-Linkable Ring Signatures and Applications in Generalized Voting . . . 1
Wonseok Choi, Xiangyu Liu, Yanxin Pang, Lirong Xia, and Vassilis Zikas
1 Introduction . . . 4
1.1 Our Contributions . . . 5
1.2 Related Work . . . 9
2 Technical Overview . . . 9
3 Preliminaries . . . 15
4 k-Linkable Ring Signatures . . . 16
5 Applications in Voting Systems . . . 18
5.1 Non-Interactive Voting over Anonymous BB . . . 18
5.2 Instantiating Different Voting Rules . . . 22
5.3 GSV (Generalized Score Voting) from k-LRS . . . 24
5.4 GMV (Generalized Multi-Voting) from k-LRS and t-LRS . . . 25
6 Privacy vs. Utility . . . 26
7 Linkability Lower Bounds . . . 27
7.1 NP-Completeness of Linkability for Our Constructions . . . 27
7.2 Lower Bound of Link (in A Black-box Manner) . . . 28
7.3 Toward Reducing the Complexity of Linking . . . 29
8 Concrete Scheme from the DL/DDH Assumption . . . 30
A Limitation of Liu et al.'s LRS Scheme . . . 37
B The Insecurity of Bultel and Lafourcade's Scheme . . . 38
C Impossiblity of Traceable Ring Signatures with Unconditional Anonymity . . . 39
D Instantiating k-LRS from Aggregatable Identification Schemes . . . 40
D.1 Identification Schemes . . . 40
D.2 Aggregatable Identification Schemes . . . 41
D.3 DL-based AIS . . . 42
D.4 k-LRS from Aggregatable Identification Schemes . . . 45
E Deferred Material in Construction . . . 47
E.1 SIS-based AIS . . . 48
E.2 Security of k-LRS . . . 53
F Concrete Scheme from the SIS Assumption . . . 56
G Multi-Event LRS: Definition and Generic Construction from k-LRS . . . 57
G.1 Defnition of Multi-Event LRS . . . 57
G.2 Generic Construction from k-LRS . . . 60
H Related Work and Further Discussions . . . 62
H.1 Related Work . . . 62
H.2 On the Necessity of Stateless k-LRS and Linking All $k+1$ Signatures . . . 64
H.3 Other Applications of k-LRS . . . 64
H.4 Further Discussion on AIS . . . 65
I Anonymous Bulletin Board . . . 65
J BCV (Borda Count Voting) from MLRS . . . 65
K Relaxed GMV and Simpler Constructions from k-LRS and MLRS . . . 66
L Deferred Proofs . . . 67
L.1 Proofs of Theorems for Voting Modes . . . 67
L.2 Proof of the Negative Result on Privacy . . . 68
M Detailed Approaches of Reducing the Complexity of Linking . . . 69
N On The Hardness of Efficient Link (Overvote Identification) . . . 70

O Lower Bound of Link 71

1 Introduction

The cryptographic voting literature typically considers: *plurality voting* [Cha81, BT94a]: Each voter casts a vote for one out of many candidates, and the candidate with the most votes wins. The goal is to eliminate 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], while ensuring *(vote) privacy*: the protocol leaks nothing to the adversary beyond the outcome of the election. Other desirable properties like receipt-freeness, e.g., [AOZZ15] and public verifiability, e.g., [KZZ15b], are also guaranteed by existing protocols (cf. Section 1.2.)

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 does not guarantee vote privacy (e.g., posts votes 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 the 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 their votes even though they are using a BB (the blockchain) to communicate. At the same time in such a setting one is typically interested in non-interactive solutions (where voters post their preferences/votes and go,) and in particular in solutions which avoid making assumptions about trusted authorities.

Thus, the motivating question of our work is:

> *How can we design non-interactive, voter-private (and anonymous)*[5] *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 *all* 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.

An important consideration for our motivating application is 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 governance mechanisms relying on an anonymous-posting BB.

[5] 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 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 could 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. The anonymity of ring signatures, which, recall, was introduced to solve the problem, now becomes a problem of its own, as one cannot prevent voters from posting more than one vote/signature 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. In addition, once two 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 (1-LRS)*.

The above works for plurality, but fails for voting rules where the voter is expected to give more structured votes, e.g., indicate their approval for each candidate (a.k.a., *approval voting*). In such settings, we are interested in an 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.

1.1.1 Defining k-Linkable Ring Signature (k-LRS).

In a nutshell, a k-linkable ring signature (k-LRS) scheme is an extension of standard 1-LRS 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 under one event label—this could be a candidate's name—up to k times. ($k \in \mathbb{Z}^+$ a parameter of the scheme.) k-LRS inherits all advantages of (ordinary) LRS schemes, including no group manager, flexible ring choices, 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.[6]

Looking ahead, the literature (on 1-LRS) 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 et al.'s LRS scheme [LASZ14] in Appendix A, where a set of *two* colluding parties generate *three* signatures without being pairwise linked. This attack makes weak linkability inappropriate for our goals. We note that [BGK+21] already pointed to the insufficiency of

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

the security model in [LASZ14] (and [LW06, ACST06]) for capturing such a situation; this gave an indication that there might be an attack, but no attack was provided. To our knowledge, ours is the first work to provide an actual linkability attack on [LASZ14] confirming the concern from [BGK+21], making it an independent contribution to the (1-)LRS literature.

Notwithstanding, motivated by this definitional gap pointed out in [BGK+21] follow-up work on (1-)LRS [BKP20, BGK+21, XLAZ24] already adopted a stronger linkability property: any set of c signers/public-keys cannot generate more than c ($= c \cdot k$) signatures (even when colluding) without being pairwise detected/linked. Our definition of k-linkability extends the latter, stronger paradigm to the detection of violation by sets of $k+1$ (instead of 2) signatures.

Devising a formal definition for unconditional k-anonymity turns out to be more trickier than its computational one. Informally, the reason is that due to the computational nature of strong linkability, it is very challenging to devise a definition that preserves anonymity for signers who produce more than k signatures, even if these are on different events (and therefore do not trigger an automatic linkage.) We call this situation a *benign over-signing*. In fact, we conjecture that an unconditional definition which preserves the anonymity of benign over-signers is unimplementable. In contrast, k-LRS with computational anonymity does not suffer from this restriction. This asymmetry, which to our knowledge is noticed for the first time in our work, is reflected to both our definitions and our schemes: On the definitional side, our unconditional anonymity definition allows the adversary to break anonymity if he triggers (even) such a benign over-signing. On the instantiation side, we provide schemes which are unconditionally anonymous when such benign over-signing does not occur.

1.1.2 Instantiations of k-LRS.

Equipped with the definition of k-LRS, we proceed to instantiations. We provide instantiations based on the discrete logarithm (DL) assumption or 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 DL-based construction, our instantiations achieve unconditional anonymity and strong linkability. Our proposed schemes are stateless and do not require a trusted ceremony for generating the k-LRS setup (parties can generate their keys locally) which makes them particularly suitable for the application we are interested in. 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.

	Unforgeab.	**Linkab.**	**Non-Sland.**	**Anony.**	**Assump.**
k-LRS-DL (Scheme 1)	✓	✓	✓	uncond.	DL & signing $\leq k$ times
				comp.	DDH
k-LRS-SIS (Scheme 2)	✓	✓	✓	uncond.	SIS & signing $\leq k$ times
				comp.	SIS

1.1.3 Applications in Voting and Social Choice Theory.

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 Section 5 and associated constructions. Our constructions focus on score voting and multi-voting which are popular in social choice theory (see Section 2 for an overview). Our (non-interactive) constructions adopt the following paradigm: Each voter posts an appropriate set of k-linkable signatures to the ABB, and the outcome is computed on the state of ABB after everyone has posted.

The underlying idea of our constructions is to abstract different scoring rules as instances of a "bins-and-balls" game: there is a certain number of bins and each voter has several balls (potentially of different

color) which they can throw in the bins. Intuitively, combinations of bins and color corresponds to events (as used in k-linkable signatures) and the number of balls of each color that a voter can use corresponds to the linkability parameter—i.e., to how many signatures the voter can issue for the respective event, without triggering a linkabiltiy violation. The way bins and colors are used for different voting rules depends on the rule. Consider, for example, score voting like the one commonly used in conferences to decide on best-paper awards: each voter/PC-member has a budget of ℓ votes an can allocate it to papers at will. In this context, the bins correspond to the, say L, candidate-papers for the award, there is only one color, and each voter can use ℓ balls of this color. Looking ahead, this rule can be privately implemented by using ℓ-LRS for L events, each corresponding to one of the paper candidates. The novelty of our constructions lies in devising appropriate instantiations of the above bins-and-balls game, i.e., (combinations of) events, colors, and number of balls, to achieve the different desired voting rules.

We instantiate three classes of voting rules (modes) as an instance of the above balls-and-bins paradigm, that capture a wide range of rules used in social choice: Generalized Score-Voting (GSV), Generalized Multi-Voting (GMV), and the Borda Count Voting (BCV). In GSV, each voter can vote their preference as an integer score (where there is only a single candidate). In GMV, each voter is eligible to cast their votes up to k_1 times with a restriction that at most $k_2 \leq k_1$ votes are allowed to be cast for each candidate. Finally, BCV (a.k.a. Borda voting), is considered one of the most generic and expressive rules in social choice theory; here, there are q candidates $C_1 \ldots, C_q$, and each voter is eligible to cast, as its vote, a sorted sequence of them.

Our results are summarized in Table 2 where: For GSV and GMV, k is the parameter of the underlying k-LRS scheme (or $\boldsymbol{k}$ for BCV 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 GSV and GMV is unconditional and follows directly from the unconditional anonymity of (k-)LRS. All our schemes have (computational) public verifiability. With regards to privacy, for GSV we get unconditional privacy, whereas for GMV, the level of privacy (unconditional or computational) depends on how we set the parameter k of our k-LRS. We note that due to the restriction posed by the unconditional anonymity definition, vanilla use of unconditional k-LRS might render the classical setup for k-LRS a one-time setup when used for an unconditional voting scheme. The reason is that if the same keys are used for multiple execution of the voting protocol (i.e., with different events in the k-LRS definition) we will run in the benign over-signing issue discussed in Section 1.1.1. To solve that we can overlay the following setup-refreshing mechanism to the protocols: We add an extra key, in addition to the k-LRS key used to create a ballot. This new refresh can be used to certify his setup (voting and refresh keys) for the next election.

Table 2. Our non-interactive ABB-based voting protocols. The last column reports the complexity of detecting violations of the voting rules—by using the k-linkability property.

	Security		**Pub.-Ver.**	**Detect. Complexity**
	Anony.	Privacy		
GSV, Sec. 5.3 (e.g., score voting)	uncond.	uncond. ($k = \mathcal{O}(k_1)$)	✓	$\binom{nk_1}{k_1+1}$
GMV, Sec. 5.4 (e.g., multi-voting)	uncond.	uncond. ($k = \mathcal{O}(k_1^2 + k_1 q)$)	✓	$\binom{nk_1}{k_1+1} + \mathcal{O}(nk_1 q)$
		comp. ($k = \mathcal{O}(k_1 + k_2 + 1)$)		$\binom{nk_1}{k_1+1} + \mathcal{O}(n(k_1 + q))$
BCV, App. J (e.g., Borda count)	comp.	comp. ($\boldsymbol{k} = (1, 1)$, MLRS)	✓	$\binom{nq}{2}$

For BCV, it is not hard to see that it can be obtained by using a GMV scheme and this would preserve unconditional anonymity. However we provide a much simpler instantiation, albeit where both the privacy the anonymity are computational. Our simple BCV scheme uses an extention extension of k-LRS, termed *multi-event* $\boldsymbol{k}$-LRS (in short, $\boldsymbol{k}$-MLRS) *with parameter/vector* $\boldsymbol{k}$,) which we propose and implement. Informally, $\boldsymbol{k}$-

MLRS with parameter/vector $\boldsymbol{k}$ incorporates multiple events into the definition of (k-)LRS (see Appendix G.) Using $\boldsymbol{k}$-MLRS we also extend the applicability of our method to additional voting rules corresponding to refinements of the above. These includes multi-voting without a cap of how many of its k votes a voter can cast to each candidate and multi-voting where different candidates have different caps (see Appendix K).

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 their 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.

We further show that k-LRS and MLRS can be applied in some other cases, including trial browsing of content [TFS04], k-times anonymous authentication, and anonymous veto [BL16].

Remark 1 (On receipt freeness). Receipt freeness is a desirable property for voting schemes that has received considerable attention as it allows voters to "lie" about who they voted for in ways that are hard to detect. This adds a degree of incoercibility to the schemes. However, we are not aware of any stateless, setup-free scheme that is also receipt-free, and as observed by Alwen et al. [AOZZ15] this would likely be impossible. Since our schemes are in this realm they also do not achieve such receipt-freeness. Revising the schemes (under an appropriate setup) to support receipt-freeness, in our opinion, is an excellent research direction.

1.1.4 Privacy vs Utility in Non-interactive Voting

Three properties that are useful when tallying ballots in a voting schemes for generalized voting rules are: privacy, the ability to identify under-voting, and the ability to identify over-voting. Our voting protocols have optimal behavior with respect to certain properties that are relevant for generalized voting rules.

More concretely, our (unconditionally) private schemes:

- 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 5 and 7, these are not shortcomings of our schemes, but they apply to any non-interactive unconditionally private voting scheme over an ABB without additional setup. In particular we prove that privacy, under-vote identification, and over-vote identification cannot be satisfied simultaneously (Theorem 5) and that traceable ring signature [FS07]—the natural adaptation of ring signatures to allow for identification of (the public key of) a party that performs over-signing—and unconditionally anonymity are mutually exclussive (Theorem 7 in Appendix C.) The latter theorem relies on a connection between (k-)LRS and the notion of traceable signatures [FS07], which may be of independent interest.

Theorem 1. *(Informally, Theorem 5) For anonymous bulletin board (ABB)-based non-interactive voting protocols, it is impossible to achieve privacy and undervote identification simultaneously.*

Theorem 2. *(Theorem 7) Traceable ring signatures cannot achieve unconditional anonymity.*

1.1.5 The Complexity of Linking

One potential concern when deploying a k-LRS scheme in a voting system that produces a total of ℓ ballots is the computational cost of linkage detection. In the worst case, determining whether there exists a $(k+1)$-linked subset among ℓ signatures requires invoking the Link algorithm $\binom{\ell}{k+1}$ times, which becomes impractical for large k. Although our primary contributions are the formalization of k-linkability, the resolution of the weak linkability problem, and the feasibility of achieving privacy-preserving non-interactive voting via ABB,

it is important to examine whether this computational overhead is intrinsic or merely an artifact of our construction.

Notably, all existing linkable ring signature schemes (i.e., 1-LRS schemes) that we are aware of—e.g., [LWW04, LW05, TW05, ACST06, LASZ14, SALY17, DV09, LAZ19, HKSS22]—employ conceptually similar Link procedures as our k-LRS which search though all $(k+1)$-size sets (for $k = 1$). In these schemes, detecting linkage among ℓ signatures requires $\binom{\ell}{2} = \mathcal{O}(\ell^2)$ pairwise invocations of the linking algorithm. One might argue that this quadratic cost already makes LRS unscalable, an argument that applies (and, depending on k, is amplified) also for k-LRS; but the gaining popularity of LRS in the blockchain space suggests that for small k and for representative governance settings (where voting power is delegated) (1-)LRS (and possibly $\mathcal{O}(1)$-LRS might be useful tools). Notwithstanding, this recurring design pattern which is common in LRS strongly suggests that the combinatorial search cost is not incidental, but rather reflects a structural limitation inherent to linkable ring signatures. This brings up the following question:

> *What is the computational lower bound for linkability? Is the $\binom{\ell}{k+1}$ complexity specific to our construction, or is it a structural property shared by all LRS schemes?*

In Section 7 we initiate the investigation of the linking-complexity of (k-)LRS and prove first results towards answering the above question which suggest that barring a complete paradigm shift in (k-)LRS the above inefficiency is inherent. To this direction, we first prove that, for our schemes, identifying a $(k+1)$-linked subset among ℓ signatures is NP-complete, thereby establishing a lower bound of $\binom{\ell}{k+1}$ in the worst case under standard complexity assumptions (see Section 7.1). We then extend the argument in Section 7.2 to show that this combinatorial complexity is unavoidable for all *stateless* k-LRS constructions under natural class of linking algorithms. Therefore, the $\binom{\ell}{k+1}$ complexity is not an artifact of our specific construction, but an inherent trade-off resulting from the stronger privacy properties provided by stateless k-LRS schemes. Along the way, we identify and interesting artifact of the traditional LRS anonymity definition which makes it unsuitable for stateful schemes.

1.2 Related Work

We briefly discuss k-Time Traceable Ring Signatures (k-TRS) [BL16] and Many-Time Linkable Ring Signatures (M-LRS) [TNL+25], which also aim to support bounded times of anonymous signing but differ in the formalization. Additional details are deferred to Appendix H.1.

Informally, k-TRS can be viewed as a traceable extension of our k-LRS, i.e., once over-signing is detected, the real identity of the signer is revealed as in traceable ring signatures [FS07]. The construction in [FS07] follows the "using LRS k times" paradigm (see the technical overview below) and therefore inherits the drawback of being stateful. Moreover, the k-TRS scheme proposed in [BL16] is insecure, and we show a concrete attack in Appendix B.

In a concurrent work [TNL+25] Tran et al. introduced T-LRS, pursuing the same goal of k-times anonymous authentication as our k-LRS. However, T-LRS do not achieve the full k-linkability considered in this work, since the scheme can only identify two of the $k+1$ signatures, rather than revealing all signatures generated by that signer. Consequently, it is more closely aligned with the "using LRS k times" paradigm and is likewise stateful. As discussed in the introduction, the notion of k-linkability adopted in this work is particularly important in voting scenarios, where full accountability of malicious participants is required.

2 Technical Overview

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 $((sk_1^{(j)}, pk_1^{(j)}), \ldots, (sk_k^{(j)}, pk_k^{(j)})$ for LRS, and then sets its secret key and public key as

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

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 $(pk_1^{(1)}, \ldots, pk_k^{(1)}, \ldots, pk_1^{(n)}, \ldots, pk_k^{(n)})$. 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 H.2 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[7]) 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 with generator g. 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 $g_{\mathsf{e}} := H(\mathsf{e})$.[8] Moreover, according to the decisional Diffie-Hellman (DDH) assumption, given the public key g^x and the event-oriented 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

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

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

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 $sk_1 := x_1, \ldots, sk_{k+1} := x_{k+1}$, and aggregates them into one public key

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

where $g_1, \ldots, g_{k+1}$ 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 $(x_1, \ldots, x_k)$ 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 $(x_1, \ldots, x_{k+1})$ and $pk := \prod_{i\in[k+1]} g_i^{x_i}$, it is infeasible to compute a distinct $(x'_1, \ldots, x'_{k+1}) \neq (x_1, \ldots, x_{k+1})$ 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 $g_{\mathsf{e}}^{x_2}, \ldots, g_{\mathsf{e}}^{x_k}$, 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 $x_1, \ldots, x_{k+1}$ 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 $f_{\mathsf{e}}(sk_1, \ldots, sk_k)$ to a probabilistic algorithm, and introduce k random coefficients $a_1, \ldots, a_k \in \mathbb{Z}_q^*$ when signing. Therefore,

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

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

The signing algorithm is now stateless. For each signing, the signer samples a new group of coefficients $(a_1, \ldots, a_k)$ independently. From k signatures, which contain k groups of coefficients and k pseudo-identities, one can recover $g_{\mathsf{e},1}^{x_1}, \ldots, g_{\mathsf{e},k}^{x_k}$ consequently, as long as all these k groups of coefficients form an invertible 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-oriented public keys*. With such a function $f_{\mathsf{e}}(sk_1, \ldots, sk_k)$, 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-oriented public keys $epk_1, \ldots, epk_k$.

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. We have the following two definitions for linkability.

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

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, 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 $cmt := g^r$ with $r \in \mathbb{Z}_q$, 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

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

where $ch_0 := H(cmt^{(1)}, \ldots, cmt^{(n)}, \mathsf{msg})$, 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

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

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-oriented commitment (i.e., it is related to event e). The verification algorithm checks if the following equations hold.

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

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

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

For signer δ to sign, it has to fix other $(n-1)$ members' challenges before determining cmt, as otherwise it cannot offer a response rsp verifying Eq. (1), 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 $g_{\mathsf{e}}^{x_1^{(\delta)}}$.[9] 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.

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

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 $ch^{(1)}, \ldots, ch^{(n)}$. 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 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 of the form

$$\begin{aligned}\sigma := (pid, &(cmt^{(1)}, ecmt^{(1)}, ch^{(1)}, rsp_1^{(1)}, rsp_2^{(1)}),\\ &\ldots, (cmt^{(n)}, ecmt^{(n)}, ch^{(n)}, rsp_1^{(n)}, rsp_2^{(n)})).\end{aligned}$$

The verification algorithm checks if the following holds for every $j \in [n]$:[10]

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

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

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

The j-th challenge is determined by the $(j-1)$-th commitment, and all n challenges and commitments form a ring connected by $H(\cdot)$. 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 for 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 (Appendix. D). Except for the DL-based construction, we give another instantiation of $(k+1)$-AIS from the short integer solution (SIS) assumption (Appendix E.1).

Applications 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 GSV 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

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

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.

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

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 e, i.e., a signer can sign up tp k times without being linked under a particular 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 G for the formal definition).

Let M be the number of event indexes (i.e., how many different bounds). In a $(k_1, \ldots, k_M)$-MLRS scheme, a signature is generated under M events $\mathsf{e}_1, \ldots, \mathsf{e}_M$. 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 $(k_1, \ldots, k_M)$-MLRS from k-LRS schemes.

Generalized Multi-Voting (GMV): Removing All Property. If there are multiple events (e.g., for score votes, each event is defined with each candidate), GSV 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 five candidates A, B, C, D, E, and each voter is allowed to select three candidates without repetition (e.g., it is not allowed to select A twice). One vote for A is in the form of

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

where

1. $\sigma_{A,1}, \ldots, \sigma_{A,3}$ are three 5-LRS signatures under event "A" (see below why we need three), 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 $\sigma_{A,1}, \ldots, \sigma_{A,3}, \sigma_B$, $\sigma_C, \sigma_D, \sigma_E$ 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".[11]

For a more general case where each voter gets k_1 votes and at most $k_2 \leq k_1$ 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 $k = k_1$, and x, t according to $k_2 x + (k - k_2) \leq t$ 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.

Privacy vs. Utility, and the Inefficiency of Link. After addressing the issue of overvotes, we further consider if it is feasible to identify undervoting (i.e., one casts less than k votes and then abstains). Unfortunately, it appears impossible to detect such behavior if we aim to preserve privacy. We formally analyze this limitation in Sec. 6. Note 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.

We argue in Sec. 7.2 that the super-linear complexity of identifying a linked set using Link is inherent to stateless schemes. We first show that stateless linkable signature schemes satisfy a stronger privacy notion, termed *subset anonymity*, which ensures that signatures reveal no additional information about the signer's state. To formalize the adversary's limited view, we introduce the linkage finding game and the notion of myopic linkers. We prove that, in the ideal world, any linker given only signature indices requires exponential time to identify a linked set, and that in the real world, linkers with access to at most $k + 1$ signatures can achieve only a negligible advantage over ideal-world adversaries in polynomial time. Consequently, for any stateless linkable signature scheme, exactly determining a linkage set necessarily requires exponential time.

3 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter. For $a, b \in \mathbb{Z}$ with $a < b$, define $[a, b] := \{a, a+1, \ldots, b\}$. 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 a distribution $\mathcal{S}$, 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 $\Delta(X, Y) := 1/2 \sum_{s \in \mathcal{S}} |\Pr[X = s] - \Pr[Y = s]|$.

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.[12] 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[13]);

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

[12] Ranked voting depends only on which of two candidates is preferred by a voter.

[13] Range voting was originally proposed with scores of 0 or 1, and it can be a range of integers without loss of generality.

– *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.

4 k-Linkable Ring Signatures

Let $N \in \mathbb{N}^+$ is the total number of users in the system, and define the set of public keys $PK_R := \{pk^{(j)}\}_{j \in R}$ for a ring R.

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).*
- $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$*: 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.*
- $1/0 \leftarrow \mathsf{Link}(\mathsf{e}, \sigma_1, \ldots, \sigma_{k+1})$*: The link algorithm that takes as input an event label e and $(k+1)$ signatures $\sigma_1, \ldots, \sigma_{k+1}$, 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*

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

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 $\mathsf{msg}_1, \ldots, \mathsf{msg}_{k+1}$, it holds that*

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

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

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

Remark 2 (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. tag, prefix in [BEHM22, HC24], or scope in [HKSS22]). 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 3 (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. See Section 1.1 for a high-level description of these security properties.

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-*LRS *be a k-LRS scheme. We say k-*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.*

Experiment $\mathsf{Exp}^{unforg}_{\mathcal{A},k}(\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda, k)$
2. **for each** $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
3. $\mathcal{S} := \mathcal{S}_{corr} := \varnothing$
4. $(R^*, \mathsf{e}^*, \mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdot,\cdot,\cdot,\cdot),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$
5. **if** $R^* \subseteq [N] \wedge R^* \cap \mathcal{S}_{corr} = \varnothing \wedge (R^*, \mathsf{e}^*, \mathsf{msg}^*, \cdot) \notin \mathcal{S}$
6. $\quad \wedge\ \mathsf{Ver}(PK_{R^*}, \mathsf{e}^*, \mathsf{msg}^*, \sigma^*) = 1$:
7. $\quad$ **return** 1
8. **return** 0

Oracle $\textsc{Sign}(R, \delta, \mathsf{e}, \mathsf{msg})$:

9. **if** $R \not\subseteq [N] \vee \delta \notin R$: **return** $\perp$
10. $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
11. $\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}, \mathsf{msg}, \sigma)\}$
12. **return** σ

Oracle $\textsc{Corr}(j)$:

13. $\mathcal{S}_{corr} := \{j\}$
14. **return** $sk^{(j)}$

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

Definition 3 (k-unconditional anonymity of k-LRS). *Let k-*LRS *be a k-LRS scheme. We say k-*LRS *has*

- *computational 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 adversaries $\mathcal{A}$, where* $\mathsf{Exp}^{anony}_{\mathcal{A},k}(\lambda)$ *is defined in Fig. 2 without the gray parts;*
- *k-unconditional 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 (even computationally unbounded) adversaries $\mathcal{A}$, where* $\mathsf{Exp}^{anony}_{\mathcal{A},k}(\lambda)$ *is defined in Fig. 2 with the gray parts, and $\mathcal{A}$ is restricted to that $\mathcal{T}[\delta] \leq k$ for all $\delta \in \mathcal{S}_{ch}$.*

Informally, a k-LRS scheme achieves k-unconditional anonymity, if an unbounded adversary obtain no more than k signatures from any challenge user $\delta \in \mathcal{S}_{ch}$ across all event labels, via Sign or Chall. Throughout this paper, the term unconditional anonymity refers specifically to the notion of k-unconditional anonymity defined here.

Remark 4 (Malicious key generation). For the simplicity of notion we do not model malicious key generation in our definitions. A generic approach is to require a user, upon registering a new public key, to provide a zero-knowledge proof of the corresponding secret key. Furthermore, our two concrete constructions (cf. Section D) are based on the DDH assumption and the SIS assumption, respectively, whose corresponding zero-knowledge proofs have been extensively studied in the literature (e.g., [Sch91, Ste96, Lyu08]).

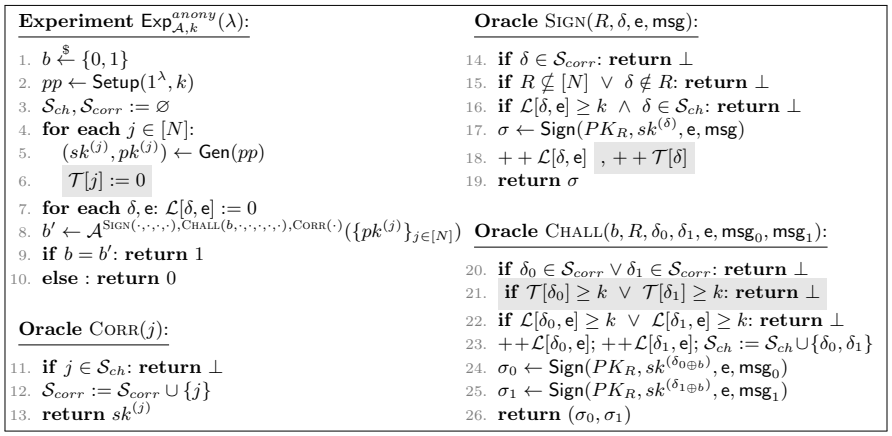

Fig. 2. The (k-unconditional) anonymity experiment of k-LRS, where the highlighted codes are only shown in the k-unconditional anonymity experiment.

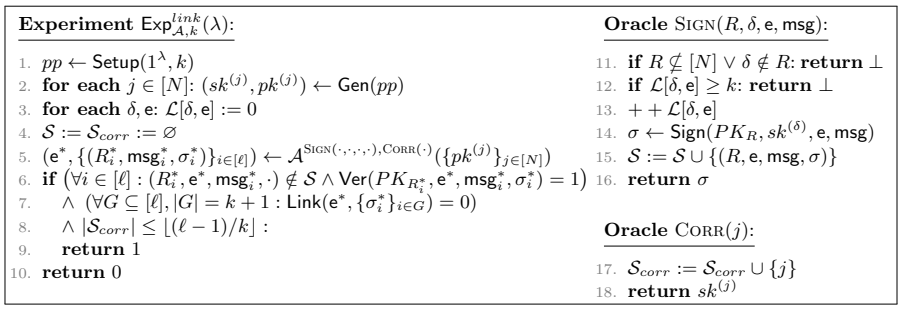

Fig. 3. The linkability experiment of k-LRS.

Definition 4 (Linkability of k-LRS). *Let k-*LRS *be a k-LRS scheme. We say k-*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.*

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 $k = 1$).

Definition 5 (Non-slanderability of k-LRS). *Let k-*LRS *be a k-LRS scheme. We say k-*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.*

5 Applications in Voting Systems

5.1 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

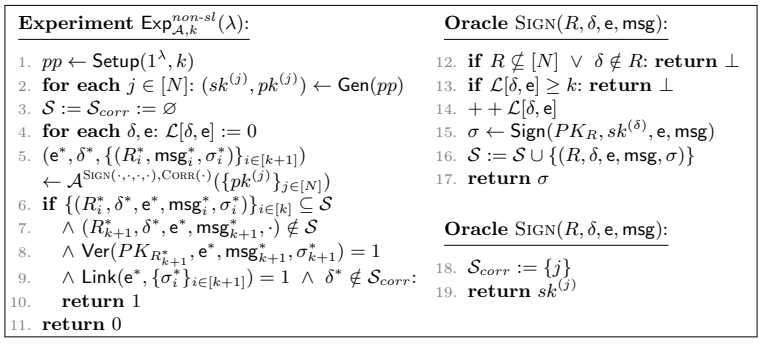

Fig. 4. The non-slanderability experiment of k-LRS.

ABB. We recall the definition of ABB in Appendix I. 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 $\mathsf{S}(\cdot) : \mathcal{E}^n \mapsto \mathcal{D}$ 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 $P := (\mathsf{vote}^{(1)}, \ldots, \mathsf{vote}^{(n)}) \in \mathcal{E}^n$ 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 $\mathsf{vote} \in \mathcal{E} \Leftrightarrow \mathsf{Hw}(\mathsf{vote}) = 1$ (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. $\triangleleft$

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., $\mathcal{E} \to (\mathcal{E}')^k$ 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 $\mathcal{E} = [0, k] \to [0, 1]^k$ 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, $\mathcal{E} = [0, k]^q$ so that it can be transformed as $\mathcal{E} = [0, 1]^{(k \cdot q)}$. 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 $[k]$. $\triangleleft$

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" $P := (\mathsf{vote}^{(1)}, \ldots, \mathsf{vote}^{(n)})$ 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. $\triangleleft$

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*

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

where λ is the security parameter, and

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

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. $\triangleleft$

Definition 11 (Privacy of voting protocols). *Let $\Gamma := (\mathcal{V} := [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a k-voting specification (i.e., $\mathsf{vote}^{(j)} := (\mathsf{vote}_1^{(j)}, \ldots, \mathsf{vote}_k^{(j)})$ 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*

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

where λ is the security parameter, and

$$P_{[i,j,G]} = \begin{pmatrix} (\mathsf{vote}_1^{(1)\prime}, \ldots, \mathsf{vote}_k^{(1)\prime}) \\ \cdots \\ (\mathsf{vote}_1^{(n)\prime}, \ldots, \mathsf{vote}_k^{(n)\prime}) \end{pmatrix} \quad s.t. \quad \begin{cases} \mathsf{vote}_\ell^{(i)\prime} = \mathsf{vote}_\ell^{(j)}, \ \ell \in [G] \\ \mathsf{vote}_\ell^{(j)\prime} = \mathsf{vote}_\ell^{(i)}, \ \ell \in [G] \\ \mathsf{vote}_\ell^{(\iota)\prime} = \mathsf{vote}_\ell^{(\iota)}, \ \textit{otherwise.} \end{cases}$$

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. ◁

5.2 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.

GSV (Generalized 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 GSV 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 $\mathcal{E} = [0, k]$ or $\mathcal{E} = [0, 1]^k$.

GMV (Generalized 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 GSV by letting $k_1 := q \cdot k_2$. While the parallel execution of GSV can only verify overvotes for a single candidate (thus preserving the anonymity of the dishonest voter's ballots for other candidates), GMV 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.

Unconditional anonymity inherently bounds the number of signatures that can be issued per key across different event labels, and it degrades to computational anonymity once a key has been used for many voting instances. This limitation can be addressed by incorporating a setup-refresh mechanism to preserve unconditional anonymity. More precisely, each user maintains both a k-LRS secret key for voting and a 1-LRS secret key for key updates. At the end of each voting period, the user generates two new key pairs—one for the next voting event and one for the subsequent update. The user then signs these two public keys using the 1-LRS key and uploads them to the ABB, thereby completing one round of key refresh.

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 $(k_1, \ldots, k_M)$-MLRS schemes (Appendix E.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 (Section 6). 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.

5.3 GSV (Generalized Score Voting) from k-LRS

We design a GSV 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 $0 \le s \le k$ 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 a predetermined event label, e.g., ("2025 ELECTION"), on behalf of $\mathcal{V}$.

Let $(\sigma_1, \ldots, \sigma_s)$ 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 GSV has unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for GSV has correctness and (unconditionally) privacy.*

Proof (of Theorem 3 (Security of GSV)). 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. □

5.4 GMV (Generalized Multi-Voting) from k-LRS and t-LRS

In this subsection we construct GVM from k-LRS and t-LRS. For better understanding our approach, we first design a scheme with computational privacy only, and then explain how to extend it to achieve unbounded privacy.

Suppose there are q candidates $C_1, \ldots, C_q$. Let $k := k_1$ and $x, t \in \mathbb{N}^+$ such that $k_1 + k_2(x-1) \leq t$ and $(k_2+1)x > t$. At a high level, we use k-LRS and t-LRS and a voter needs to sign on the desired candidate x times. 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 GSV and GMV is how to generate ballots that contain signatures. In GSV, a ballot can be divided into parts, with each part only containing one signature, and they look independent of each other. However, in GMV, 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 GSV, 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 6 (On the necessity of using distinct random strings). We emphasize that in GMV 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 L.1.

Theorem 4. *If the underlying k-LRS and t-LRS schemes in GMV have unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for GMV has correctness and privacy.*

One drawback of the above GMV 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. 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 GMV, we consider the scheme in the computational anonymity setting.

6 Privacy vs. Utility

The GMV protocol (Section 5.4) achieves almost all desirable properties of an ABB-based non-interactive voting scheme, except that it permits undervoting, i.e., a malicious voter may cast fewer than the required k votes. In this section, we formally show that this limitation is inherent, and that our protocol in fact achieves optimality. We establish a fundamental trilemma: privacy, overvote prevention, and undervote prevention cannot be achieved simultaneously in the non-interactive ABB setting.

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 $C_1, \ldots, C_q$ be q candidates of the voting event.

Definition 12 (Overvote and undervote). *Let $\Gamma := (\mathcal{V} := [n], \mathcal{E}, \mathcal{D}, \mathsf{S})$ be a k-voting specification (i.e., a well-formed vote is in the form of $\mathsf{vote}^{(u)} := (\mathsf{vote}_1^{(u)}, \ldots, \mathsf{vote}_k^{(u)})$ 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 $\mathsf{vote}^{(u)} := (\mathsf{vote}_1^{(u)}, \ldots, \mathsf{vote}_{k'}^{(u)}) \in P$ 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 5. *There exist no voting protocol Π, which achieves overvote identification, undervote identification, and privacy, simultaneously. The statement is true even if we settle for computational privacy.*

The proof is deferred to Appendix L.2.

Remark 7. The negative result in Theorem 5 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 $\mathsf{vote} := (\mathsf{vote}_1, \ldots, \mathsf{vote}_k)$, 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 8. 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.

7 Linkability Lower Bounds

We next turn our attention to the complexity of linking. We prove that the dependence on $\binom{\ell}{k+1}$ is inherent both for our scheme, and, under common assumptions on the way the linking algorithm work, for any stateless (k-)LRS scheme, unless a fundamentally different design approach is employed from what is currently being used in the literature.

7.1 NP-Completeness of Linkability for Our Constructions

For our concrete constructions, we formally show the hardness of such efficient "Link" by reducing it to an NP-complete coding problem.

We first abstract the problem as follows:

Definition 13. *Let $\mathbb{F}$ be a field and integers $n > k$. Given n groups $(\mathbf{a}_i, z_i) \in \mathbb{F}^n \times \mathbb{F}$, return a subgroup $S \subseteq [n]$ such that $|S| = k + 1$, and there exists $\mathbf{b} \in \mathbb{F}^k$ such that $\langle \mathbf{a}_i, \mathbf{b} \rangle = z_i$ holds for all $i \in S$. If such S does not exist, return 0.*

We provide the whole proof of reduction from the following NP-complete problem to our previously abstracted problem in Appendix N.

Definition 14. *Let $G \in \mathbb{F}^{(k+1)\times n}$ be a generator matrix for a linear code*

$$\mathcal{C} := \{\mathbf{y}^\top G \mid \mathbf{y} \in \mathbb{F}^{k+1}\},$$

and let $d < n$ be an integer. Return a non-zero codeword $\mathbf{c}$ such that $\mathsf{Hw}(\mathbf{c}) \leq d$, or return 0 if such codeword does not exist. Here $\mathsf{Hw}(\mathbf{c})$ denote the Hamming weight (the number of non-zero coordinates of a vector) of $\mathbf{c} \in \mathbb{F}^n$.

7.2 Lower Bound of Link (in A Black-box Manner)

Despite the apparent advantage of eliminating state and auxiliary memory, stateless linkable ring signatures in fact achieve a strictly stronger notion of anonymity. Intuitively, because a stateless signer maintains no local information across invocations, the resulting signatures cannot encode or leak information about the signer's current state or the states of other users. To formalize this intuition, we introduce a strengthened privacy notion, which we call *subset anonymity*. This notion captures the setting in which an adversary may interact adaptively with signing and challenge oracles but is only allowed to observe a subset of the generated signatures. Subset anonymity models the limited and potentially fragmented view available to adversaries in realistic environments and precisely characterizes the privacy guarantees offered by stateless schemes beyond standard anonymity.

Definition 15 (Subset (k-Unconditional) anonymity of k-LRS). *Let k-LRS be a k-LRS scheme. We say k-LRS has*

- *subset computational anonymity, if $\mathsf{Adv}^{sub-anony}_{\mathcal{A},k}(\lambda) := |\Pr[\mathsf{Exp}^{sub-anony}_{\mathcal{A},k}(\lambda) = 1] - 1/2| \leq negl(\lambda)$ holds for all PPT adversaries $\mathcal{A}$, where $\mathsf{Exp}^{sub-anony}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 5 without the gray parts;*
- *subset k-unconditional anonymity, if $\mathsf{Adv}^{sub-anony}_{\mathcal{A},k}(\lambda) := |\Pr[\mathsf{Exp}^{sub-anony}_{\mathcal{A},k}(\lambda) = 1] - 1/2| \leq negl(\lambda)$ holds for all (even computationally unbounded) adversaries $\mathcal{A}$, where $\mathsf{Exp}^{sub-anony}_{\mathcal{A},k}(\lambda)$ is defined in Fig. 5 with the gray parts, and $\mathcal{A}$ is restricted to that $\mathcal{T}[\delta] \leq k$ for all $\delta \in \mathcal{S}_{ch}$.*

The subset anonymity experiment refines the standard anonymity game by explicitly modeling partial visibility through the view flag in oracle queries. Only when $\mathsf{view} = \mathsf{True}$ does the adversary obtain access to the corresponding signatures, and internal counters ensure that no signer exceeds the k-linkability bound for a given event. As a result, the adversary's information is restricted to a subset of all generated signatures, even though it may adaptively choose rings, events, and messages. This restriction is essential for capturing the privacy guarantees of stateless constructions, in which no persistent information can be carried across signing sessions.

We show in Appendix O that subset anonymity is strictly stronger than the standard notion of (k-unconditional) anonymity. In particular, while all stateless k-LRS schemes satisfying (k-unconditional) anonymity also satisfy subset anonymity, the converse does not hold for stateful schemes. Stateful constructions may exploit internal memory to correlate signatures across invocations, thereby leaking information once only a subset of signatures is revealed.

Lemma 1. *If a stateless k-LRS scheme satisfies (unconditional) anonymity, then it also satisfies subset (unconditional) anonymity. This implication does not hold in general for stateful schemes.*

While statelessness strengthens privacy guarantees, it also introduces an inherent drawback in terms of link efficiency. In particular, the absence of state precludes the use of auxiliary information to accelerate linkage detection. As a consequence, identifying a single $(k+1)$-oversigned subset among ℓ signatures may require $\mathcal{O}(n^{k+1})$ time in the worst case. Since adversaries are always allowed to query the linking algorithm, the amount of information they can retain from past interactions becomes a critical factor in determining the feasibility of linkage discovery.

To formalize this limitation, we introduce a restricted class of adversaries called *myopic linkers*. These adversaries are characterized by a bounded view of the system: they may only retain a limited amount of

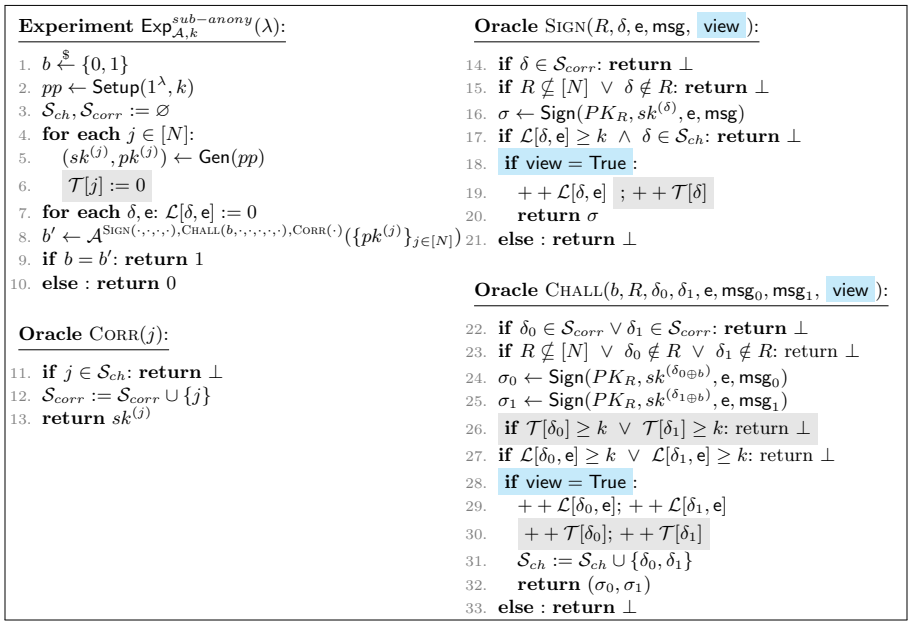

Fig. 5. The subset (k-unconditional) anonymity experiment of k-LRS, where the highlighted codes are only shown in the unconditional anonymity experiment. Blue highlighted codes are the difference between the subset (k-unconditional) anonymity experiment and the original anonymity experiment.

information from previous oracle interactions, reflecting a constrained observational capability. The objective of a myopic linker is to identify linked subsets among the observed signatures using the linking oracle. Full formal definitions and proofs are deferred to Appendix O.

Lemma 2. *Provided that link queries are restricted to sets of size at most $k+1$, any myopic linker achieves at most advantage $\mathsf{Adv}^{find-link}_{\boldsymbol{\mathcal{A}},k}(\lambda)$ with an exponential number of linker invocations. In particular, no polynomial-time myopic linker can identify a linked set with non-negligible advantage.*

Together, these results demonstrate that the high complexity of linkage discovery in stateless k-LRS schemes is not an artifact of our construction, but a direct consequence of their stronger anonymity guarantees.

7.3 Toward Reducing the Complexity of Linking

We now discuss some approaches to reduce the complexity $\binom{\ell}{k+1}$ in GSV, though coming with trade-offs in functionality or security. Formal descriptions are deferred to Appendix M.

7.3.1 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 Section 2). 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 $(\hat{sk}_1, \ldots, \hat{sk}_{k+1}, sk'_1, \ldots, sk'_k)$, where $(\hat{sk}_1, \ldots, \hat{sk}_{k+1})$ is a secret key of k-LRS, and $sk'_1, \ldots, sk'_k$ 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 $f_{\mathsf{e}}(\hat{sk}_1, \ldots, \hat{sk}_k; a_1, \ldots, a_k)$ (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 overvotes 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.[14]

7.3.2 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.[15] Say, $[n]$ is divided by m subsets: $N_1, \ldots, N_m$ 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.

8 Concrete Scheme from the DL/DDH Assumption

As discussed, at high-level, in our technical overview (Section 2,) to make our constructions generic and instantiable by different assumptions, we introduce a template based on a new primitive, which we call $(k+1)$-aggregatable identification scheme (AIS). We instantiate AIS from the DL/DDH assumption or the SIS assumption. Due to page limitation, rather than providing here our detailed template we present directly the DL/DDH-based construction (i.e., the construction where AIS is replaced by its DL/DDH instantiation). We believe that this already give sufficient intuition about our methodology. The formalization of AIS, the way we use it for k-LRS and its SIS-based instantiation are described in detail in Appendices D and F.

Scheme 1. [k-LRS-DL] Let $\mathbb{G}$ is a cyclic group of order q with generator g, and $H_0 : \{0,1\}^* \mapsto \mathbb{G}$, $H : \{0,1\}^* \mapsto \mathbb{Z}_q$ 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 $\mathsf{pp} := (\mathbb{G}, q, g_1, \ldots, g_{k+1})$.
- 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 $(sk, pk) := ((x_1, \ldots, x_{k+1}), X)$.
- Sign: Let $PK_R := (X^{(1)}, \ldots, X^{(n)})$ 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 $a_1, \ldots, a_k \xleftarrow{\$} \mathbb{Z}_q^*$, compute $H_0(\mathsf{e}) = (e_1, \ldots, e_k) \in \mathbb{G}^k$, and $\widetilde{X} := \prod_{i\in[k]} e_i^{a_i x_i^{(1)}}$.
  2. $r_1, \ldots, r_{k+1} \xleftarrow{\$} \mathbb{Z}_q$, 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 $j = 2, \ldots, n$:
     (a) $\mathbb{Z}_q \ni ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, \ldots, a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$.
     (b) $z_1^{(j)}, \ldots, z_{k+1}^{(j)} \xleftarrow{\$} \mathbb{Z}_q$.
     (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. $ch^{(1)} := H(\widetilde{cmt}^{(n)}, \widetilde{ecmt}^{(n)}, a_1, \ldots, a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$.
  5. For $i \in [k+1]$: $z_i^{(1)} := r_i - x_i^{(1)} \cdot ch^{(1)}$.
  6. Return $\sigma := \left(a_1, \ldots, a_k, \widetilde{X}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}, z_1^{(j)}, \ldots, z_{k+1}^{(j)}\}_{j\in[n]}\right)$.

[14] 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.

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

– Verification: Let $PK_R := (X^{(1)}, \ldots, X^{(n)})$. For $j \in [n]$, compute $ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, \ldots, a_k, \widetilde{X}, PK_R, \mathsf{e}, \mathsf{msg})$ (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 $\left(\sigma_\iota := (a_{\iota,1}, \ldots, a_{\iota,k}, \widetilde{X}_\iota, \{\widetilde{cmt}_\iota^{(j)}, \widetilde{ecmt}_\iota^{(j)}, z_{\iota,1}^{(j)}, \ldots, z_{\iota,k+1}^{(j)}\})\right)_{\iota\in[k+1]}$ be $k+1$ signatures for the same event e (note that different σ_ι may contain different rings). Let $H_0(\mathsf{e}) = (e_1, \ldots, e_k)$. If there exists a solution $(epk_1, \ldots, epk_k) \in \mathbb{G}^k$ for the following linear equation system

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

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

Acknowledgments. We would like to thank the anonymous reviewers 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: 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: 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: 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: 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: Public Key Cryptography - PKC 2003, Lecture Notes in Computer Science, vol. 2567, pp. 145–160. Springer (2003)

[GV05] Guruswami, V., Vardy, A.: Maximum-likelihood decoding of reed-solomon codes is np-hard. IEEE Transactions on Information Theory vol. 51(7), pp. 2249–2256 (2005)

[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: 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 (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: 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: 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)

[Ste96] Stern, J.: A new paradigm for public key identification. IEEE Trans. Inf. Theory vol. 42(6), pp. 1757–1768 (1996)

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

[TNL+25] Tran, N., Nguyen, K., Liu, D., Pieprzyk, J., Susilo, W.: Many-time linkable ring signatures. In: ProvSec 2025, vol. 16172, pp. 3–22. Springer (2025)

[TS06] Teranishi, I., Sako, K.: k-times anonymous authentication with a constant proving cost. In: Public Key Cryptography - PKC 2006, 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)

[Var02] Vardy, A.: The intractability of computing the minimum distance of a code. IEEE Transactions on Information Theory vol. 43(6), pp. 1757–1766 (2002)

[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)

[XZC+24] Xie, J., Zhou, J., Cao, Z., Dong, X., Choo, K.K.R.: Linkable, k-times traceable and revocable ring signature for fine-grained accountability in blockchain transactions. IEEE Internet of Things Journal (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 $H_0 : \{0,1\}^* \mapsto \mathbb{G}$, $H : \{0,1\}^* \mapsto \mathbb{Z}_q^*$ are two hash functions which are modeled as random oracles.

$\underline{\mathsf{Setup}(1^\lambda)}$:

1. **return** $pp := (\mathbb{G}, q, g_1, g_2)$

$\underline{\mathsf{Gen}(\mathsf{pp})}$:

2. $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$; $X := g_1^{x_1} g_2^{x_2}$
3. **return** $(sk, pk) := ((x_1, x_2), X)$

$\underline{\mathsf{Ver}(R, PK_R, \mathsf{e}, \mathsf{msg}, \sigma)}$:

4. $g_\mathsf{e} := H_0(\mathsf{e})$
5. $cmt' := g_1^{z_1} g_2^{z_2} \prod_{j\in[n]} (X^{(j)})^{c^{(j)}}$
6. $ecmt' := g_\mathsf{e}^{z_1} \prod_{j\in[n]} (pid)^{c^{(j)}}$
7. **if** $\sum_{j\in[n]} c^{(j)} = H(PK_R, pid, cmt', ecmt', \mathsf{e}, \mathsf{msg})$:
8. **return** 1
9. **return** 0

$\underline{\mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})}$:

10. Parse $sk^{(\delta)} = (x_1, x_2)$
11. Parse $PK_R = (X^{(1)}, X^{(2)}, \ldots, X^{(n)})$
12. $g_\mathsf{e} := H_0(\mathsf{e})$; $pid := g_\mathsf{e}^{x_1}$
13. $r_1, r_2, c^{(1)}, \ldots, c^{(\delta-1)}, c^{(\delta+1)}, \ldots, c^{(n)} \xleftarrow{\$} \mathbb{Z}_q$
14. $cmt := g_1^{r_1} g_2^{r_2} \prod_{j\in[n], j\neq\delta} (X^{(j)})^{c^{(j)}}$
15. $ecmt := g_\mathsf{e}^{r_1} \prod_{j\in[n], j\neq\delta} (pid)^{(c^{(j)})}$
16. $c_0 := H(PK_R, pid, cmt, ecmt, \mathsf{e}, \mathsf{msg})$
17. $c^{(\delta)} := c_0 - \sum_{j\in[n], j\neq\delta} c^{(j)}$
18. $z_1 := r_1 - c^{(\delta)} x_1$; $z_2 := r_2 - c^{(\delta)} x_2$
19. **return** $\sigma := (pid, z_1, z_2, c^{(1)}, \ldots, c^{(n)})$

$\underline{\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2)}$:

20. Parse $\sigma_1 = (pid_1, \ldots)$ and $\sigma_2 := (pid_2, \ldots)$
21. **if** $pid_1 = pid_2$: **return** 1
22. **return** 0

Fig. 6. The LRS scheme in [LASZ14].

Theorem 6. *The LRS scheme in Fig. 6 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 $\sigma := (pid, z_1, z_2, c^{(1)}, \ldots, c^{(n)})$ 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 $\mathbb{G}_2$, $e : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_t$ is a non-degenerate bilinear map. Two hash functions are defined as $H_0 : \{0,1\}^* \mapsto \mathbb{G}_1$, and $H_2 : \{0,1\}^* \mapsto \mathbb{Z}_q^*$.

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.

$\underline{\mathsf{Setup}(1^\lambda)}$:

1. **return** $pp := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, q, g_1, g_2, e, H_0, H_1)$

$\underline{\mathsf{Gen}(pp, k)}$:

2. $x, x_1, \ldots, x_k \xleftarrow{\$} \mathbb{Z}_q^*$
3. $X := g_1^x$; $X_1 := g_1^{x_1}, \ldots, X_k := g_1^{x_k}$
4. $sk := (x, x_1, \ldots, x_k)$; $pk := (X, X_1, \ldots, X_k)$
5. **return** (sk, pk)

$\underline{\mathsf{Sign}(sk^{(j)}, PK_R, \mathsf{e}, \mathsf{msg}, i)}$: //the i-th signing of j

6. Parse $sk^{(j)} = (x^{(j)}, x_1^{(j)}, \ldots, x_k^{(j)})$
7. $r \xleftarrow{\$} \mathbb{Z}_q^*$
8. $A := H_0(\mathsf{e}, 0)$; $B := H_0(\mathsf{e}, 1)$
9. $C := H_0(\mathsf{e}, 2)$; $W := H_0(\mathsf{e}, 3)$
10. $u := H_1(\mathsf{e}, \mathsf{msg}, 0, g_2^r)$; $v := H_1(\mathsf{e}, \mathsf{msg}, 1, g_2^r)$
11. $T_1 := A^{x_i^{(j)}}$; $T_2 := B^{x_i^{(j)}} \cdot g_1^{u \cdot x^{(j)}}$
12. $T_3 := C^{x_i^{(j)}} \cdot W^{v \cdot x^{(j)}}$; $T_4 := g_2^r$
13. $T_5 := e(W, T_4)^{(x^{(j)})}$
14. $\pi \leftarrow \mathsf{NIZK.Prove}(T_1, T_2, T_3, T_4, T_5, PK_R)$
15. **return** $\sigma := (T_1, T_2, T_3, T_4, T_5, \pi)$

$\underline{\mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma)}$:

16. Parse $\sigma = (T_1, T_2, T_3, T_4, T_5, \pi)$
17. Compute A, B, C, W, u, v as Sign
18. **return** $\mathsf{NIZK.Ver}((T_1, T_2, T_3, T_4, T_5, PK_R), \pi)$

$\underline{\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2)}$:

19. Parse $\sigma_1 = (T_{1,1}, T_{1,2}, T_{1,3}, T_{1,4}, T_{1,5}, \pi_1)$
20. Parse $\sigma_2 = (T_{2,1}, T_{2,2}, T_{3,3}, T_{4,4}, T_{5,5}, \pi_1)$
21. **if** $T_{1,1} = T_{2,1}$: **return** 1
22. **return** 0

$\underline{\mathsf{Trace}(\mathsf{e}, \mathsf{msg}_1, \sigma_1, \mathsf{msg}_2, \sigma_2, \mathsf{msg}, \sigma)}$:

23. **if** $\mathsf{Link}(\mathsf{e}, \sigma_1, \sigma_2) = 0$: return $\perp$
24. Parse $\sigma_1 = (T_{1,1}, T_{1,2}, T_{1,3}, T_{1,4}, T_{1,5}, \pi_1)$
25. Parse $\sigma_2 = (T_{2,1}, T_{2,2}, T_{3,3}, T_{4,4}, T_{5,5}, \pi_1)$
26. $u_1 := H_1(\mathsf{e}, \mathsf{msg}_1, 0, T_{1,4})$; $v_1 := H_1(\mathsf{e}, \mathsf{msg}_1, 1, T_{1,4})$
27. $u_2 := H_1(\mathsf{e}, \mathsf{msg}_2, 0, T_{2,4})$; $v_1 := H_1(\mathsf{e}, \mathsf{msg}_2, 1, T_{2,4})$
28. $id := (T_{1,2}/T_{2,2})^{(u_1-u_2)^{-1}}$
29. **return** id // return a signer's identity

Fig. 7. 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 16 (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 $pp \leftarrow \mathsf{Setup}(1^\lambda, k = 1)$, $(sk^{(1)}, pk^{(1)}), \ldots, (sk^{(N)}, pk^{(N)}) \leftarrow \mathsf{Gen}(pp)$, $\sigma_1 \leftarrow \mathsf{Sign}(PK_{R_1}, sk^{(\delta_1)}, \mathsf{e}, \mathsf{msg}_1)$, $\sigma_2 \leftarrow \mathsf{Sign}(PK_{R_2}, sk^{(\delta_2)}, \mathsf{e}, \mathsf{msg}_2)$, and the probability is taken over the choice of Setup, Gen, *and* Sign.

Theorem 7. *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 $sk_1, \ldots, sk_\eta$, 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 $\{sk_1^{(0)}, \ldots, sk_\eta^{(0)}\}$ 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 9. 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 Instantiating k-LRS from Aggregatable Identification Schemes

D.1 Identification Schemes

Definition 17 (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.,* $rsp \leftarrow \mathsf{Sim}_1(pp)$*, 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*

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

where $pp, pp_1, \dots, pp_Q$ *are independent and random parameters,* sk *is a randomly sampled secret key, and* $u_1, \dots, u_Q$ *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 18 (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. 17 can be extended to output a simulated transcript* $(ecmt, ch, rsp)$ *that is statistically close to the output by an honest execution, where* $rsp \leftarrow \mathsf{Sim}_1(pp)$*, and* $ecmt \leftarrow \mathsf{Sim}_2(pp, epk, ch, a)$*.*

D.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 19 ($(k+1)$-Aggregatable identification schemes). *Let* $k \in \mathbb{N}^+$*. A* $(k+1)$*-aggregatable identification scheme is defined as an identification scheme in Def. 17 with the following additional properties.*

– ***Aggregation***. *For* $(k+1)$ *valid key pairs* $(sk_1, pk_1), \dots, (sk_{k+1}, pk_{k+1})$ *under* $(k+1)$ *different parameters* $pp_1, \dots, pp_{k+1}$*, there exist two deterministic algorithms* AggrPK *and* $\mathsf{AggrCMT}$ *satisfying the following properties.*
  - $\widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, \dots, pk_{k+1})$*. The algorithm* AggrPK *aggregates all public keys into a public key* $\widetilde{pk}$ *via operation "+", i.e.,* $\widetilde{pk} := \sum_{i \in [k+1]} pk_i$*.*
  - $\widetilde{cmt} \leftarrow \mathsf{AggrCMT}(cmt_1, \dots, cmt_{k+1})$*. The algorithm* $\mathsf{AggrCMT}$ *aggregates all commitments into a commitment* $\widetilde{cmt}$*, where* $cmt_i \leftarrow \mathsf{Commit}(pp_i; \rho_i)$ *for all* $i \in [k+1]$*.*
  - $pp := (pp_1, \dots, pp_{k+1})$ *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* $sk := (sk_1, \dots, sk_{k+1})$ *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* $rsp := (rsp_1, \dots, rsp_{k+1})$*. 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, $sk := (sk_1, \ldots, sk_{k+1})$ *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* $pp := (pp_1, \ldots, pp_{k+1})$ *is randomly sampled. Fixing any* $sk_1, \ldots, sk_k$, *if* sk_{k+1} *is distributed uniformly, then* $\widetilde{pk}$ *has a distribution statistically close to the uniform distribution over* $\widetilde{\mathcal{PK}}$, *where* $\widetilde{pk} \leftarrow \mathsf{AggrPK}(pk_1, \ldots, pk_{k+1})$, *and* $pk_i := f_{pp_i}(sk_i)$ *for* $i \in [k+1]$.

Definition 20 (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 21 (Homomorphism for AIS). *We say a* k*-AIS is homomorphic on coefficient space* $\mathbb{F}$, *if it additionally satisfies the following.*

– *For any coefficients* $a_1, \ldots, a_k \in \mathbb{F}$, *any* k *valid key pairs* $(sk_1, pk_1), \ldots, (sk_k, pk_k)$ *under* k *different parameters* $pp_1, \ldots, pp_k$, *let* $(\widetilde{cmt}, ch, rsp := (rsp_1, \ldots, rsp_k))$ *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* $\widetilde{cmt} \leftarrow \mathsf{AggrCMT}(cmt_1, \ldots, cmt_k)$. *Then there is a coefficient-verification algorithm* $\mathsf{CoeVer}(pp, \widetilde{epk}, (\widetilde{ecmt}, ch, rsp), a_1, \ldots, a_k)$ *that outputs the same result as the verification algorithm* $\mathsf{Ver}(pp, \widetilde{pk}, (\widetilde{cmt}, ch, rsp))$, *where* $\widetilde{epk} \leftarrow \mathsf{AggrPK}(a_1 \cdot pk_1, \ldots, a_k \cdot pk_k)$, *and* $\widetilde{ecmt} \leftarrow \mathsf{AggrCMT}(\mathsf{Commit}(pp_1; a_1 \cdot \rho_1), \ldots, \mathsf{Commit}(pp_k; a_k \cdot \rho_k))$.
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* $rsp \leftarrow \mathsf{Sim}_1(pp)$, *and* $\widetilde{ecmt} \leftarrow \mathsf{Sim}_2(pp, \widetilde{epk}, ch, a_1, \ldots, a_k)$.

– ***Infeasiblity of kernel***. *For randomly sampled* $\{pk_i\}_{i\in[k]}$, *it is computationally infeasible to find a group of efficients* $(a_1, \ldots, a_k) \in \mathbb{F}^k$ *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* $\mathcal{PK}$.

D.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 22 (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 23 (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 24 (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 $Q = poly(\lambda)$.

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

Now, we show the first $(k+1)$-AIS scheme from the DL assumption in Fig. 8. Here $pp := (\mathbb{G}, q, g)$ defines $\mathcal{SK} := \mathbb{Z}_q$, $\mathcal{PK} := \mathbb{G}$, and the extended secret key space $\ddot{\mathcal{SK}} := \mathcal{SK} := \mathbb{Z}_q$.

$\underline{\mathsf{Setup}(1^\lambda)}$:	$\underline{\mathsf{Gen}(pp)}$:
$(\mathbb{G},q,g) \leftarrow \mathsf{GGen}(1^\lambda)$	$x \xleftarrow{\$} \mathbb{Z}_q$; $X := g^x$
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. 8. The DL-based identification scheme [Sch91].

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. 8 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. 9.

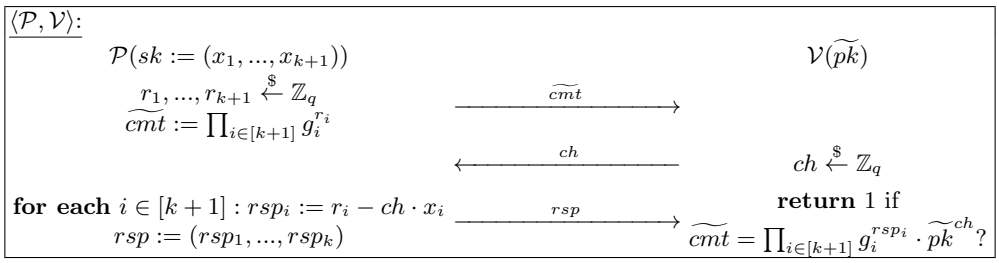

Fig. 9. The aggregation of the DL-based identification scheme.

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. $\mathsf{Sim}_1(pp)$ 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 $\widetilde{ecmt} \stackrel{?}{=} \prod_{i\in[k]} g_i^{rsp_i a_i} \cdot \widetilde{epk}^{ch}$. 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. $\mathsf{Sim}_1(pp)$ 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 $\widetilde{ecmt} := \prod_{i\in[k]} (g_i^{rsp_i})^{a_i} \cdot \widetilde{epk}^{ch}$.

□

D.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 $\mathcal{H}_0 : \{0,1\}^* \mapsto \mathcal{PP}^k$ and $\mathcal{H} : \{0,1\}^* \mapsto \mathcal{CH}$ 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 $pp := (pp_1, \ldots, pp_{k+1})$.
- Key Generation: Let $(sk_i, pk_i) \leftarrow \mathsf{AIS.Gen}(pp_i)$ for $i \in [k+1]$, and $\widetilde{pk} \leftarrow \mathsf{AIS.AggrPK}(pk_1, \ldots, pk_{k+1})$. Return the secret key $\mathsf{sk} := (sk_1, \ldots, sk_{k+1})$ and the public key $\mathsf{pk} := \widetilde{pk}$.
- Sign: Let $(\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(n)})$ be the public keys of n users in the ring. W.l.o.g., we assume the signer's index is 1. Parse $\mathsf{sk}^{(1)} := (sk_1^{(1)}, \ldots, sk_{k+1}^{(1)})$. To sign on message msg under event label e, the signer generates the signature as follows.
  1. Let $\mathcal{H}_0(\mathsf{e}) = epp := (epp_1, \ldots, epp_k) \in \mathcal{PP}^k$. For $i \in [k]$, compute $epk_i := f_{epp_i}(sk_i^{(1)})$.
  2. Sample $a_1, \ldots, a_k \xleftarrow{\$} \mathbb{F}$, and compute $\mathsf{epk} \leftarrow \mathsf{AIS.AggrPK}(a_1 \cdot epk_1, \ldots, a_k \cdot epk_k)$.

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 $\widetilde{cmt}^{(1)} \leftarrow \mathsf{AIS.AggrCMT}(cmt_1^{(1)}, \ldots, cmt_{k+1}^{(1)})$ and $\widetilde{ecmt}^{(1)} \leftarrow \mathsf{AIS.AggrCMT}(ecmt_1^{(1)}, \ldots, ecmt_k^{(1)})$.
4. For $j \in [2, n]$: 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.,
$rsp^{(j)} := (rsp_1^{(j)}, \ldots, rsp_k^{(j)}, rsp_{k+1}^{(j)}) \leftarrow \mathsf{AIS.Sim}_1(ch^{(j)})$,
$\widetilde{cmt}^{(j)} \leftarrow \mathsf{AIS.Sim}_2(pp, \mathsf{pk}^{(j)}, ch^{(j)}, rsp^{(j)})$,
$\widetilde{ecmt}^{(j)} \leftarrow \mathsf{AIS.Sim}_2(epp, \mathsf{epk}, ch^{(j)}, (rsp_1^{(j)}, \ldots, rsp_k^{(j)}), a_1, \ldots, a_k)$.
5. Let $ch^{(1)} := \mathcal{H}(\widetilde{cmt}^{n}, \widetilde{ecmt}^{n}, \{a_j\}_{j\in[k]}, \mathsf{epk}, \{\mathsf{pk}^{(\iota)}\}_{\iota\in[n]}, \mathsf{msg}, \mathsf{e})$.
Compute $rsp^{(1)} \leftarrow \mathsf{AIS.Response}((sk_1^{(1)}, \ldots, sk_{k+1}^{(1)}), ch^{(1)}, (\rho_1, \ldots, \rho_{k+1}))$.
6. Return $\sigma := (a_1, \ldots, a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$.

– 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
- $\mathsf{AIS.CoeVer}(epp, \mathsf{epk}, (\widetilde{ecmt}^{(j)}, ch^{(j)}, rsp_{|k}^{(j)}), a_1, \ldots, a_k) = 1$,

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

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

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

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

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

– Verification on user P_1 (the signer). Let $(\rho_1, \ldots, \rho_k, \rho_{k+1})$ 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 $\widetilde{cmt}^{(1)} \leftarrow \mathsf{AggrCMT}(\mathsf{Commit}(pp_1; \rho_1), \ldots, \mathsf{Commit}(pp_{k+1}; \rho_{k+1}))$. According to the correctness of AIS, with overwhelming probability $(\widetilde{cmt}, ch, rsp^{(1)} := (rsp_1^{(1)}, \ldots, rsp_{k+1}^{(1)}))$ is a valid transcript under $pp := (pp_1, \ldots, pp_{k+1})$ 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 $epp := (epp_1, \ldots, epp_k)$ maps the first k secret keys $(sk_1^{(1)}, \ldots, sk_k^{(1)})$ into a group of *event-related public keys* $(epk_1, \ldots, epk_k)$. Due to the homomorphism of AIS, introducing the coefficients $a_1, \ldots, a_k$ does not affect the correctness of verification. Namely,

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

will return 1 if

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

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

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

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

– If all these signatures are signed by the same signer holding secret key $\mathsf{sk} := (sk_1, \ldots, sk_k, sk_{k+1})$, then its event-related public keys are $(epk_1, \ldots, epk_k) := (f_{epp_1}(sk_1), \ldots, f_{epp_k}(sk_k))$. 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, $\mathsf{Link}(\mathsf{e}, \sigma_1, \ldots, \sigma_{k+1})$ will return 1.

– On the other hand, assume all these signatures are signed by two different users holding secret keys $\mathsf{sk}^{(1)} := (sk_1^{(1)}, \ldots, sk_k^{(1)}, sk_{k+1}^{(1)})$ and $\mathsf{sk}^{(2)} := (sk_1^{(2)}, \ldots, sk_k^{(2)}, sk_{k+1}^{(2)})$, 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 $(epk_1^{(1)}, \ldots, epk_k^{(1)}) := (f_{epp_1}(sk_1^1), \ldots, f_{epp_k}(sk_k^{(1)}))$ of user 1, from $(a_{\ell,i})_{\ell,i \in [k]}$ and $(\mathsf{epk}_1, \ldots, \mathsf{epk}_k)$. Meanwhile, we have $\sum_{i \in [k]} a_{k+1,i} \cdot epk_i^{(2)} = \mathsf{epk}_{k+1}$, where $(epk_1^{(1)}, \ldots, epk_k^{(1)}) := (f_{epp_1}(sk_1^{(1)}), \ldots, f_{epp_k}(sk_k^{(1)}))$ are the event-related public keys of user 2. If $\mathsf{Link}(\mathsf{e}, \sigma_1, \ldots, \sigma_{k+1})$ 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 $(a_{k+1,1}, \ldots, a_{k+1,k})$ such that $\sum_{i \in [k]} a_{k+1,i} \cdot (epk_i^{(2)} - epk_i^{(1)}) = 0_{\mathcal{PK}}$. Namely, if $(\sigma_1, \ldots, \sigma_{k+1})$ are signed by at least two different users, then with overwhelming probability $\mathsf{Link}(\mathsf{e}, \sigma_1, \ldots, \sigma_{k+1})$ 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., it has unforgeability, unconditional or computational anonymity, linkability, and non-slanderability.*

We defer Appendix E.2 for the security proof.

E Deferred Material in Construction

In this section we provide the SIS-based construction of $(k+1)$-AIS and the proof of Theorem 5.

E.1 SIS-based AIS

Definition 25 (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 3 (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. 10. Here $pp := \mathbf{A}$ defines $\mathcal{SK} := \{0,1\}^m$, $\mathcal{PK} := \mathbb{Z}_q^n$, and the extended secret key space $\bar{\mathcal{SK}} := \{\mathbf{x} \in \mathbb{Z}_q^m \mid \|\mathbf{x}\| \leq 10m^{1.5}\}$. SAFE is defined as $\{1, \ldots, 10m-1\}^m$. For the purpose of aggregation, in this work, we change the sampling of $\mathbf{r}^{[\![\iota]\!]}$ ($\iota \in [t]$) from $\{0, 1, \ldots, 10m-1\}^m$ instead of $\{0, 1, \ldots, 5m-1\}^m$ 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.

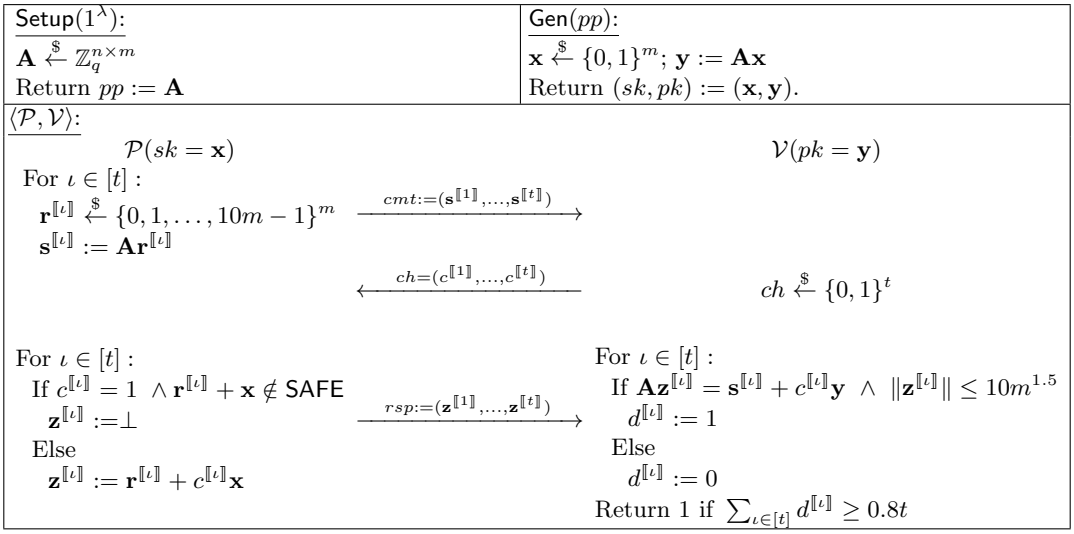

Fig. 10. The SIS-based identification scheme [Lyu08]

Theorem 11. *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. 10 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,

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

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

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

One-wayness. Let $pp = \mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $sk = \mathbf{x} \in \ddot{\mathcal{SK}} = \{\mathbf{x} \in \mathbb{Z}_q^m \mid \|\mathbf{x}\| \leq 10m^{1.5}\}$. Define $f_{pp}(sk) = \mathbf{Ax}$. 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 $(\mathbf{A}, \mathbf{Ax})$ 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 $\mathbf{Ax} = \mathbf{Ax}'$. 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 3, $\mathbf{Ar}^{[\![\iota]\!]}$ 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 $((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$ and $((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]'}, \ldots, c^{[\![t]\!]'}), (\mathbf{z}^{[\![1]\!]'}, \ldots, \mathbf{z}^{[\![t]\!]'}))$ 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 $\left\|\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}\right\| \leq \left\|(\mathbf{z}^{[\![\iota^*]\!]})\right\| + \left\|\mathbf{z}^{[\![\iota^*]\!]'}\right\| \leq 20m^{1.5}$.

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

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

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 $ch := (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}) \in \{0,1\}^t$ 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 $\mathbf{s}^{[\![\iota]\!]} := \mathbf{Ar}^{[\![\iota]\!]}$, where $\mathbf{r}^{[\![\iota]\!]} := (\mathbf{r}^{[\![\iota]\!]}[1], \ldots, \mathbf{r}^{[\![\iota]\!]}[m])$, and $\mathbf{r}^{[\![\iota]\!]}[\zeta] := 0$ if $b^{[\![\iota]\!]}[\zeta] = 0$ and $\mathbf{r}^{[\![\iota]\!]}[\zeta] \xleftarrow{\$} \{1, \ldots, 10m-1\}$ 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 $\mathbf{z}^{[\![\iota]\!]} \xleftarrow{\$} \{1, \ldots, 10m-1\}^m$ (the SAFE range), and $\mathsf{Sim}_2(pp, pk, ch, rsp)$ outputs $\mathbf{s}^{[\![\iota]\!]} := \mathbf{Az}^{[\![\iota]\!]} - \mathbf{y}$;
   (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 $\mathbf{s}^{[\![\iota]\!]} := \mathbf{Az}^{[\![\iota]\!]}$ with $\mathbf{r}^{[\![\iota]\!]} \xleftarrow{\$} \{0, \ldots, 10m-1\}^m$.
3. The simulated transcript is $(cmt = (\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), ch, rsp = (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$.

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 $\{1, \ldots, 10m-1\}$ if $\mathbf{x}[\zeta] = 0$ or $\{0, \ldots, 10m-2\}$ if $\mathbf{x}[\zeta] = 1$. With high probability, there are very few unsafe points among total m choices, which means that $\mathbf{r} \in Y \subseteq \{0, \ldots, 10m-1\}^m$ with $|Y|$ very large. Then, according to Lemma 3, $\mathbf{Ar}$ 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 $\{1, \ldots, 10m-1\}$, 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 $\mathbf{A}, \mathbf{A}_1, \ldots, \mathbf{A}_Q \in \mathbb{Z}_q^{n \times m}$ be two random matrices, and let $\mathbf{x} \in \{0,1\}^m$ be a secret key sampled uniformly. By applying Lemma 3 on the matrix $\widetilde{\mathbf{A}} := (\mathbf{A}^\top \| {\mathbf{A}_1}^\top \| \ldots \| {\mathbf{A}_Q}^\top)^\top \in \mathbb{Z}_q^{Qn \times m}$, $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{A}}x)$ 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:

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

where $\mathbf{u}_1, \ldots, \mathbf{u}_Q \xleftarrow{\$} \mathbb{Z}_q^n$.[16]

Homomorphism on coefficient space $\mathbb{Z}_q^*$. Given $a \in \mathbb{Z}_q^*$ and a transcript $(cmt, ch, rsp) := ((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$ where $(\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}) := (\mathbf{Ar}^{[\![1]\!]}, \ldots, \mathbf{Ar}^{[\![t]\!]})$, 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 $ecmt := (a\mathbf{s}^{[\![1]\!]}, \ldots, a\mathbf{s}^{[\![t]\!]}) = (\mathbf{A}(a\mathbf{r}^{[\![1]\!]}), \ldots, \mathbf{A}(a\mathbf{r}^{[\![t]\!]}))$.

1. For $\iota \in [t]$: if $a\mathbf{Az}^{[\![\iota]\!]} = (a\mathbf{s}^{[\![\iota]\!]}) + c^{[\![\iota]\!]}(a\mathbf{y})$ 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.

1. Let $ch := (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}) \in \{0,1\}^t$ 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 $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{Ar}^{[\![\iota]\!]}$, where $\mathbf{r}^{[\![\iota]\!]} := (\mathbf{r}^{[\![\iota]\!]}[1], \ldots, \mathbf{r}^{[\![\iota]\!]}[m])$, and $\mathbf{r}^{[\![\iota]\!]}[\zeta] := 0$ if $b^{[\![\iota]\!]}[\zeta] = 0$ and $\mathbf{r}^{[\![\iota]\!]}[\zeta] \xleftarrow{\$} \{1, \ldots, 10m-1\}$ if $b^{[\![\iota]\!]}[\zeta] = 1$;
   (b) else if $c^{[\![\iota]\!]} = 1$, $\mathsf{Sim}_1(ch)$ outputs $\mathbf{z}^{[\![\iota]\!]} \xleftarrow{\$} \{1, \ldots, 10m-1\}^m$ (the SAFE range), and $\mathsf{Sim}_2(pp, pk, ch, rsp, a)$ outputs $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{Az}^{[\![\iota]\!]} - (a\mathbf{y})$;
   (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 $a\mathbf{s}^{[\![\iota]\!]} := a\mathbf{Az}^{[\![\iota]\!]}$ with $\mathbf{r}^{[\![\iota]\!]} \xleftarrow{\$} \{0, \ldots, 10m-1\}^m$.
3. The simulated transcript is $(ecmt := (a\mathbf{s}^{[\![1]\!]}, \ldots, a\mathbf{s}^{[\![t]\!]}), ch, rsp := (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$.

□

Aggregation. Define $\mathsf{SAFE} := \{0, 1, \ldots, 10m-1\}^{km+m}$. For $i \in [k+1]$, let $pp_i := \mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. Define $pp := (pp_1, \ldots, pp_{k+1}) = \mathbf{A} := (\mathbf{A}_1 \| \ldots \| \mathbf{A}_{k+1}) \in \mathbb{Z}_q^{n \times (km+m)}$. 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.

[16] 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.

– $\mathsf{AggrPK}(pk_1, \ldots, pk_{k+1})$ 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 $cmt_i := (\mathbf{A}_i\mathbf{r}_i^{[\![1]\!]}, \ldots, \mathbf{A}_i\mathbf{r}_i^{[\![t]\!]})$ with $\mathbf{r}_i^{[\![\iota]\!]} \xleftarrow{\$} \{0, 1, \ldots, 10m-1\}^m$ for $\iota \in [t]$. $\mathsf{AggrCMT}(cmt_1, \ldots, cmt_{k+1})$ returns $\widetilde{cmt} := (\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}) = (\sum_{i\in[k+1]} \mathbf{A}_i\mathbf{r}_i^{[\![1]\!]}, \ldots, \sum_{i\in[k+1]} \mathbf{A}_i\mathbf{r}_i^{[\![t]\!]})$.
– $pp := (pp_1, \ldots, pp_{k+1})$ defines the 3-move identification protocol in Fig. 11.

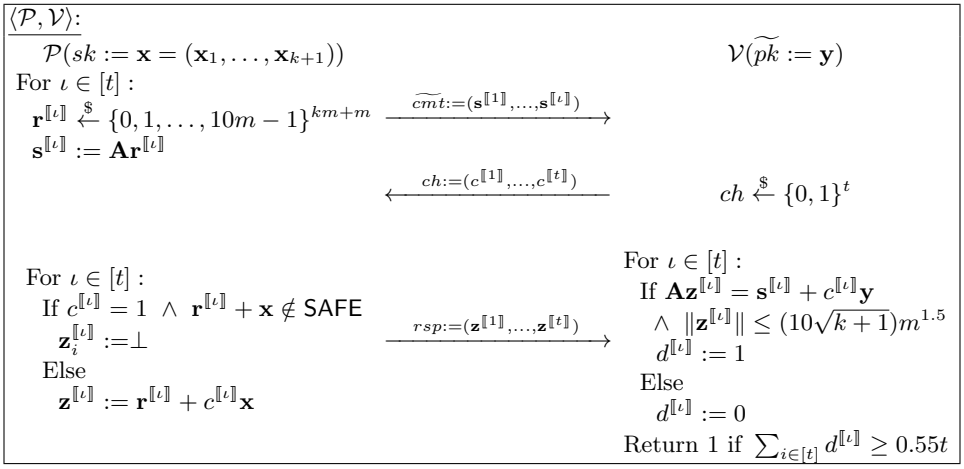

Fig. 11. The aggregation of the SIS-based identification scheme

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 $(\mathbf{A}_1\|\mathbf{A}_2\|\ldots\|\mathbf{A}_{k+1}) \in \mathbb{Z}_q^{n\times(km+m)}$ as the parameter $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$.

Theorem 12. *Assume $k \le 4$, $m \ge 10$, $k \ll n$, $m = Q\lceil n \log n\rceil$ 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,

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

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

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

One-wayness. Let $pp := \mathbf{A} \in \mathbb{Z}_q^{n\times(km+m)}$ and $sk := \mathbf{x} \in \ddot{\mathcal{SK}} = \{\mathbf{x} \in \mathbb{Z}_q^{km+m} \mid \|\mathbf{x}\| \le (10\sqrt{k+1})m^{1.5}\}$. Define $f_{pp}(sk) := \mathbf{Ax}$. 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 $(\mathbf{A}, \mathbf{Ax})$ 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 $\mathbf{Ax} = \mathbf{Ax}'$. 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 $\widetilde{cmt} := (\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}) = (\sum_{i\in[k+1]} \mathbf{A}_i \mathbf{r}_i^{[\![1]\!]}, \ldots, \sum_{i\in[k+1]} \mathbf{A}_i \mathbf{r}_i^{[\![t]\!]})$, where $\mathbf{r}_i^{[\![\iota]\!]} \leftarrow \{0,1,\ldots,5m-1\}^m$ for all $i \in [k+1]$ and $\iota \in [t]$. According to Lemma 3, $\mathbf{A}_i \mathbf{r}_i^{[\![\iota]\!]}$ 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 $((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$ and $((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]'}, \ldots, c^{[\![t]\!]'}), (\mathbf{z}^{[\![1]\!]'}, \ldots, \mathbf{z}^{[\![t]\!]'}))$ 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 $\left\|\mathbf{z}^{[\![\iota^*]\!]} - \mathbf{z}^{[\![\iota^*]\!]'}\right\| \leq \left\|(\mathbf{z}^{[\![\iota^*]\!]})\right\| + \left\|\mathbf{z}^{[\![\iota^*]\!]'}\right\| \leq (20\sqrt{k+1})m^{1.5}$.

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 3.[17]

Collision resistance of secret key. Given randomly sampled $\mathbf{A} := (\mathbf{A}_1 \| \ldots \| \mathbf{A}_{k+1}) \in \mathbb{Z}_q^{km+m}$, if the adversary finds two distinct extended secret keys $\mathbf{x} := (\mathbf{x}_1; \ldots; \mathbf{x}_{k=1}), \mathbf{x}' := (\mathbf{x}'_1; \ldots; \mathbf{x}'_{k+1}) \in \mathbb{Z}_q^{km+m}$ 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 3, 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 $\mathbf{x}_1, \ldots, \mathbf{x}_k$, if $sk_{k+1} = \mathbf{x}_{k+1}$ 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 $a_1, \ldots, a_k \in \mathbb{Z}_q^*$, a transcript $(cmt, ch, rsp) := ((\mathbf{s}^{[\![1]\!]}, \ldots, \mathbf{s}^{[\![t]\!]}), (c^{[\![1]\!]}, \ldots, c^{[\![t]\!]}), (\mathbf{z}^{[\![1]\!]}, \ldots, \mathbf{z}^{[\![t]\!]}))$ where $\mathbf{s}^{[\![\iota]\!]} := \sum_{i\in[k]} \mathbf{s}_i^{[\![\iota]\!]} = \sum_{i\in[k]} \mathbf{A}_i \mathbf{r}_i^{[\![\iota]\!]}$ and $\mathbf{z}^{[\![\iota]\!]} := (\mathbf{z}_1^{[\![\iota]\!]}, \ldots, \mathbf{z}_k^{[\![\iota]\!]})$ for $\iota \in [t]$, the coefficient-verification algorithm $\mathsf{CoeVer}(pp, \widetilde{epk}, (\widetilde{ecmt}, ch, rsp), a_1, \ldots, a_k)$ works as follows. Here

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

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

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

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 $\left\|\mathbf{z}^{[\![\iota]\!]}\right\| \leq 10m^{1.5}$, 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 $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{Z}_q^n$, if $k \ll n$, then with high probability $\mathbf{y}_1, \ldots, \mathbf{y}_k$ are linear independent, which means that there exits no $a_1, \ldots, a_k \in \mathbb{Z}_q^*$ such that $\sum_{i\in[k]} a_i \mathbf{y}_i = 0^n$. □

Remark 10 (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.

E.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., 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 $\mathsf{G}_0, \ldots, \mathsf{G}_3$. 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 $\mathsf{sk}^{(\delta)} := (sk_1^{(\delta)}, \ldots, sk_k^{(\delta)}, sk_{k+1}^{(\delta)})$, the challenger $\mathcal{C}$ works as follows.

1. Let $epp := (epp_1, \ldots, epp_k) := \mathcal{H}_0(\mathsf{e})$. For $i \in [k]$, compute $epk_i := f_{epp_i}(sk_i^{(\delta)})$.
2. $a_1, \ldots, a_k \xleftarrow{\$} \mathbb{F}$, and compute $\mathsf{epk} := \mathsf{AggrPK}(a_1 \cdot epk_1, \ldots, a_k \cdot epk_k)$.
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 $(\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(n)})$ be the public keys of n users in R. For $j = \delta+1, \ldots n, \ldots, \delta-1$, 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 $(\widetilde{ecmt}^{(j)}, ch^{(j)}, rsp_{|k}^{(j)})$.
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 $\sigma := (a_1, \ldots, a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$. 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 $\textsc{Sign}$ 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 $\mathcal{H}(\widetilde{cmt}^{\delta-1}, \ldots)$ 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 $\mathsf{sk}^{(\delta)} := (sk_1^{(\delta)}, \ldots, sk_k^{(\delta)}, sk_{k+1}^{(\delta)})$ is used only for generating the event-oriented public keys. Furthermore, only the first k parts of $\mathsf{sk}^{(\delta)}$ are required.

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

G_3 and G_2 are indistinguishable due to the following equations.

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

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 $\ddot{\mathsf{sk}}^{(\delta)} := (\ddot{sk}_1^{(\delta)}, \ldots, \ddot{sk}_{k+1}^{(\delta)}) \in \ddot{\mathcal{SK}}^{k+1}$ 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 $\sigma^* := (a_1^*, \ldots, a_k^*, \mathsf{epk}^*, \{\widetilde{cmt}^{*(j)}, \widetilde{ecmt}^{*(j)}\}_{j \in R^*}, \{rsp^{*(j)}\}_{j \in R^*})$. 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 $ch^{*(\delta)\prime}$ to $\mathcal{A}$. According to the forking lemma [BN06], with non-negligible probability $\mathcal{A}$ will return another valid transcript $(\widetilde{cmt}^{*(\delta)}, ch^{*(\delta)\prime}, rsp^{*(\delta)\prime})$ under pp and $\mathsf{pk}^{(\delta)}$. Meanwhile, both $ch^{*(\delta)}$ and $ch^{*(\delta)\prime}$ 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 $(sk_1^{(0)}, \ldots, sk_k^{(0)})$ (used for signing msg_0) and $(sk_1^{(1)}, \ldots, sk_k^{(1)})$ (used for signing msg_1). However, due to the uniformity of $(k+1)$-AIS, fixed the first k parts of the secret key, $\widetilde{pk} \leftarrow \mathsf{AggrPK}(f_{pp_1}(sk_1^{(b)}), \ldots,$

$f_{pp_k}(sk_k^{(b)}), f_{pp_{k+1}}(sk_{k+1}^{(b)}))$ distributes uniformly as long as $sk_{k+1}^{(b)}$ distributes uniformly ($b \in \{0,1\}$). Therefore, the probability $(sk_1^{(0)}, \ldots, sk_k^{(0)})$ is linked to $\mathsf{pk}^{(\delta_0)}$ and the probability $(sk_1^{(1)}, \ldots, sk_k^{(1)})$ 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 SIGN and challenge oracle 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 $\text{SIGN}(R, \delta, \mathsf{e}, \mathsf{msg})$ or a query $\text{CHALL}(b, R, \delta_0, \delta_1, \mathsf{e}, \mathsf{msg}_0, \mathsf{msg}_1)$, $\mathcal{C}$ 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 $(sk_1^{(\delta)}, \ldots, sk_k^{(\delta)})$, the challenger $\mathcal{C}$ independently samples a group of public keys $(epk_{\mathsf{e},1}^{\delta}, \ldots, epk_{\mathsf{e},k}^{(\delta)})$, and uses them to generate the event-oriented public key $\mathsf{epk} := \mathsf{AggrPK}(a_1 \cdot epk_{\mathsf{e},1}^{(\delta)}, \ldots, a_k \cdot epk_{\mathsf{e},k}^{(\delta)})$. Meanwhile, $\mathcal{C}$ records $(epk_{\mathsf{e},1}^{\delta}, \ldots, epk_{\mathsf{e},k}^{(\delta)})$ 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 $\mathcal{H}_0(\mathsf{e}^*) = (epp_1, \ldots, epp_k)$. Now, $\mathcal{A}$'s goal is to generate a valid forgery $\sigma^* := (a_1, \ldots, a_k, \mathsf{epk}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}\}_{j\in[n]}, \{rsp^{(j)}\}_{j\in[n]})$ 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 $ch^{(\delta)\prime}$. Then again according to the forking lemma [BN06], with non-negligible probability $\mathcal{A}$ will return another valid transcript $(\widetilde{cmt}^{(\delta)}, ch^{(\delta)\prime}, rsp^{(\delta)\prime})$ 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 $\mathsf{Exp}_{\mathcal{A},k}^{link}(\lambda)$. If $\ddot{\mathsf{sk}}^{(\delta)} = \mathsf{sk}^{(\delta)}$, then $\mathcal{A}$ fails in $\mathsf{Exp}_{\mathcal{A},k}^{link}(\lambda)$. 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 $\mathsf{Exp}_{\mathcal{A},k}^{non\text{-}sl}(\lambda)$, where the first k signatures are signed using $\mathsf{sk}^{(\delta^*)}$, and the last one

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

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 $\mathsf{sk}^{(\delta')} := (sk_1^{(\delta')}, \ldots, sk_k^{(\delta')}, sk_{k+1}^{(\delta')})$ be the secret key of $\mathsf{pk}^{(\delta')}$. Similar to the argument above, by rewinding we can extract an extended secret key $\ddot{\mathsf{sk}}^{(\delta')} := (\ddot{sk}_1^{(\delta')}, \ldots, \ddot{sk}_k^{(\delta')}, \ddot{sk}_{k+1}^{(\delta')})$ 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 $\mathcal{H}_0(\mathsf{e}^*) = (epp_1, \ldots, epp_k)$. From the k signatures signed by $\mathsf{sk}^{(\delta^*)} := (sk_1^{(\delta^*)}, \ldots, sk_k^{(\delta^*)}, sk_{k+1}^{(\delta^*)})$, the event-oriented public keys

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

of δ^* are totally exposed. At the same time, the event-oriented public keys $(epk_1^{(\delta')}, \ldots, epk_k^{(\delta')}) = (f_{epp_1}(sk_1^{(\delta')}), \ldots, f_{epp_k}(sk_k^{(\delta')}))$ of δ' are fixed. Namely, the adversary $\mathcal{A}$ has to find a group of coefficients $(a_1, \ldots, a_k)$ 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 $\mathcal{A}$.

□

F Concrete Scheme from the SIS Assumption

Scheme 2 (k-LRS-SIS). Let $H_0 : \{0,1\}^* \mapsto (\mathbb{Z}_q^{n\times m})^k$ and $H : \{0,1\}^* \mapsto \{0,1\}^t$ be two hash functions. Define $\mathsf{SAFE} := \{0, 1, \ldots, 10m-1\}^{km+m}$. 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 $pp := \mathbf{A} = (\mathbf{A}_1 \| \ldots \| \mathbf{A}_{k+1})$.
– Key Generation: For $i \in [k+1]$, $\mathbf{x}_i \xleftarrow{\$} \{0,1\}^m$. Set $\mathbf{x}^\top := (\mathbf{x}_1^\top \| \ldots \| \mathbf{x}_{k+1}^\top)$ and $\mathbf{y} := \sum_{i\in[k+1]} \mathbf{A}_i\mathbf{x}_i$. Return $(sk, pk) := (\mathbf{x}, \mathbf{y})$.
– Sign: Let $PK_R := (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)})$ be the set of public keys in the ring[18]. 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 $a_1, \ldots, a_k \xleftarrow{\$} \mathbb{Z}_q^*$, compute $H_0(\mathsf{e}) = (\mathbf{A}_{\mathsf{e},1}, \ldots, \mathbf{A}_{\mathsf{e},k}) \in (\mathbb{Z}_q^{n\times m})^k$, and $\tilde{\mathbf{y}} := \sum_{i\in[k]} a_i\mathbf{A}_{\mathsf{e},i}\mathbf{x}_i^{(1)}$.
  2. For $\iota = 1, \ldots, t$: (the underlying identification protocol repeats t times)
     (a) $\mathbf{r}_{\iota,1}, \ldots, \mathbf{r}_{\iota,k+1} \xleftarrow{\$} \{0, 1, \ldots, 10m-1\}^m$, and $\mathbf{r}_\iota^\top := (\mathbf{r}_{\iota,1}^\top \| \ldots \| \mathbf{r}_{\iota,k+1}^\top)$.
     (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) $\widetilde{cmt}^{(1)} := (\widetilde{cmt}_1^{(1)}, \ldots, \widetilde{cmt}_t^{(1)})$, and $\widetilde{ecmt}^{(1)} := (\widetilde{ecmt}_1^{(1)}, \ldots, \widetilde{cmt}_t^{(1)})$.
  3. For $j = 2, \ldots, n$:
     (a) $\{0,1\}^t \ni ch^{(j)} := H(\widetilde{cmt}^{(j-1)}, \widetilde{ecmt}^{(j-1)}, a_1, \ldots, a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$.
     (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 $\mathbf{r}_{\iota,1}, \ldots, \mathbf{r}_{\iota,k+1} \xleftarrow{\$} \{0, 1, \ldots, 10m-1\}^m$, and $\mathbf{r}_\iota^\top := (\mathbf{r}_{\iota,1}^\top \| \ldots \| \mathbf{r}_{\iota,k+1}^\top)$.
        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 $\mathbf{z}_{\iota,1}, \ldots, \mathbf{z}_{\iota,k+1} \xleftarrow{\$} \{0, 1, \ldots, 10m-1\}^m$, 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 $\mathbf{z}_\iota^\top = (\mathbf{z}_{\iota,1}^\top \| \ldots \| \mathbf{z}_{\iota,k+1}^\top)$.

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

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

4. $ch^{(1)} := H(\widetilde{cmt}^{(n)}, \widetilde{ecmt}^{(n)}, a_1, \ldots, a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$.
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 $\sigma := \left(a_1, \ldots, a_k, \tilde{\mathbf{y}}, \{\widetilde{cmt}^{(j)}, \widetilde{ecmt}^{(j)}, rsp^{(j)} := \{\mathbf{z}_\iota\}_{\iota\in[t]}\}_{j\in[n]}\right)$

– Verification: Let $PK_R = (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)})$ be the set of public keys. For $j = 1, \ldots, n$, compute $ch^{(j)} := H(\widetilde{cmt}^{(i-1)}, \widetilde{ecmt}^{(i-1)}, a_1, \ldots, a_k, \tilde{\mathbf{y}}, PK_R, \mathsf{e}, \mathsf{msg})$ (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 $\left\|\mathbf{z}_\iota^{(j)}\right\| \leq (10\sqrt{k+1})m^{1.5}$, then $sum^{(j)} := sum^{(j)} + 1$.
3. If $sum^{(j)} \geq 0.55t$ for all $j \in [n]$, then return 1, otherwise, return 0.

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

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

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

Remark 11. 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 12 (On the parameter of the SIS-based construction). The SIS-based construction above supports only a small parameter k (e.g., $k \leq 4$), 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., $Q \leq 10$) to ensure anonymity.

G 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.

G.1 Defnition of Multi-Event LRS

Definition 26 ($(k_1, \ldots, k_M)$-MLRS). *Let $M, k_1, \ldots, k_M \in \mathbb{N}^+$. A $(k_1, \ldots, k_M)$-multi-event linkable ring signature ($(k_1, \ldots, k_M)$-MLRS) scheme consists of the following five algorithms. Namely,*
$(k_1, \ldots, k_M)$-$\mathsf{MLRS} := (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Link})$.

– $pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, \ldots, k_M)$. *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 $k_1, \ldots, k_M$, 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*.*
– $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$*: 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* $\mathsf{e}_1, \ldots, \mathsf{e}_M$ *and a message* msg*, and outputs a signature* σ*.*
– $1/0 \leftarrow \mathsf{Ver}(PK_R, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma)$*: The verification algorithm* Ver *takes as input* R*,* PK_R*,* $\mathsf{e}_1, \ldots, \mathsf{e}_M$*,* msg *and* σ*, and outputs a bit indicating the validity of* σ*.*
We say that a signature σ *(w.r.t. events* $\mathsf{e}_1, \ldots, \mathsf{e}_M$*, message* msg *and ring* R*) is valid (resp., invalid), if* $\mathsf{Ver}(PK_R, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma) = 1$
(resp.,$\mathsf{Ver}(PK_R, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma) = 0$*).*
– $1/0 \leftarrow \mathsf{Link}(\pi, \mathsf{e}_\pi, \sigma_1, \ldots, \sigma_{k_\pi+1})$*: The link algorithm that takes as input an event index* $\pi \in [M]$*, an event label* e_π *and* $(k_\pi + 1)$ *signatures* $\sigma_1, \ldots, \sigma_{k_\pi+1}$*, 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* $M, N, k_1, \ldots, k_M \in \mathbb{N}^+$*. For any* $R \subseteq [N]$ *and* $\delta \in R$*, any events* $(\mathsf{e}_1, \ldots, \mathsf{e}_M)$ *and message* msg*, it holds that*

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

2. *Let* $M, N, k_1, \ldots, k_M \in \mathbb{N}^+$*. 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* $\mathsf{msg}_1, \ldots, \mathsf{msg}_{k_\pi+1}$*, it holds that*

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

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

Remark 13. 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 $(k_1, \ldots, k_M)$-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 27 (Unforgeability of $(k_1, \ldots, k_M)$-MLRS). *Let* MLRS *be a* $(k_1, \ldots, k_M)$*-MLRS scheme. We say that* MLRS *has unforgeability, if,* $\mathsf{Adv}^{unforg}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) := \Pr[\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) = 1] \leq negl(\lambda)$ *holds for all PPT adversary* $\mathcal{A}$*, where the experiment* $\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ *is defined in Fig. 12.*

Experiment $\mathsf{Exp}^{unforg}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$:

$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, \ldots, k_M)$
for each $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
$\mathcal{S} := \mathcal{S}_{corr} := \varnothing$
$(R^*, \mathsf{e}_1^*, \ldots, \mathsf{e}_M^*, \mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$
if $R^* \subseteq [N] \wedge R^* \cap \mathcal{S}_{corr} = \varnothing \wedge (R^*, \mathsf{e}_1^*, \ldots, \mathsf{e}_M^*, \mathsf{msg}^*, \cdot) \notin \mathcal{S}$
  $\wedge\ \mathsf{Ver}(PK_{R^*}, \mathsf{e}_1^*, \ldots, \mathsf{e}_M^*, \mathsf{msg}^*, \sigma^*) = 1$:
  return 1
return 0

Oracle $\textsc{Sign}(R, \delta, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$:

if $R \not\subseteq [N] \ \vee\ \delta \notin R$: **return** $\perp$
$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$
$\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma)\}$
return σ

Oracle $\textsc{Corr}(j)$:

$\mathcal{S}_{corr} := \{j\}$
return $sk^{(j)}$

Fig. 12. The unforgeability experiment of $(k_1, \ldots, k_M)$-MLRS.

Definition 28 ((Unconditional) anonymity of $(k_1, \ldots, k_M)$-MLRS). *Let* MLRS *be a* $(k_1, \ldots, k_M)$*-MLRS scheme. We say that* MLRS *has anonymity, if,* $\mathsf{Adv}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) := |\Pr[\mathsf{Exp}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) = 1] - 1/2| \leq negl(\lambda)$ *holds for all PPT adversary* $\mathcal{A}$*, where* $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ *is defined in Fig. 13.*

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

Experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$:

$b \xleftarrow{\$} \{0,1\}$
$pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, \ldots, k_M)$
$\mathcal{S}_{pair} := \varnothing$
for each $j \in [N]$:
  $(sk^{(j)}, pk^{(i)}) \leftarrow \mathsf{Gen}(pp)$
  $\mathcal{T}[j] := 0$
for each δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$
$b' \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Chall}(b,\cdots)}(\{pk^{(j)}\}_{j\in[N]})$
if $b = b'$: **return** 1

return 0

Oracle $\textsc{Sign}(R, \delta, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$:

if $R \not\subseteq [N] \ \vee\ \delta \notin R$: **return** $\perp$
$\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$
for each $\iota \in [M]$: $++\,\mathcal{L}[\delta, \mathsf{e}_\iota]$
$++\,\mathcal{T}[\delta]$
return σ

Oracle $\textsc{Chall}(b, R, \delta_0, \delta_1, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}_0, \mathsf{msg}_1)$:

if $R \not\subseteq [N] \ \vee\ \delta_0 \notin R \ \vee\ \delta_1 \notin R$: **return** $\perp$
If $\mathcal{T}[\delta_0] \geq \min_\iota(k_\iota) \vee \mathcal{T}[\delta_1] \geq \min_\iota(k_\iota)$: **return** $\perp$
for each $\iota \in [M]$:
  if $\mathcal{L}[\delta_0, \mathsf{e}_\iota] \geq k_\iota \ \vee\ \mathcal{L}[\delta_1, \mathsf{e}_\iota] \geq k_\iota$: **return** $\perp$
for each $\iota \in [M]$: $++\,\mathcal{L}[\delta_0, \mathsf{e}_\iota]$, $++\,\mathcal{L}[\delta_1, \mathsf{e}_\iota]$
$\sigma_0 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_0 \oplus b)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}_0)$
$\sigma_1 \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta_1 \oplus b)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}_1)$
return (σ_0, σ_1)

Fig. 13. The anonymity experiment of $(k_1, \ldots, k_M)$-MLRS, where the hightlighted codes are only shown in the unconditional anonymity experiment.

Definition 29 (Linkability of $(k_1, ., , , .k_M)$-MLRS). *Let* MLRS *be a* $(k_1, \ldots, k_M)$*-MLRS scheme. We say that* MLRS *has unforgeability, if,* $\mathsf{Adv}^{link}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) := \Pr[\mathsf{Exp}^{link}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) = 1] \leq negl(\lambda)$ *holds for all PPT adversary* $\mathcal{A}$*, where the experiment* $\mathsf{Exp}^{link}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ *is defined in Fig. 14.*

Definition 30 (Non-slanderability of $(k_1, ., , , .k_M)$-MLRS). *Let* MLRS *be a* $(k_1, \ldots, k_M)$*-MLRS scheme. We say* MLRS *has unforgeability, if* $\mathsf{Adv}^{non\text{-}sl}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) := \Pr[\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,\ldots,k_M}(\lambda) = 1] \leq negl(\lambda)$ *holds for all PPT adversary* $\mathcal{A}$*, where the experiment* $\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ *is defined in Fig. 15.*

$\underline{\mathsf{Exp}^{link}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)}$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, \ldots, k_M)$
2. **for each** $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
3. **for each** δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$
4. $\mathcal{S} := \mathcal{S}_{corr} := \varnothing$
5. $(\pi^*, \mathsf{e}^*, \{(R_i^*, \mathsf{msg}_i^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \sigma_i^*)\}_{i\in[\ell]}) \leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^j\}_{j\in[N]})$
6. **if** $(\forall i \in [\ell] : (R_i^*, \mathsf{e}_{i,1}, \ldots, \mathsf{e}_{i,M}, \mathsf{msg}_i^*, \cdot) \notin \mathcal{S}$
7. $\quad \wedge\ \mathsf{Ver}(PK_{R_i^*}, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \mathsf{msg}_i^*, \sigma_i^*) = 1\ \wedge\ \mathsf{e}_{i,\pi^*} = \mathsf{e}^*\big)$
8. $\wedge\ (\forall G \subseteq [\ell], |G| = k_{\pi^*} + 1 : \mathsf{Link}(\pi^*, \mathsf{e}^*_{\pi^*}, \{\sigma_i^*\}_{i\in G}) = 0)$
9. $\wedge\ |\mathcal{S}_{corr}| \le \lfloor (\ell - 1)/k_{\pi^*} \rfloor$: **return** 1
10. **return** 0

Oracle $\textsc{Sign}(R, \delta, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$:

11. **if** $R \not\subseteq [N]\ \vee\ \delta \notin R$: **return** $\perp$
12. **for each** $\iota \in [M]$: **if** $\mathcal{L}[\delta, \mathsf{e}_\iota] \ge k_\iota$: **return** $\perp$
13. **for each** $\iota \in [M]$: $++\,\mathcal{L}[\delta, \mathsf{e}_\iota]$
14. $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$
15. $\mathcal{S} := \mathcal{S} \cup \{(R, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma)\}$
16. **return** σ

Oracle $\textsc{Corr}(j)$:

17. $\mathcal{S}_{corr} := \mathcal{S}_{corr} \cup \{j\}$
18. **return** $sk^{(j)}$

Fig. 14. The linkability experiment of $(k_1, \ldots, k_M)$-MLRS.

Experiment$\mathsf{Exp}^{non\text{-}sl}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda, M, k_1, \ldots, k_M)$
2. **for each** $j \in [N]$: $(sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
3. $\mathcal{S} := \mathcal{S}_{corr} := \varnothing$
4. **for each** δ, e: $\mathcal{L}[\delta, \mathsf{e}] := 0$
5. $(\pi^*, \mathsf{e}^*, \delta^*, \{(R_i^*, \mathsf{msg}_i^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \sigma_i^*)\}_{i\in[k_{\pi^*}+1]})$
   $\leftarrow \mathcal{A}^{\textsc{Sign}(\cdots),\textsc{Corr}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$
6. **if** $\{(R_i^*, \delta^*, \{\mathsf{e}_{i,\iota}\}_{\iota\in[M]}, \mathsf{msg}_i^*, \sigma_i^*)\}_{i\in[k_{\pi^*}]} \subseteq \mathcal{S}$
7. $\quad \wedge\ (R^*_{k_{\pi^*}+1}, \delta^*, \{\mathsf{e}_{k_{\pi^*}+1,\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_{k_{\pi^*}+1}, \cdot) \notin \mathcal{S}$
8. $\quad \wedge\ (\forall i \in [k_{\pi^*} + 1] : \mathsf{e}_{i,\pi^*} = \mathsf{e}^*)$
9. $\quad \wedge\ \mathsf{Ver}(PK_{R^*_{k_{\pi^*}+1}}, \{\mathsf{e}_{k_{\pi^*}+1,\iota}\}_{\iota\in[M]}, \mathsf{msg}^*_{k_{\pi^*}+1}, \sigma^*_{k_{\pi^*}+1}) = 1$
10. $\quad \wedge\ \mathsf{Link}(\pi^*, \mathsf{e}^*_{\pi^*}, \{\sigma_i^*\}_{i\in[k_{\pi^*}+1]}) = 1\ \wedge\ \delta^* \notin \mathcal{S}_{corr}$:
11. $\quad$ **return** 1
12. **else** : **return** 0

Oracle $\textsc{Sign}(R, \delta, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$:

13. **if** $R \not\subseteq [N]\ \vee\ \delta \notin R$: **return** $\perp$
14. **for each** $\iota \in [M]$: **if** $\mathcal{L}[\delta, \mathsf{e}_\iota] \ge k_\iota$: **return** $\perp$
15. **for each** $\iota \in [M]$: $++\,\mathcal{L}[\delta, \mathsf{e}_\iota]$
16. $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg})$
17. $\mathcal{S} := \mathcal{S} \cup \{(R, \delta, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}, \sigma)\}$
18. **return** σ

Oracle $\textsc{Corr}(j)$:

19. $\mathcal{S}_{corr} := \{j\}$
20. **return** $sk^{(j)}$

Fig. 15. The non-slanderability experiment of $(k_1, \ldots, k_M)$-MLRS.

G.2 Generic Construction from k-LRS

Let N be the total number of users, M be the total number of event indexes, and $k_1, \ldots, k_M$ 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 $(k_1, \ldots, k_M)$-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 $pp := (pp_1, \ldots, pp_M)$.
- Key Generation: For all $i \in [M]$: $(sk_i, pk_i) \leftarrow k_i\text{-LRS.Gen}(pp_i)$. Return the secret key $\mathsf{sk} := (sk_1, \ldots, sk_M)$ and $\mathsf{pk} := (pk_1, \ldots, pk_M)$.
- Sign: Let $(\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(n)})$ 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 $(\mathsf{e}_1, \ldots, \mathsf{e}_M)$, the signer does as follows.
  1. Let $\overline{\mathsf{msg}} := (R \| \mathsf{msg} \| \mathsf{e}_1 \| \ldots \| \mathsf{e}_M)$.
  2. Parse $\mathsf{sk}^{(1)} = (sk_1, \ldots, sk_M)$. For $i \in [M]$, use sk_i to sign $\overline{\mathsf{msg}}$ under event e_i, i.e., $\sigma_i \leftarrow k_i\text{-LRS.Sign}(PK_R, sk_i, \mathsf{e}_i, \overline{\mathsf{msg}})$.
  3. Return $\sigma := (\sigma_1, \ldots, \sigma_M)$.
- 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 $k_i\text{-LRS.Ver}(PK_{R,i}, \mathsf{e}_i, \overline{\mathsf{msg}}, \sigma_i) = 1$ for all $i \in [M]$, then return 1. Otherwise, return 0.
- Link: Let $\{\sigma_i := (\sigma_{i,1}, \ldots, \sigma_{i,M})\}_{i\in[k_\pi+1]}$ be $(k_\pi + 1)$ signatures whose corresponding π-th event labels are e_π. Return $k_\pi\text{-LRS.Link}(\mathsf{e}_\pi, \sigma_{1,\pi}, \ldots, \sigma_{k_\pi+1,\pi})$.

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

Theorem 13. *If k_1-$\mathsf{LRS}, \ldots, k_M$-$\mathsf{LRS}$ are secure LRS schemes (i.e., they have unforgeability, unconditional (resp., computational) anonymity, linkability, and non-slanderability), then the MLRS scheme $(k_1, \ldots, k_M)$-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 $(R^*, \mathsf{e}_1^*, \ldots, \mathsf{e}_M^*, \mathsf{msg}^*, \sigma^*)$ be $\mathcal{A}$'s final forgery and $\sigma^* := (\sigma_1^*, \ldots, \sigma_M^*)$. Recall $\mathcal{A}$ has never queried the signing oracle with $(R^*, \mathsf{e}_1^*, \ldots, \mathsf{e}_M^*, \mathsf{msg}^*)$, as otherwise $\mathcal{A}$ fails in the unforgeability experiment. Therefore,

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

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 $i = 2, \ldots, M$).

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

Game G_i $(0 \le i \le M)$. In this game, upon receiving $\textsc{Chall}(R, \delta_0, \delta_1, \mathsf{e}_1, \ldots, \mathsf{e}_M, \mathsf{msg}_0, \mathsf{msg}_1)$, the challenger $\mathcal{C}$ first sets $\overline{\mathsf{msg}}_0 := (R \| \mathsf{msg}_0 \| \mathsf{e}_1 \| \ldots \| \mathsf{e}_M)$ and $\overline{\mathsf{msg}}_1 := (R \| \mathsf{msg}_1 \| \mathsf{e}_1 \| \ldots \| \mathsf{e}_M)$, 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 $(\sigma_0, \sigma_1) := ((\sigma_{0,1}, \ldots, \sigma_{0,M}), (\sigma_{1,1}, \ldots, \sigma_{1,M}))$.

It is easy to see that G_0 is just the experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ with $b = 0$, and G_M is just the experiment $\mathsf{Exp}^{anony}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$ 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 $(k_1, \ldots, k_M)$-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 $\mathsf{Exp}^{link}_{\mathcal{A},k_1,\ldots,k_M}(\lambda)$, and $\sigma_i^* := (\sigma_{i,1}^*, \ldots, \sigma_{i,M}^*)$. 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_{π^*}-$\mathsf{LRS.Ver}(PK_{R_i^*,\pi^*}, \mathsf{e}_{i,\pi^*}, \overline{\mathsf{msg}}_i^*, \sigma_{i,\pi^*}^*) = 1$, where

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

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

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 $\sigma_i^* = (\sigma_{i,1}^*, \ldots, \sigma_{i,M}^*)$. 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^*}-LRS.Ver$(PK_{R_{k_{\pi^*}+1}^*,\pi^*}, \mathsf{e}^*, \overline{\mathsf{msg}}_{k_{\pi^*}+1}^*, \sigma_{k_{\pi^*}+1,\pi^*}^*) = 1$, where

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

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

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

H Related Work and Further Discussions

H.1 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] 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 payments [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 and hence it is one-time-used.

Relation with k-Times Traceable Ring Signatures. k-times (full) traceable ring signatures (k-TRS) were introduced by Bultel and Lafourcade [BL16], which allow a signer to sign on behalf of the ring within k-times. 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]. Recently, Xie et al. [XZC+24] extended the method in [BL16] to additionally support revocation.

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 7). 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 H.2.

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.

Relation with Many-Time Linkable Ring Signatures [TNL+25]. Recently in an independent work [TNL+25] Tran et al. introduced many-time linkable ring signatures (T-LRS), which allow a signer to remain anonymous as long as the number of their signatures is smaller than a system-global threshold. Both T-LRS and our k-LRS aim to extend the threshold of signing one can make while preserving anonymity, but they differ in the formalization and technical treatments.

- T-LRS adopts a less general linking approach rather than using an event-oriented notion of linking as in this work. This limitation stems from the design of the linkability tag (equivalent to the pseudoidentity in this work), which is uniquely determined by the secret key and a counter.
- T-LRS cannot achieve the full k-linkability considered in this work. Namely, once a signer signs $k+1$ times, T-LRS can only identify two (of the $k+1$ signatures) by checking the same linkability tag, but not reveal all signatures of this signer. As discussed in the introduction, k-linkability is desirable in scenario like voting where a full accountability of malicious users is required.
- Similar to k-times TRS [BL16] and k-TAA [TFS04], the T-LRS scheme in [TNL+25] is stateful, while our schemes are stateless. As shown in Appendix O, such statelessness is essential to argue the lower bound of linking.

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.

H.2 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 Section 2) 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. $O(n)$).

H.3 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.

H.4 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.

I Anonymous Bulletin Board

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

Definition 31 (Anonymous Bulletin Boards). *Let $P^{(1)}, \ldots, P^{(n)}$ be n parties. An anonymous bulletin board ABB among $P^{(1)}, \ldots, P^{(n)}$ is a functionality such that,*

- *in the writing phase, ABB takes a set of messages $X^{(j)} := \{x_1^{(j)}, \ldots, x_{N_j}^{(j)}\}$ as inputs from every party $P^{(j)}$ ($j \in [n]$), and*
- *in the reading phase, ABB outputs a set of messages $\Gamma := \{y_1, y_2, \ldots, y_{\sum_j N_j}\}$, 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 $j \in [n]$.

J BCV (Borda Count Voting) from MLRS

In this section we consider BCV for the Borda count (a.k.a. Borda voting), where there are q candidates $C_1 \ldots, C_q$, and each voter is eligible to cast a sort of those q candidates as their votes. More formally, $\mathcal{E} \subseteq [q]^{*q}$ (all permutations of $[q]$). 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 5, 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 BCV by $(1,1)$-MLRS (cf. Appendix G 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 GMV 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 GMV, here the above BCV 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.

K Relaxed GMV and Simpler Constructions from k-LRS and MLRS

In this section we consider two relaxed variants of GMV, 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 GMV from MLRS. We can also design a GMV system from (w,t)-MLRS, where the i-th vote of a voter u on candidate C_{c_i} is in the form of

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

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.

L Deferred Proofs

L.1 Proofs of Theorems for Voting Modes

Proof (of Theorem 14 (Security of GSV 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.[19] 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.[20] 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. □

Proof (of Theorem 4 (Security of GMV)). 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.

[19] 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.

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

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. □

L.2 Proof of the Negative Result on Privacy

Proof (of Theorem 5). 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[21] 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)

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

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. □

[21] 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.

M Detailed Approaches of Reducing the Complexity of Linking

M.0.1 Variant 1: Use deterministic pseudoidentity additionally. Formally, we consider the following relaxed variant of GSV.

Preparing phase. u generates a key pair $(\hat{sk}_u, \hat{sk}_u)$ of a k-LRS scheme, and k key pairs $(sk'_{u,1}, sk'_{u,1}), \ldots, (sk'_{u,k}, pk'_{u,k})$ of an LRS scheme, and registers $pk_u = (\hat{pk}_u, pk'_{u,1}, \ldots, pk'_{u,k})$ 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. 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 $(\sigma_1, \ldots, \sigma_s)$ 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 L.1.

Theorem 14. *If the underlying k-LRS scheme and LRS scheme in GSV has unforgeability, anonymity, linkability, and non-slanderability, then the stateful voting protocol for GSV has correctness and privacy.*

M.0.2 Variant 2: Partition the anonymous ring. Formally we 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 $\mathcal{V} := (v_1, \ldots, v_n)$ be the universe of all voters, sorted in alphabetical order. Given that $m \mid n$ (m divides n), partition $\mathcal{V}$ into m disjoint subsets $\mathcal{V}_1, \ldots, \mathcal{V}_m$, where for each $j \in [m]$, $\mathcal{V}_j = (v_{(j-1)n/m+1}, \ldots, v_{jn/m})$.

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 the predetermined event label on behalf of $\mathcal{V}_j$, where $\mathcal{V}_j$ denotes the subset of voters to which u belongs.

Let $(\sigma_1, \ldots, \sigma_s)$ 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, 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 15. *If $m \mid n$ and the underlying k-LRS scheme in GSV has unforgeability, (unconditional) anonymity, linkability, and non-slanderability, then the voting protocol for GSV has correctness and (unconditionally) privacy among the sub-ring of size n/m.*

N On The Hardness of Efficient Link (Overvote Identification)

When using k-LRS in a voting scheme, $\binom{\ell}{k+1}$ invocations of Link are required. In this section, we argue that further reducing this complexity appears infeasible by a reduction to an NP-complete coding problem.

Recall that in both our two schemes, a (valid) signature is of the form

$$\sigma = (a_1, \ldots, a_k, \tilde{y}, \ldots),$$

where $a_1, \ldots, a_k \in \mathbb{Z}_q$ are random coefficients sampled in signing, and $\tilde{y}$ is the so-called pseudorandom identity that is computed as $\tilde{y} = \sum_{i=1}^{k} a_i \cdot x_i$ with x_i being (an bijective image of) the i-th secret key of the signer.

Now consider $k+1$ signatures $\sigma_1, \ldots, \sigma_{k+1}$ such that

$$\begin{aligned}
\sigma_1 &:= (\ a_{1,1}, \quad a_{1,2}, \ \cdots \ a_{1,k}, \quad \tilde{y}_1, \ \cdots), \\
\sigma_2 &:= (\ a_{2,1}, \quad a_{2,2}, \ \cdots \ a_{2,k}, \quad \tilde{y}_2, \ \cdots), \\
&\cdots \\
\sigma_{k+1} &:= (\ a_{k+1,1}, a_{k+1,2}, \cdots a_{k+1,k}, \tilde{y}_{k+1}, \cdots).
\end{aligned}$$

- If all signatures are created by the same signer, then there is a solution $(x_1, \ldots, x_k)$ satisfying the above $k+1$ equations.
- If not, then with high probability that such a solution does not exist, due to the randomness of $a_{i,j}$ $(i \in [k+1],\ j \in [k])$ and independence of secret keys.

In the voting applications, assume that there are n signatures. During the tally, one needs to check if overvote happens, i.e., to check if there are $k+1$ signatures linked, and output the subgroup if so. We can abstract this problem as follows.

Definition 32 (Problem 1). *Let $\mathbb{F}$ be a field and integers $n > k$. Given n groups $(\mathbf{a}_i, z_i) \in \mathbb{F}^n \times \mathbb{F}$, return a subgroup $S \subseteq [n]$ such that $|S| = k+1$, and there exists $\mathbf{b} \in \mathbb{F}^k$ such that $\langle \mathbf{a}_i, \mathbf{b} \rangle = z_i$ holds for all $i \in S$. If such S does not exist, return 0.*

A trivial solution is to try every subgroup of size $k+1$ and check the linear dependence, which leads to a computation cost $O\left(\binom{n}{k}(k^3 + nk)\right)$.

Let $\mathsf{Hw}(\mathbf{c})$ denote the Hamming weight (the number of non-zero coordinates of a vector) of $\mathbf{c} \in \mathbb{F}^n$. Now consider the following coding problem.

Definition 33 (Problem 2). *Let $G \in \mathbb{F}^{(k+1)\times n}$ be a generator matrix for a linear code*

$$\mathcal{C} := \{\mathbf{y}^\top G \mid \mathbf{y} \in \mathbb{F}^{k+1}\},$$

and let $d < n$ be an integer. Return a non-zero codeword $\mathbf{c}$ such that $\mathsf{Hw}(\mathbf{c}) \leq d$, or return 0 if such codeword does not exist.

Problem 2 is known be NP-complete in [Var02] (for $\mathbb{F} = \mathbb{F}_2$) and [GV05] (for $\mathbb{F} = \mathbb{F}_q$).

Theorem 16. *There is a polynomial reduction that reduces an instance of Problem 2 with parameters $n, k, d = (n - k - 1)$ to an instance of Problem 1 with parameters n, k. (Therefore Problem 1 is at least as hard as Problem 2.)*

Proof. Let $G \in \mathbb{F}^{(k+1)\times n}$ be the generator matrix for an instance of Problem 2. Then we define an instance of Problem 1 as follows. Set $[\mathbf{A}|\mathbf{z}] := G^\top \in [\mathbb{F}^{n\times k}|\mathbb{F}^n]$, and parse $\mathbf{A}^\top = [\mathbf{a}_1^\top|\ldots|\mathbf{a}_n^\top]$, $\mathbf{z}^\top = (z_1, \ldots, z_n)$.

Then we show that a solution to Problem 1 implies a solution to Problem 2.

- Assume $S \subseteq [n]$ is a solution. By Gaussian elimination it is easy to compute $\mathbf{b} \in \mathbb{F}^k$ such that $\mathbf{A}_S \cdot \mathbf{b} = \mathbf{z}_S$. Here $\mathbf{A}_S \in \mathbb{F}^{(k+1)\times k}$ (resp. $\mathbf{z}_S \in \mathbb{F}^{k+1}$) denotes the submatrix of $\mathbf{A}$ (resp. $\mathbf{z}$) with rows $i \in S$. Therefore, $\mathbf{c}^\top := (\mathbf{b}^\top, -1) \cdot G \in \mathbb{F}^{1\times n}$ has 0-coordinate on all positions $i \in S$, which means that $\mathsf{Hw}(\mathbf{c}) \leq n-(k+1) = d$.
- Assume there is no solution. In this case, $[\mathbf{A}_S|\mathbf{z}_S] \in \mathbb{F}^{(k+1)\times(k+1)}$ has full rank for all $S \subseteq n$ and $|S| = k + 1$, and it is impossible for a codeword $\mathbf{c}^\top = \mathbf{y}^\top \cdot G = \mathbf{y}^\top \cdot [\mathbf{A}|\mathbf{z}]^\top$ to have at least $k + 1$ 0-coordinates. Consequently, $\mathsf{Hw}(\mathbf{c}) > n - (k + 1) = d$ holds for all codewords $\mathbf{c}$. □

O Lower Bound of Link

In contrast to stateful algorithms, stateless algorithms, as the name suggests, do not maintain internal state; each output depends solely on the input arguments and the randomness used during execution. This design provides stronger privacy guarantees, as it inherently reveals no information about the state of other signers and does not leak any information about the signer's own past behavior.

To capture this stronger notion of privacy, we introduce a strengthened anonymity definition, which we refer to as subset (unconditional) anonymity (Def. 15). The corresponding anonymity game (Fig. 5) closely follows the original anonymity game, with the sole modification that the signing oracle takes an additional input bit view, indicating whether the generated signature is revealed to the adversary. We require only that the challenged signatures provided to the adversary do not oversign, while still allowing the adversary to issue signing queries that may result in oversigning elsewhere in the system.

Once the adversary obtains more than k signatures from the same signer, it can invoke the linking algorithm to test linkability among those signatures. Such information would trivially allow the adversary to distinguish between different challenge histories. Therefore, we restrict the adversary from obtaining any $(k+1)$-subset of signatures, hereafter referred to as an *oversigned set*, from signers involved in the challenge oracle, while imposing no such restriction on signers outside the challenge set.

We next discuss the scope of anonymity captured by our new definition (Def. 15).

Specifically, we show that subset anonymity implies the original notion of anonymity. (Lma. 4) In addition, we present a separating example demonstrating that the converse does not hold (Lma. 5): there exists a stateful algorithm that satisfies the original anonymity definition but fails to achieve subset anonymity.

Lemma 4. *The subset (k-unconditional) anonymity (Def. 15) implies the (k-unconditional) anonymity (Def. 15).*

Proof. By setting $\mathsf{view} := \mathsf{True}$ for all signing queries, the subset anonymity experiment reduces to the original (k-unconditional) anonymity experiment. Consequently, any adversary against (k-unconditional) anonymity can be directly transformed into an adversary against subset (k-unconditional) anonymity, which proves the claim.

Lemma 5. *Assume there exists a k-LRS scheme that satisfies subset (unconditional) anonymity (Def. 15), we can construct a new k-LRS scheme such that it satisfies (unconditional) anonymity (Def. 3) but does not satisfy subset (unconditional) anonymity.*

Proof. Given the k-LRS scheme Π that satisfies subset (unconditional) anonymity (Def. 15), we construct the following k-LRS scheme Π':

- $pp \leftarrow \mathsf{Setup}'(1^\lambda, k) := \mathsf{Setup}(1^\lambda, k)$. Remains the same.
- $(sk, pk) \leftarrow \mathsf{Gen}'(pp) := \mathsf{Gen}(pp)$. Remains the same.
- $\sigma' \leftarrow \mathsf{Sign}'(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$: The algorithm has a counter vector cnt such that each time calling the signing algorithm will increment $\mathsf{cnt}[\delta]$ by 1.
  - Initially, the counter vector is set to 0 for all entries.
  - For $\mathsf{cnt}[\delta] < k+1$, $\sigma' := (\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg}), \emptyset)$. It outputs the signature as the original scheme Π did, with an empty set.
  - Once $\mathsf{cnt}[\delta] \geqslant k+1$, $\sigma' := (\sigma, \{\sigma_{\delta,1}, \sigma_{\delta,2}, \ldots, \sigma_{\delta,\mathsf{cnt}[\delta]}\})$, where it outputs the signature as the original scheme Π did $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$, but also provided with the set of all signatures that generated from the same signer.
- $\mathsf{Ver}'(PK_R, \mathsf{e}, \mathsf{msg}, \sigma')$: The verification algorithm Ver' takes as input σ', decompose it into $\sigma' := (\sigma, \mathcal{S})$ and outputs the bit by $0/1 \leftarrow \mathsf{Ver}(PK_R, \mathsf{e}, \mathsf{msg}, \sigma)$
- $\mathsf{Link}'(\mathsf{e}, \sigma'_1, \ldots, \sigma'_{k+1})$: decompose all signatures $\sigma'_i := (\sigma_i, \mathcal{S}_i)$ for all $i \in [k+1]$, and then runs the original algorithm $0/1 \leftarrow \mathsf{Link}(\mathsf{e}, \sigma_1, \ldots, \sigma_{k+1})$.

It is trivial to check that this new scheme Π' is indeed a k-LRS scheme. However, this scheme does not satisfy the subset anonymity property. Specifically, an adversary can call the challenge oracle $k+2$ times by fixing one signer to be δ^* and pairing them with different signers each time. The adversary can select to observe only the last k signatures. According to the definition above, the last signature observed reveals the complete signing history of δ^*, allowing the adversary to identify which signature was generated by δ^* and thus win the experiment with probability 1.

Next, we argue that the modified scheme Π' satisfies (k-unconditional) anonymity. By construction, for each signer involved in the challenge oracle, it is guaranteed that the signer produces at most k signatures. For signers that appear only in the signing oracle, any information revealed through the linking of their signing history does not provide the adversary with an advantage, as this information is independent of the secret challenge bit b. □

Lemma 6. *If a stateless k-LRS scheme satisfies (k-unconditional) anonymity, it also satisfies the subset (k-unconditional) anonymity.*

Proof. This argument relies critically on the fact that stateless algorithms do not maintain internal counters or state, so that each execution depends solely on the input arguments and the randomness used. Consequently, for a stateless algorithm, it is sound to reorder the sequence of oracle calls: as long as the input arguments remain unchanged, the output distribution is identical.

Therefore, we may rearrange the oracle interactions by moving all queries with $\mathsf{view} = \mathsf{True}$ to the front of the execution. This transformation yields an experiment that is equivalent to the original anonymity experiment, in which the adversary can obtain at most a negligible advantage in winning. □

Despite the advantages provided by stateless algorithms, which most notably their stronger anonymity guarantees, the stateless property also introduces an undesirable drawback. In particular, stateless algorithms incur significant complexity in the linking procedure: identifying a single $(k+1)$-oversigned set among n signatures may require time on the order of $\mathcal{O}(n^{k+1})$ in the worst case.

We first introduce the *linkability-finding game* (Def. 34). In this game, an adversary $\mathcal{B}$ is given access to a signing oracle SIGN, which it may query to generate a collection of signatures for later use.

We then define a new class of adversaries, referred to as *myopic linkers* (Def.35), whose goal is to identify linkages among the signatures generated by adversary $\mathcal{B}$. The myopic linkers win the game if it successfully output a set of indices corresponding to signatures that are linked.

Definition 34 (Linkability Finding Game). *Let k-LRS be a k-LRS scheme. We define the experiment $\mathsf{Exp}^{find-link}_{\mathcal{A},k}(\lambda)$ in Fig. 16. The advantage of the myopic linkers $\mathcal{A}$ are defined as $\mathsf{Adv}^{find-link}_{\mathcal{A},k}(\lambda) := \Pr[\mathsf{Exp}^{find-link}_{\mathcal{A},k}(\lambda) = 1]$.*

We call the myopic linkers that have access to linking oracle LINK *that only returns the indicating bit the* ideal-world *myopic linkers, and the ones with additional information of signatures the* real-world *myopic linkers.*

We also restrict that each time the myopic linkers are only allowed to call the linking oracle LINK *once for each linker.* π *is a random permutation of size* $\mathcal{I}_{\mathsf{full}}$. idx *never repeats.*

Experiment $\mathsf{Exp}^{find-link}_{\mathcal{A},k}(\lambda)$:

1. $\mathcal{I}_{\mathsf{full}} := \emptyset$
2. $pp \leftarrow \mathsf{Setup}(1^\lambda, k)$
3. **for each** $j \in [N]$:
4. $\quad (sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
5. $\quad \mathcal{I}_{\mathsf{user}}[j] := \emptyset$
6. **for each** idx: $\mathcal{H}[\mathsf{idx}] := \perp$
7. $\mathcal{B}^{\text{SIGN}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$
8. $\mathcal{S}_{link} \leftarrow \boldsymbol{\mathcal{A}}^{\text{LINK}(\cdot)}(\{pk^{(j)}\}_{j\in[N]}, \pi(\mathcal{I}_{\mathsf{full}}))$ // π a random permutation over $\mathcal{I}_{\mathsf{full}}$
9. **for each** $j \in [N]$:
10. $\quad$ **if** $\mathcal{S}_{link} \subseteq \mathcal{I}_{\mathsf{user}}[j]$: **return** 1
11. **else** : **return** 0

Oracle SIGN(δ):

12. **if** $R \nsubseteq [N] \ \vee \ \delta \notin R$: **return** $\perp$
13. $\mathsf{msg} \xleftarrow{\$} \mathcal{M}$; $\mathsf{idx} \xleftarrow{\$} \{0,1\}^\lambda$
14. $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
15. $\mathcal{I}_{\mathsf{user}}[\delta] \leftarrow \mathcal{I}_{\mathsf{user}}[\delta] \cup \mathsf{idx}$
16. $\mathcal{I}_{\mathsf{full}} \leftarrow \mathcal{I}_{\mathsf{full}} \cup \mathsf{idx}$
17. $\mathcal{H}[\mathsf{idx}] \leftarrow (\sigma, \mathsf{msg}, \delta)$
18. **return** $\perp$

Oracle LINK($\mathcal{S}_{idc}$):

19. $\mathsf{flag} := 0$; $\mathcal{H}_{sig} := \emptyset$
20. **for each** $\mathcal{S}_{k+1} \subseteq \mathcal{S}_{idc} \wedge |\mathcal{S}_{k+1}| = k+1$:
21. $\quad$ **for each** $j \in [N]$:
22. $\quad\quad$ **if** $\mathcal{S}_{k+1} \subseteq \mathcal{I}_{\mathsf{user}}[j]$: $\mathsf{flag} \leftarrow 1$
23. **for each** $\mathsf{idx} \in \mathcal{S}_{idc}$:
24. $\quad (\sigma, \mathsf{msg}, \mathsf{e}) \leftarrow \mathcal{H}[\mathsf{idx}]$
25. $\quad \mathcal{H}_{sig} \leftarrow \mathcal{H}_{sig} \cup \sigma$
26. **return** $(\mathsf{flag}, \mathcal{H}_{sig})$

Fig. 16. The linkability finding experiment of k-LRS, where the highlighted codes are only shown for the real world adversaries.

Definition 35 (Myopic Linkers). *We define myopic linkers* $\boldsymbol{\mathcal{A}}_k$ *as a series of* $n(k)$ *p.p.t. adversaries* $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{n(k)}$*, where* $n(k)$ *is a function in parameter* k*. The myopic linkers operate according to the following syntax:*

- *The initial linker* $\mathcal{A}_0$ *is given the set of public keys* $\{pk^{(j)}\}_{j\in[N]}$ *and the full index space* $\mathcal{I}_{\mathsf{full}}$*. It outputs an initial subset of indices where* $\mathcal{S}_{\mathsf{idx},1} \subseteq \mathcal{I}_{\mathsf{full}}$.

$$\mathcal{S}_{\mathsf{idx},1} \leftarrow \mathcal{A}_0(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}). \tag{8}$$

- *For each* $i \in [n(k)]$*, the linker* $\mathcal{A}_i$ *is given the set of public keys* $\{pk^{(j)}\}_{j\in[N]}$*, the full index space* $\mathcal{I}_{\mathsf{full}}$*, the set of previously queried indices* $\mathcal{S}_{past} := \bigcup_{j\leq i} \mathcal{S}_{\mathsf{idx},j}$*, the set of last queried signatures* $\mathcal{H}_{last}$ *where* $|\mathcal{H}_{last}| \leqslant k+1$*, and a bit string* $L \in \{0,1\}^i$ *representing the outcomes of the previous linking queries. The linker then outputs a new subset of indices where* $\mathcal{S}_{\mathsf{idx},i+1} \subseteq \mathcal{I}_{\mathsf{full}}$.

$$\mathcal{S}_{\mathsf{idx},i+1} \leftarrow \mathcal{A}_i(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}, \mathcal{S}_{past}, \mathcal{H}_{last}, L) \tag{9}$$

We note that the adversary is allowed to query of indices of any size, the oracle LINK will return whether there exists a $k+1$ linked subset of signatures.

In the most straightforward approach, the adversary may exhaustively testing all $(k+1)$-subsets of signatures and querying the Link algorithm on each.

The central question, therefore, is whether the adversary can exploit structural information contained in the signatures to accelerate the process of determining and identifying an oversigned set. Since the adversary can invoke the Link algorithm on $(k+1)$ signatures at a time, it is natural to observe that, at each query,

the adversary gains *only* information pertaining to that specific $(k+1)$-subset, and no additional global information about the remaining signatures.

To capture this limitation more precisely, we introduce the new class of adversaries, which we refer to as *myopic linkers*. These adversaries are characterized by their limited view of the system state; specifically, they are restricted to accessing only a small amount of recent information, reflecting a constrained observational scope.

We also note that, compared to the previous signing oracle, we restrict the adversary's flexibility to choosing only the signer. This restriction is necessary because the adversary $\mathcal{B}$ could otherwise encode auxiliary information into the signing parameters and thereby signal the myopic linkers. We illustrate several such attacks below:

- If signer i is instructed to sign message i for all $i \in [N]$, then the signer's identity can be trivially recovered from the message itself.
- If signer i always uses the ring $[N] \setminus \{i+1\}$ (where we identify $N+1$ with 1), then the signer's identity can be embedded in the ring description.
- If signer i always uses an event tag signer_i for all $i \in [N]$, then the signer's identity can be encoded in the event information.
- If the adversary records the correspondence between indices and signers, then the linkers can directly identify the linked signature set.

To prevent such trivial signaling and ensure that the game remains nontrivial, we remove these input freedoms. Specifically, we fix the event tag and the ring membership for all signing queries, and require that messages be generated independently from the same distribution, so that the adversary cannot infer which signer produced which signature. Moreover, all signatures are relabeled so that the indices themselves carry no identifying information. Under these restrictions, the only information that the myopic linkers $\boldsymbol{\mathcal{A}}$ may obtain from adversary $\mathcal{B}$ is the number of signatures generated by each signer.

Lemma 7. *The ideal-world linkers could only achieve maximum advantage* $\mathsf{Adv}^{find-link}_{\boldsymbol{\mathcal{A}},k}(\lambda)$ *with exponential length of linkers provided. Provided that the linkers are only allowed to query sets of size no more than* $k+1$.

Proof. We prove the claim by contradiction. Suppose there exists a series of linkers that can find an oversigned set exists using only $q' < \binom{\ell}{k+1} - 1$ queries. Since $q' < \binom{\ell}{k+1} - 1$, there must exist a subset of indices $\mathcal{S}^* \subseteq \mathcal{I}_{\mathsf{full}}$ with $|\mathcal{S}^*| = k+1$ that is never queried by the adversary.

Consider an execution in which the oracle responds 0, which means *unlinked*, to all of the linkers' queries. We now define two indistinguishable instances:

- In the first instance, the oracle also responds 0, *unlinked* on the unqueried set $\mathcal{S}^*$. Indicating the whole instance have no oversigned at all.
- In the second instance, the oracle responds 1, *linked* on $\mathcal{S}^*$ and 0, *unlinked* on all other $(k+1)$-subsets. Indicating the whole instance exists an oversigned set.

From the linkers' perspective, the two instances are identical, as they receive the same responses to all of their queries. However, the correct outputs in the two instances differ: in the second instance, an oversigned set exists, while in the first, it does not. Therefore, the adversary cannot correctly determine the existence of an oversigned set in both cases, contradicting our assumption. Even though the adversary may guess one of the unqueried sets as the oversigned sets, it does not have the information to guess it correctly with probability 1.

Hence, to achieve the maximum advantage $\mathsf{Adv}^{find-link}_{\boldsymbol{\mathcal{A}},k}(\lambda)$, the ideal-world linkers must query all possible $(k+1)$-subsets of signatures, yielding

$$q = \binom{\ell}{k+1} - 1 \approx \mathcal{O}(n^{k+1})$$

which completes the proof. □

Lemma 8. *For any k-LRS scheme that satisfies subset anonymity (Def. 15), the advantage of real-world linkers are at most a negligible function of λ larger than the advantage of ideal-world linkers that queried the same sets with polynomial running time. Provided that the linkers are only allowed to query sets of size no more than $k+1$.*

Proof. We analyze the two cases separately.

Case Never Hitting the Set during Queries. In this case, all calls to the linking oracle return the bit 0, indicating that no linked signatures exist. We replace the original oracles with the modified oracles shown in Fig. 17, which record only the queried index sets and, when signatures are requested, return freshly generated signatures produced by a fixed set of the first $(k+1)$ signers.

Clearly, ideal-world linkers observe identical behavior in this setting. Moreover, real-world linkers receive signatures that are generated by a fixed set of signers on identical messages and labels. Since this information is independent of the parameters used by the linkers, it does not provide any additional advantage. Therefore, when the sequences of queries to the linking oracle are identical, both ideal-world and real-world linkers achieve the same advantage.

It remains to show that providing real-world linkers with genuine non-linked signatures yields at most a negligible advantage over providing them with the randomly generated fixed signatures returned by the oracle in Fig. 17.

Assume, for contradiction, that the difference in advantage is non-negligible. Then there exists a polynomial $p(\lambda)$ and an index i such that linker $\mathcal{A}_i$ is the first adversary in the sequence able to distinguish the two settings with probability at least $p(\lambda)$. We construct a hybrid experiment in which $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{i-1}$ interact with the modified oracle in Fig. 17, while $\mathcal{A}_i$ interacts with the original oracle in Fig. 16.

Let the input to $\mathcal{A}_i$ be

$$\left(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}, \mathcal{S}_{past}, \mathcal{H}_{last}, L\right).$$

The output and inputs to $\mathcal{A}_{i-1}$ is:

$$\mathcal{S}_{\mathsf{idx},i} \leftarrow \left(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}, \mathcal{S}'_{past}, \mathcal{H}'_{last}, L'\right).$$

We have that

$$\mathcal{S}_{past} = \mathcal{S}'_{past} \cup \mathcal{S}_{\mathsf{idx},i}.$$

We now construct an adversary $\mathcal{C}$ that breaks subset anonymity as follows:

- Adversary $\mathcal{C}$ queries the signing oracle $(i-1)(k+1)$ times using signers indexed by $\mathcal{S}'_{past}$ and the corresponding messages, with $\mathsf{view} = \mathsf{False}$ for all queries. We know the identity of signers since $\mathcal{C}$ plays the role of the challenger of our linkage finding game, and $\mathcal{H}$ record all information we need.
- Adversary $\mathcal{C}$ invokes the challenge oracle $k+1$ times with two signing histories: one corresponding to the real signatures indexed by $\mathcal{S}_{\mathsf{idx},i}$, which is the output from the previous linker $\mathcal{A}_{i-1}$, and the other generated using signers in $[k+1]$ on message msg sampled from the same distribution.
- Upon receiving the $k+1$ challenge signatures, adversary $\mathcal{C}$ partitions them into two sets: $\mathcal{H}_0$, consisting of the first signature from each challenge response, and $\mathcal{H}_1$, consisting of the last signature from each response.
- Adversary $\mathcal{C}$ runs

$$\mathcal{A}_i\left(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}, \mathcal{S}_{past}, \mathcal{H}_0, L\right) \quad \text{and} \quad \mathcal{A}_i\left(\left\{pk^{(j)}\right\}_{j\in[N]}, \mathcal{I}_{\mathsf{full}}, \mathcal{S}_{past}, \mathcal{H}_1, L\right),$$

  and submits the resulting index sets to the linking oracle. If exactly one query returns 1, $\mathcal{C}$ outputs the corresponding guess for the hidden bit b; otherwise, it outputs a random guess.

By assumption, this strategy gives $\mathcal{C}$ a non-negligible advantage in distinguishing the challenge bit b, contradicting the subset anonymity property.

Case Hitting the Set within a Query. If one of the linker queries intersects the linked signature set, then both the ideal-world and real-world linkers will necessarily identify and output the correct set. Therefore, it suffices to consider the execution up to the point at which the linkers have not yet queried any linked $(k+1)$-subset.

Prior to this event, all responses of the linking oracle are identical to those in the case of never hitting the set. Consequently, the difference in advantage between ideal-world and real-world linkers before encountering a linked set is negligible. This completes the analysis for this case.

Summary Therefore, the advantage of real-world linkers interacting with genuine non-linked signatures differs from that of ideal-world linkers interacting with the modified oracle by at most a negligible function in the security parameter λ. This completes the proof. □

Experiment $\mathsf{Exp}^{find-link}_{\boldsymbol{\mathcal{A}},k}(\lambda)$:

1. $\mathcal{I}_{\mathsf{full}} := \emptyset$
2. $pp \leftarrow \mathsf{Setup}(1^\lambda, k)$
3. **for each** $j \in [N]$:
4. $\quad (sk^{(j)}, pk^{(j)}) \leftarrow \mathsf{Gen}(pp)$
5. $\quad \mathcal{I}_{\mathsf{user}}[j] := \emptyset$
6. **for each** idx: $\mathcal{H}[\mathsf{idx}] := \perp$
7. $\mathcal{B}^{\textsc{Sign}(\cdot)}(\{pk^{(j)}\}_{j\in[N]})$
8. $\mathcal{S}_{link} \leftarrow \boldsymbol{\mathcal{A}}^{\textsc{Link}(\cdot)}(\{pk^{(j)}\}_{j\in[N]}, \pi(\mathcal{I}_{\mathsf{full}}))$
   // π a random permutation over $\mathcal{I}_{\mathsf{full}}$
9. **for each** $j \in [N]$:
10. $\quad$ **if** $\mathcal{S}_{link} \subseteq \mathcal{I}_{\mathsf{user}}[j]$: **return** 1
11. **else** : **return** 0

Oracle $\textsc{Sign}(\delta)$:

12. **if** $R \not\subseteq [N] \ \vee \ \delta \notin R$: **return** $\perp$
13. $\mathsf{msg} \xleftarrow{\$} \mathcal{M}$; $\mathsf{idx} \xleftarrow{\$} \{0,1\}^\lambda$
14. $\sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(\delta)}, \mathsf{e}, \mathsf{msg})$
15. $\mathcal{I}_{\mathsf{user}}[\delta] \leftarrow \mathcal{I}_{\mathsf{user}}[\delta] \cup \mathsf{idx}$
16. $\mathcal{I}_{\mathsf{full}} \leftarrow \mathcal{I}_{\mathsf{full}} \cup \mathsf{idx}$
17. $\mathcal{H}[\mathsf{idx}] \leftarrow (\sigma, \mathsf{msg}, \delta)$
18. **return** $\perp$

Oracle $\textsc{Link}(\mathcal{S}_{idc})$:

19. $\mathsf{flag} := 0$; $\mathcal{H}_{sig} := \emptyset$
20. **for each** $\mathcal{S}_{k+1} \subseteq \mathcal{S}_{idc} \wedge |\mathcal{S}_{k+1}| = k+1$:
21. $\quad$ **for each** $j \in [N]$:
22. $\quad\quad$ **if** $\mathcal{S}_{k+1} \subseteq \mathcal{I}_{\mathsf{user}}[j]$: $\mathsf{flag} \leftarrow 1$
23. $\quad$ **for each** $i \in [k+1]$:
24. $\quad\quad \sigma \leftarrow \mathsf{Sign}(PK_R, sk^{(i)}, \mathsf{e}, \mathsf{msg} \xleftarrow{\$} \mathcal{M})$
25. $\quad\quad \mathcal{H}_{sig} \leftarrow \mathcal{H}_{sig} \cup \sigma$
26. **return** $(\mathsf{flag}, \mathcal{H}_{sig})$

Fig. 17. The linkability finding experiment of k-LRS, where the highlighted codes are only shown for the real world adversaries.

By combining the above lemmas, we conclude that no polynomial-time algorithm can determine whether a linkage exists with non-negligible advantage.

We also note that, since we do not restrict the query size in the linkability finding game (Def. 34), allowing the adversary to issue queries of size larger than $k+1$ and receive a single bit indicating whether there exists a $(k+1)$-subset of linked signatures immediately leads to a trivial attack. In particular, even an ideal-world linker could employ the following linear-time search strategy:

- Query the full set of signatures. If the oracle returns 0, then no linkage exists and the algorithm aborts.
- While the link oracle Link returns 1, remove signatures one by one. An index is retained if removing it causes the oracle output to flip to 0, since this implies that the removed signature was part of the linked subset and its removal eliminates all linkages.

By iteratively applying this procedure, the adversary can determine, one index at a time, whether a given signature belongs to a linked set. This yields a linear-scan algorithm that completely recovers the linkage structure, rendering the game trivial to win.