It wasn't me!

Repudiability and Claimability of Ring Signatures

Sunoo Park[1] and Adam Sealfon[2]

[1] MIT and Harvard
[2] MIT

Abstract. Ring signatures, introduced by [RST01], are a variant of digital signatures which certify that *one among a particular set* of parties has endorsed a message while hiding *which* party in the set was the signer. Ring signatures are designed to allow *anyone* to attach anyone else's name to a signature, as long as the signer's own name is also attached.

But what guarantee do ring signatures provide if a purported signatory wishes to denounce a signed message — or alternatively, if a signatory wishes to later come forward and claim ownership of a signature? Prior security definitions for ring signatures do not give a conclusive answer to this question: under most existing definitions, the guarantees could go either way. That is, it is consistent with some standard definitions that a non-signer might be able to *repudiate* a signature that he did not produce, or that this might be impossible. Similarly, a signer might be able to later convincingly claim that a signature he produced is indeed his own, or not. Any of these guarantees might be desirable. For instance, a whistleblower might have reason to want to later claim an anonymously released signature, or a person falsely implicated in a crime associated with a ring signature might wish to denounce the signature that is framing them and damaging their reputation. In other circumstances, it might be desirable that even under duress, a member of a ring cannot produce proof that he did or did not sign a particular signature. In any case, a *guarantee* one way or the other seems highly desirable.

In this work, we formalize definitions and give constructions of the new notions of *repudiable*, *unrepudiable*, *claimable*, and *unclaimable* ring signatures. Our repudiable construction is based on VRFs, which are implied by several number-theoretic assumptions (including strong RSA or bilinear maps); our claimable construction is a black-box transformation from any standard ring signature scheme to a claimable one; and our unclaimable construction is derived from the lattice-based ring signatures of [BK10], which rely on hardness of SIS. Our repudiable construction also provides a new construction of standard ring signatures.

1 Introduction

Ring signatures, introduced by [RST01], are a variant of digital signatures which certify that *one among a particular set* of parties has signed a particular message, without revealing which specific party is the signer. This set is called a "ring."

Ring signatures can be useful, for example, to certify that certain leaked information comes from a privileged set of government or company officials without revealing the identity of the whistleblower, to issue important orders or directives without setting up the signer to be a scapegoat for repercussions, or to enable untraceable transactions in cryptocurrencies (as in Monero [Mon]).

In a ring signature scheme, just as in a traditional digital signature scheme, any party can create a key pair for signing and verification, and publish the verification key. Signers can produce signatures that verify with respect to any set of verification keys that includes their own, and unforgeability guarantees that no party can produce a valid signature with respect to a set of verification keys without possessing a corresponding secret key.

But what guarantee does a ring signature scheme provide if a purported signatory wishes to denounce a signed message — or alternatively, if a signatory wishes to later come forward and *claim* ownership of a signature? Given the motivation of anonymity behind the notion of a ring signature, a natural first intuition might be that parties should be able neither to denounce nor to claim a signature in a convincing way. However, depending on the threat model, we believe that the opposite guarantees — that is, to guarantee the ability to denounce or claim signatures — may be useful too, as elaborated below. Furthermore, whatever one's preference, a guarantee one way or the other seems more desirable than no guarantee either way.

Prior security definitions for ring signatures do not conclusively provide these guarantees one way or the other. That is, a non-signer might be able to *repudiate* a signature that he did not produce ("repudiability"), or this might be impossible ("unrepudiability"). Similarly, a signer might be able to later convincingly claim that a signature he produced is indeed his own ("claimability"), or be unable to do so ("unclaimability").

The most detailed taxonomy of security definitions for ring signatures was given by [BKM09], which presents a series of anonymity guarantees of increasing strength. A natural anonymity guarantee defined by [BKM09], called "anonymity against adversarially chosen keys," is informally described as follows: an adversary who controls all but $t \geq 2$ parties in a ring, and who may produce his own malformed key pairs as well as corrupt honest parties' keys, must have negligible advantage at guessing which of the t honest parties produced a given signature. This anonymity definition might allow a party to ascertain whether a given signature was produced by her own signing key, and perhaps also to convince others of this fact — but it does not *guarantee* or *prohibit* either of these capabilities.

On the other hand, the strongest of the anonymity definitions of [BKM09] (called "anonymity against full key exposure") requires that even if an adversary compromises every single party in a ring, the adversary cannot identify the signers of past signatures. It is relatively straightforward to see that under such a strong anonymity guarantee, Alice would have no way to convince anyone that she did not produce the objectionable message; indeed, she herself cannot tell the difference between a signature produced using her own signing key and one produced using someone else's.

The ability to identify whether one's own signing key was used to produce a particular signature can be a feature or a bug. To protect anonymity of past signatures against a very strong adversary who might compromise all the secret keys in a ring, it seems desirable to prevent distinguishing one's own signatures from those generated by someone else. On the other hand, without the ability to distinguish, it would be virtually impossible to tell if someone had stolen your signing key. Moreover, as discussed below, it could be beneficial in certain circumstances for members of a ring to have the ability to disown signatures of messages that they have strong reasons to denounce; and conversely, in some circumstances the signer of a message might later wish to prove to the world that he was the one who produced a particular signature in the past.

We have now identified four potentially useful notions for ring signatures: *repudiability*, *unrepudiability*, *claimability*, and *unclaimability*. The main contributions in this paper consist both of new definitions and constructions of each of these notions. Before diving into an overview of definition and constructions, we provide some discussion of why each of these notions — some of which directly oppose each other — may be meaningful and desirable: the following scenarios explore a few of the circumstances in which various of the above guarantees might be appropriate. Though some of the scenarios are phrased somewhat whimsically, we believe that each scenario illustrates a meaningful threat model motivating the definition concerned.

Scenario 1 (Repudiability) Let us consider a hypothetical tale, wherein two candidates Alice and Bob are running for president in the land of Oz. Oz is notorious for its petty partisan politics and its tendency to prefer whomever appears friendlier in a series of nationally televised grinning contests between the main-party candidates. At the peak of election season, a disgruntled citizen Eve decides to help out her preferred candidate Bob by publishing the following message, which goes viral on the social networks of Bob supporters:

I created a notorious terrorist group and laundered lots of money!
Signed: Alice or Eve or Alice's campaign chairman.

Of course, the virally publicized message does not actually incriminate Alice at all, since any one of the signatories could have produced it. However, perhaps there is nothing that Alice can do to allay the doubt in the minds of her suspicious detractors. As mentioned above, ring signatures are deliberately designed to allow *anyone* to attach anyone else's name to a signature, without the latter's knowledge or consent. Despite this, there could be realistic situations in which non-signing members of a ring associated with a particular message could suffer serious consequences through no fault of their own, perhaps due to the real signer adversarially trying to damage their reputation. In light of this, perhaps it would be desirable in some contexts for the owner of a verification key to be able to denounce messages, e.g., to clear her name of a crime or hate speech accusation that might otherwise impact her life in terms of reputation, job prospects, or incarceration.

Scenario 2 (Claimability) Our next story concerns a talented brewery employee who developed new statistical techniques to test the quality of beers. Naturally, his employer was protective of its competitive advantage since other breweries at the time may not have been using similar statistical methods. Yet, in the interest of science, they allowed him to publish his results — on condition of anonymity.[3] A credible way to prove authorship at a later date, after the need for anonymity has ceased to exist, might be very useful — especially in case of competing claims by impostors. As we see here, claiming authorship of an anonymous work may become appropriate after a passage of time. The next example illustrates quite a different type of situation in which claimability at the signer's discretion may be valuable.

Consider an employee Emily who is concerned about unethical practices at her company, and takes it upon herself to expose what is going on and publish a critical commentary. Concerned about her job security and possible retribution, as well as the credibility of her allegations, she maintains her anonymity using ring signatures. It emerges, in fact, that similar practices are prevalent across the industry: related revelations drive a wider movement of reform. Some time later, after her company has substantially reformed its practices and her fears of retribution have been allayed — perhaps by her promotion, or by a change in leadership — Emily seeks to reveal her identity and add her voice to the growing movement, providing her solidarity, legitimation, and follow-up story. In addition, if following the reforms, those involved in the earlier unethical practices were subject to stigma or even prosecution, claimability of her earlier ring signatures would allow Emily to exculpate herself.

Scenario 3 (Unrepudiability and unclaimability) Let us return to the government of the fictional country of Oz. The parliament of Oz is mired in partisan gridlock, with legislators from each party ruthlessly voting down any bills, however reasonable, proposed by members of the opposing party — preventing any laws at all from being enacted and effectively shutting down the government, which is in no party's interest. Suppose that instead of directly proposing a new law, a legislator of Oz anonymously publishes the text of the proposed bill using a ring signature scheme:

Proposed: that free ice cream shall be provided every Tuesday.[4]
Signed: a member of the Parliament of Oz.

If the signer used an unclaimable ring signature scheme, then she could not decide to reveal her identity upon a later change of heart, allowing legislators of both parties to support or oppose the bill on its merits without worrying about purely political considerations.

[3] This is the true story of William Sealy Gosset's invention of the Student's t-test at Guinness Brewery in 1908 [Man00].

[4] Even if each party might support this legislation, they may be unwilling to do so if it were proposed by the other party, decrying their respective opponents as either fiscally irresponsible or in the pocket of Big Ice Cream.

Unclaimability and unrepudiability may be particularly useful guarantees in scenarios where the placement of whole groups of people under duress is a substantial concern. For instance, in circumstances where an employer or authoritarian government may coercively compel individuals to provide a repudiation or proof of authorship (e.g. signing randomness) for a signature, the provable inability to do so convincingly may be essential. Unrepudiability may also be desirable in situations in which members of a ring are likely to have conflicting individual incentives but there is a possibility of collective benefit in case of cooperation, as in a prisoner's dilemma scenario.

Summary of technical contributions. We formalize *repudiability*, *unrepudiability*, *claimability*, and *unclaimability* of ring signatures, as well as strengthened anonymity and unforgeability definitions which are compatible with each of these notions. We show that *unclaimability* implies *unrepudiability* (intuitively, because a failed repudiation can be used as a claim). Anonymity against adversarially chosen keys is the strongest anonymity notion compatible with *repudiability* and *claimability*, and anonymity against full key exposure is implied by *unclaimability* and equivalent to *unrepudiability*.

We provide three constructions based on different assumptions, one for each of the three notions of *repudiability*, *claimability*, and *unclaimability*. Perhaps the most surprising of these is *unclaimability*, which guarantees that the signer cannot later credibly convince others that she produced a particular signature. A natural first intuition is that meaningful notions of unclaimability might be impossible to achieve, since a signer can always remember the signing randomness (and later present it as "proof" of having produced a signature). The key insight for our definition and construction of unclaimable ring signatures is that the signing randomness does not constitute a convincing claim if *anyone in the ring can also produce credible signing randomness* for any signature in which they are implicated. Our construction of *unclaimable* ring signatures is an augmentation of the lattice-based ring signature scheme of [BK10] that adds additional algorithms allowing anyone in the ring to generate credible signing randomness; this capability is achieved via lattice trapdoors.

Our construction of *repudiable* ring signatures is based on verifiable random functions (VRFs), which are implied by either the (strong) RSA assumption, assumptions on bilinear maps, or NIWIs and commitments; see [Bit17,GHKW17] and references therein for more detailed discussion of the assumptions that imply VRFs. Our construction does not use standard ring signatures as a building block, and as such can also be viewed as a new construction of standard ring signatures. Our construction of *claimable* ring signatures, on the other hand, is a simple and generic black-box transformation from any standard ring signature scheme to a claimable one. We overview our contributions in more detail below.

1.1 Definitional contributions

Repudiability. We define a *repudiable ring signature scheme* as a ring signature scheme that is equipped with additional algorithms Repudiate and VerRepud as follows. Repudiate takes as input a signing key sk, a ring signature σ, and a "ring" R (i.e., a set of verification keys), and outputs a *repudiation* ξ. VerRepud takes as input a ring R, a signature σ, a repudiation ξ, and a verification key vk, and outputs a a single bit indicating whether or not ξ is a valid repudiation attesting that σ was not produced by vk. The two requirements for a ring signature scheme to be *repudiable* are, informally, as follows.

1. *Correctness:* Any member of a ring must be able to produce valid repudiations of any signature that he did not produce.
2. *Soundness:* A cheating signer must not be able to produce a valid signature with respect to a ring, and also be able to produce valid repudiations of that signature under every verification key in that ring that he owns.

Once a ring signature scheme is equipped with these additional repudiation algorithms, the standard definitions of *unforgeability* and *anonymity* against adversarially chosen keys are insufficient to capture the natural guarantees that would be desired for a repudiable ring signature scheme: we need the release of repudiations not to compromise the unforgeability or anonymity of any future signatures. Accordingly, we modify the definitions of unforgeability and anonymity for repudiable ring signatures (Definitions 12 and 13), by additionally giving the adversary access to a *repudiation oracle*. This ensures that repudiations of past signatures do not affect the security guarantees of future signatures. See Section 3.1 for formal definitions of repudiability.

Claimability. We define a *claimable ring signature scheme* as a ring signature scheme equipped with additional algorithms Claim and VerClaim as follows. Claim takes as input a signing key sk, a signature σ, and a ring R, and outputs a claim ζ. VerClaim takes a input a ring R, a verification key vk, a signature σ, and a claim ζ, and outputs a single bit indicating whether or not ζ is a valid claim attesting that σ was produced by vk. The three requirements for a claimable ring signature scheme are, informally, as follows.

1. *Correctness:* Any honest signer must be able to produce a valid claim with respect to any signature that he produced.
2. *Soundness:* No adversary can produce a valid claim with respect to a signature produced by an honest signer, even if the adversary can choose the message and ring with respect to which the signature is produced, and can insert malformed verification keys into the ring.
3. *No framing:* No adversary can produce a signature together with a valid claim of that signature on behalf of an honest (non-signing) party.

As above, once a ring signature scheme is equipped with these additional claiming algorithms, the standard definitions of *unforgeability* and *anonymity* against adversarially chosen keys are insufficient. We modify the definitions of anonymity and unforgeability for claimable ring signatures (Definitions 18 and 19), by additionally giving the adversary access to a *claim oracle*. See Section 3.3 for formal definitions of repudiability.

Repudiability and *claimability* are compatible, i.e., a ring signature scheme can be both repudiable and claimable. Indeed, our repudiable and claimable constructions together give rise to such a scheme. Notably, the unforgeability and anonymity definitions corresponding to the natural notion of a repudiable-and-claimable ring signature scheme are *not* the conjunction of unforgeability and anonymity for repudiable ring signatures and for claimable ring signatures. Rather, the unforgeability and anonymity definitions for a repudiable-and-claimable ring signature scheme involve a stronger adversary which is simultaneously given access to both a repudiation oracle and a claim oracle. See Section 3.5 for further discussion on repudiable-and-claimable schemes.

Unclaimability We also introduce *unclaimable* ring signature schemes, in which the signer *provably cannot* convincingly claim that she was the one who produced the signature. As briefly mentioned above, while the signer can always save the signing randomness and reveal it along with her secret key in an attempt to claim authorship of a signature, it is not always true that this constitutes a convincing claim. In particular, such a claim is not credible if *any* member of the ring can take a valid signature and produce fake randomness that produces the desired signature using her own signing key.

The idea that a non-signer can adaptively produce fake randomness is reminiscent of deniable encryption [CDNO97], in which an encryptor and/or recipient is required to produce fake randomness "explaining" that a particular ciphertext is an encryption of an adversarially chosen message.

We define an *unclaimable* ring signature scheme to capture just this requirement: that is, any member of the ring must be able to produce fake signing randomness for a signature that is distributed indistinguishably from real signing randomness. Intuitively, the only information potentially possessed by a signer but not by the other members of the ring is the signing randomness, so non-signers that can generate convincing simulated signing randomness can also convincingly simulate any additional information that might be released by the signer in an attempt to claim the signature. We consider a strong flavor of this definition in which the indistinguishability property, described informally below, is statistical.

1. *Indistinguishability:* Any member of a ring must be able to produce fake signing randomness given a signature. The signature and fake signing randomness must be distributed statistically close to an honestly generated signature and corresponding signing randomness used by that individual to sign the same message, even given all verification keys and signing keys.

.

Remark 1. Even under this definition, if the signer chooses a message to sign that corresponds to a secret known only to herself, then she may still be able to convince others that she was the signer. For instance, if the signed message is the output of a one-way function, she may be able to convince others that she was the signer by subsequently revealing the preimage. Even more flagrantly, the signed message could contain a signature using a standard (non-ring) signature scheme, directly identifying the signer. This property is rather inherent: if knowledge of the contents of the message itself at the time of signing are enough to identify the signer, then no security property on the signature scheme can enforce that

Repudiable	VRF (Section 4)
Unrepudiable	RS anonymous against FKE (Section 3.2)
Claimable	Transformation from any RS (Section 5)
Unclaimable	SIS (Section 6)

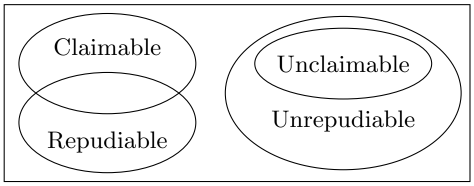

Fig. 1: Summary of our results and assumptions relied on. VRF = verifiable random function, RS = ring signature, FKE = full key exposure, SIS = short integer solution problem.

the signer remains hidden, since the identification of the signer is unrelated to the signature and based only on the signed message.

Indeed, ring signatures were not designed to provide anonymity for signers who *want* to identify themselves, but rather for those who desire anonymity. Similarly, our unclaimability definition does not guarantee unclaimability for those who *want* to identify themselves, but rather provides credibility for a signer who *wants* to later be able to claim (e.g., under duress) that she could not convincingly claim the signature even if she wanted to. In particular, even an adversary with unlimited computational power who obtains the secret keys belonging to every member of the ring and a purported signing randomness from an alleged signer, he still will not be convinced of the identity of the signer, since fake signing randomness from the right distribution can be produced for every member of the ring.

Unrepudiability Unclaimability intuitively guarantees that no member of the ring can convincingly prove that she was the signer. A related, weaker notion that might be desirable in some circumstances is that of *unrepudiability*, which guarantees that no member of the ring can convincingly prove that she was *not* the signer. Unrepudiability is equivalent to anonymity against full key exposure and is implied by unclaimability.

1.2 Overview of our constructions

Our repudiable construction Our construction relies on ZAPs (two-round public-coin witness-indistinguishable proofs) and verifiable random functions (VRFs) as building blocks.[5] Our building blocks have some overlap with those of the ring signature construction of [BKM09], which uses ZAPs, public-key encryption (PKE), and a digital signature scheme. Both our scheme and theirs use ZAPs to achieve anonymity of the ring signatures, but with different approaches: the statements proven by the ZAPs are quite unrelated in the two constructions. Moreover, in our scheme, we do not need PKE or signature schemes, and instead

[5] VRFs imply ZAPs, so it suffices to assume VRFs. [GO92,DN07]

use VRFs directly to achieve unforgeability and repudiability. The structure of our construction is thus very different from that of [BKM09].

At a very high level, each signing key in our construction contains a tuple of four VRF keys. A signature consists of the output of each of the signer's VRFs on the message, along with a ZAP proof that (several of) the VRF values in the signature are correct w.r.t. the VRF verification key of some member of the ring. A repudiation for individual i consists of a ZAP proof that some of the VRF values in the signature are different from the correct values for party i's VRFs evaluated at the message. One complication arises because we must guarantee that the release of a repudiation for individual i on a message does not subsequently allow a different member of the ring to produce a signature on the message that cannot be repudiated by individual i. We overcome this difficulty by relying on the witness indistinguishability property of the ZAP and ensuring that the repudiation does not reveal the actual VRF outputs of the repudiator; that is, the ZAP proof is produced with the VRF proof as a *witness*. The specific statement proven by the ZAPs is that some specific combination of at least two of the purported VRF outputs is correct. Although in the honest usage of the scheme, all four are produced correctly, we design the specific structure of the statements proved in order to allow a hybrid argument to argue indistinguishability between signatures of different signers in a ring. This scheme of proving the correctness of VRF outputs turns out also to imply unforgeability, not only repudiability, so we do not need to rely on any underlying signature scheme as building block. (In other words, our scheme can also be seen as a new construction of standard ring signatures based on VRFs.)

Our claimable construction We give a generic transformation from any standard ring signature scheme RS to a claimable one. The transformation uses commitment schemes, standard signatures, and PRFs (which are all achievable from one-way functions). The basic idea is to take a signature σ_{RS} under RS and append to it a *commitment* c to $(vk, \sigma_{\mathsf{RS}})$ where vk is the verification key of the signer. The verification algorithm simply checks whether σ_{RS} verifies. The claim consists of a decommitment revealing that c is a commitment to $(vk, \sigma_{\mathsf{RS}})$. Intuitively, by the hiding property of the commitment scheme, the identity of the signer is hidden until he chooses to publish a claim.

The simple transformation just described runs into a couple of problems when examined in detail. First, what if a signer commits to $(\sigma_{\mathsf{RS}}, vk')$ where vk' is not his own key but that of someone else in the ring? This ability would violate equation (6) of Definition 17 (claimability). To prevent such behavior, our construction actually commits to a *standard (non-ring) signature* on $(vk, \sigma_{\mathsf{RS}})$. The unforgeability property of standard signatures then guarantees, intuitively, that a signer cannot convincingly make a claim with respect to any verification key unless he knows a corresponding signing key.

A second issue encountered by the scheme thus far described is that the signer must remember the commitment randomness in order to produce a claim. It is preferable that the signer not be stateful between signing and claiming; indeed, Definition 17 requires this. To resolve this, our construction derives commitment randomness from a PRF. For similar reasons, the signing randomness for the standard (non-ring) signature in our construction is also derived from a PRF.

Remark 2. Among the constructions presented in this paper, claimability is by far the simplest. Moreover, as a generic transformation, it has the advantage of adding minimal efficiency overhead to the existing state of the art in ring signatures. The simplicity of achieving claimability is perhaps unsurprising in light of the natural intuition that claiming should be possible simply by remembering the signing randomness. As evidenced by *un*claimability, this intuition is not strictly true in general, as in certain schemes, producing signing randomness may not prove authorship. In a nutshell, our generic transformation ensures that signing randomness is indeed a convincing proof of authorship in the resulting scheme, and moreover builds into the scheme a simple method of efficiently recovering the signing randomness without storing it explicitly.

Our unclaimable construction Our construction of unclaimable ring signatures is an extension of the SIS-based ring signature scheme of Brakerski and Kalai [BK10]. The construction is based on trapdoor sampling. In this overview, we describe a simplified version of the scheme. The full scheme is described in Section 6. The basic idea for obtaining unclaimability is that each identity corresponds to a public matrix $A_i \in \mathbb{Z}_q^{n\times m}$ sampled together with a secret trapdoor T_i. A signature will consist of short vectors $x_i \in \mathbb{Z}_q^m$ such that

$$\sum_i A_i x_i = y,$$

where y is a target value. For this overview, we can think of y as the output of a random oracle on the message; in the actual construction, y will be obtained as the sum of additional matrix-vector products. In order to sign the message, signer i first samples short vectors x_j for each $j \neq i$. Then, using the lattice trapdoor T_i, he samples a short vector x_i such that the equation

$$x_i = y - \sum_{j\neq i} A_j x_j$$

is satisfied. The signature is the list of vectors $\sigma = (x_i)_i$. Using properties of lattice trapdoors, it follows that the distribution over $(x_i)_i$ can be made statistically close no matter which trapdoor was used to produce the signature. Moreover, given a vector x^* to be produced, we can sample random coins that will yield that vector under either the ordinary sampling algorithm or the trapdoor sampling algorithm. Consequently, we obtain an algorithm that can produce explanatory randomness for a signature under any identity in the ring.

Removing the random oracle to obtain ring signatures in the plain model (and unclaimable ones) requires several complications. [BK10] first describes a basic ring signature scheme with weaker unforgeability properties, in which the target vector y is determined using additional matrix-vector products for matrices that depend on the bits of the message. They then amplify the security of the scheme through a sequence of transformations that ultimately yield a scheme with full unforgeability. In Section 6, we first define an algorithm for producing explanatory randomness for their basic scheme, and then describe how to modify this algorithm for each modification of the basic scheme, ultimately yielding an unclaimable ring signature scheme based on the SIS assumption.

Remark 3. The idea that a non-signer of a given signature can adaptively produce fake signing randomness is reminiscent of deniable encryption [CDNO97], in which an encryptor of a given ciphertext can adaptively produce fake randomness consistent with it being an encryption of a different message. In this context, it may seem somewhat surprising that our construction relies on a relatively standard assumption (SIS) while many natural definitions of deniable encryption are not known to be achievable without heavier assumptions such as indistinguishability obfuscation [SW14,CPP18]. A subtle difference that is significant here is that a deniably encrypted message must still be recoverable by the honest decryptor, while in the unclaimable ring signature setting, the signer's identity need not be recoverable by anyone.

1.3 Other related work

Several constructions of ring signatures based on lattice assumptions have been proposed (e.g., [BK10,MBB$^+$13,BLO18]). The only other construction of ring signatures based on ZAPs is [BKM09], to our knowledge. Numerous other ring signature constructions have been proposed, mostly based on various assumptions on bilinear maps, many but not all of which are in the random oracle model (e.g., [Ngu05,SS10,BCC$^+$15]).

Two additional works in the lattice trapdoor literature bear mentioning: the seminal [Ajt99], and the more recent [MP12]. The latter is more recent than [GPV08], whose trapdoors our unclaimable construction relies on (this reliance is carried over from the [BK10] construction).

Ring signatures with additional guarantees Since the original proposal of ring signatures by [RST01], various variant definitions have been proposed. For example, *linkable* ring signatures [LWW04] allow identification of signatures that were produced by the same signer, without compromising the anonymity of the signer within the ring. An enhancement to this notion called *designated linkability* [LSW06] does not allow linkability by default, but instead allows links to be revealed at will by a designated party. Another notion called *traceable* ring signatures [FS07] considers a setting where signatures are generated with respect to "tags" and each member may sign at most a single message (say, a vote) with respect to a particular tag, or else his identity will be revealed. *Accountable* ring signatures [XY04,BCC$^+$15] allow a signer to assign the power to de-anonymize her signature to a specific publicly identified party.

It may seem that some of these variants of ring signature schemes have properties that would be useful for constructing *claimable* ring signatures as introduced in this paper. This implication is unsurprising in the context of our results: *all* of the above types of ring signature schemes in fact imply claimable ring signatures, since our construction of claimable ring signatures is a generic transformation from *any* ring signature scheme. It is unclear if leveraging the additional features of variant schemes would be more desirable than applying our generic transformation, which has very low overhead and moreover can be applied to a simpler, more efficient ring signature scheme that may lack these additional properties.

Group signatures Group signatures [CvH91] are a different type of signature that allow signing w.r.t. a set of verification keys and provide anonymity of the signer within that set. This concept differs most strikingly from ring signatures in that there is a central authority that (1) sets up the group (i.e., set of signers) and issues keys to members of the group and (2) has the power to revoke the anonymity of the signer of a signature. Notions such as (un)linkability, described above, have been applied to the group signature setting as well. Notably, there has also been proposed a notion of *deniable group signatures* [IEH$^+$16], in which the group manager may issue proofs that a particular group member did *not* sign a particular signature. This bears a little resemblance to our notion of *repudiability* in ring signatures; however, the presence of a central authority in the group signature setting means these problems are technically rather disparate. [LNWX17] construct lattice-based deniable group signatures; however, their technique for deniability is very different from ours, and relies on zero-knowledge proofs of plaintext inequality for LWE ciphertexts, which do not suffice in our setting.

2 Anonymity and unforgeability of ring signatures

This section overviews standard ring signature definitions: syntax, correctness, anonymity, and unforgeability. We express the anonymity and unforgeability definitions differently from prior work, as explained in their respective subsections. However, our definitions are equivalent to the correspondingly named definitions from prior work. Throughout the paper, k denotes the security parameter.

Definition 1 (Ring signature). *A* ring signature scheme *is a triple of PPT algorithms* $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$*, satisfying the three properties of* correctness *(Definition 2),* anonymity *(Definitions 5–6), and* unforgeability *(Definition 8). The syntax of* Gen*,* Sign*, and* Verify *follows.*

- $\mathsf{Gen}(1^k)$ *takes k as input and outputs* verification key vk *and* signing key sk*.*
- $\mathsf{Sign}(R, sk, m)$ *takes as input a signing key sk, a message m, and a set of verification keys $R = \{vk_1, \dots, vk_N\}$, and outputs a signature σ. The set R is also known as a "ring."*
- $\mathsf{Verify}(R, \sigma, m)$ *takes as input a set R of verification keys, a signature σ, and a message m, and outputs a single bit indicating whether or not σ is a valid signature on m w.r.t. R.*

Where it may not be clear from context, we sometimes write $\mathsf{RS.Gen}, \mathsf{RS.Sign}, \mathsf{RS.Verify}$ *to denote the* $\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify}$ *algorithms belonging to* RS*.*

Definition 2 (Correctness). *A ring signature scheme* $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfies* correctness *if there is a negligible function ε s.t. for any $N = \mathsf{poly}(k)$,* *any* $(vk_1, sk_1), \ldots, (vk_N, sk_N) \leftarrow \mathsf{Gen}(1^k)$*, any* $i \in [N]$*, and any message* m*,*

$$\Pr\left[\mathsf{Verify}(R, \mathsf{Sign}(R, sk_i, m), m) = 1\right] = 1 - \varepsilon(k) \ , \tag{1}$$

where $R = \{vk_1, \ldots, vk_N\}$*.* RS *satisfies* perfect correctness *if* (1) *holds for* $\varepsilon = 0$*.*

2.1 Anonymity

Prior work, notably [RST01,BKM09], has presented several ring signature anonymity definitions. Two of the definitions from prior work are relevant to this paper: anonymity against adversarially chosen keys and against full key exposure.

This section presents a new, generalized anonymity definition parametrized by *oracle sets*, and expresses the two relevant anonymity definitions as instantiations of the generalized definition. This generalized definition is useful to consolidate the existing definitions and make clear their relationship to one another; it captures not only the two definitions we rely on here, but also others from prior work. Moreover, the generalized definition will be essential to concisely express the new anonymity definitions that we introduce in later sections for anonymity of *repudiable* and *claimable* ring signature schemes (in Sections 3.1 and 3.3 respectively). In a nutshell, this is because the new definitions need to allow the adversary access to additional oracles related to repudiation and/or claiming.

The generalized definition follows. It is parametrized by sets of oracles $\mathcal{O}_1, \mathcal{O}_2$ and an additional parameter $\alpha \in \{0, 1, 2\}$ that limits the adversary's corruptions.

Definition 3 ($(\mathcal{O}_1, \mathcal{O}_2, \alpha)$-anonymity). *Let* $\mathcal{O}_1, \mathcal{O}_2$ *be sets of oracles, where each oracle in the set is parametrized by a list of key-pairs. Define* $\mathsf{Corr}_{(vk_1, sk_1), \ldots, (vk_N, sk_N)}$ *to take as input* $i \in [N]$ *and output* $\omega_i \leftarrow \mathsf{Gen}^{-1}(vk_i, sk_i)$*.*[6]

A ring signature scheme $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfies* $(\mathcal{O}_1, \mathcal{O}_2, \alpha)$-anonymity *if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $N = \mathsf{poly}(k)$*,* $\Pr[b' = b]$ *in the above game is negligibly close to* $1/2$*. That is, formally,* $\forall$ *PPT* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,* $N = \mathsf{poly}(k)$*, there is a negligible function* ε *such that*

$$\Pr\left[\begin{array}{l} (vk_1, sk_1), \ldots, (vk_N, sk_N) \leftarrow \mathsf{Gen}(1^k) \\ ((m^*, i_0^*, i_1^*, R^*), \mathfrak{s}) \leftarrow \mathcal{A}_1^{\mathcal{O}_1, \mathsf{Corr}}(vk_1, \ldots, vk_N) \\ b \leftarrow \{0, 1\} \\ \sigma \leftarrow \mathsf{Sign}(R^* \cup \{vk_{i_0^*}, vk_{i_1^*}\}, sk_{i_b^*}, m^*) \\ b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2, \mathsf{Corr}}(\mathfrak{s}, \sigma) \end{array} : b' = b \wedge |\{i_0^*, i_1^*\} \cap I| \leq \alpha \right] < \frac{1}{2} + \varepsilon(k) \ , \tag{2}$$

[6] The function Gen^{-1} takes as input a verification key vk and signing key sk produced by Gen, and produces the randomness used by Gen to produce this key pair. That is, it samples from the set $\{\omega : \mathsf{Gen}(1^k; \omega) = (vk, sk)\}$. In practice we will only ever invoke Gen^{-1} on a key pair produced by Gen, so we could invert efficiently by simply remembering the randomness used by Gen, but for the purposes of this definition we will describe it as a sampling procedure. Upon the first invocation on an input i, Corr samples $\omega_i \leftarrow \mathsf{Gen}^{-1}(vk_i, sk_i)$, stores it, and outputs it. If Corr is queried twice on the same input i then it outputs the same ω_i that was previously stored.

where I is the set of queries to the corruption oracle; and the notation $\mathcal{A}^{\mathcal{O},\mathsf{Corr}}$ means that for each oracle O in $\mathcal{O}$, $\mathcal{A}$ has oracle access to $O_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$, and $\mathcal{A}$ also has oracle access to $\mathsf{Corr}_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$.

Definitions 5 and 6 are instantiations of Definition 3. They are equivalent to the correspondingly named definitions in [BKM09].

Definition 4 (Signing oracle OSign). *For a ring signature scheme RS, the oracle $\mathsf{OSign}_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$ is defined to take as input $i \in [n]$, a message m, and a set R, and output $\mathsf{RS}.\mathsf{Sign}(R \cup \{vk_i\}, sk_i, m)$. When the oracle is invoked with respect to a single key pair (i.e., $\mathsf{OSign}_{(vk,sk)}$), we treat the oracle as taking only two inputs, m and R, since i is superfluous in this case.*

Definition 5 (Anonymity against adversarially chosen keys). *A ring signature scheme $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ satisfies* anonymity against adversarially chosen keys *if it is $(\{\mathsf{OSign}\}, \varnothing, 0)$-anonymous. Moreover, RS satisfies* adaptive anonymity against adversarially chosen keys *if it is $(\{\mathsf{OSign}\}, \{\mathsf{OSign}\}, 0)$-anonymous.*

Definition 5 captures the guarantee that as long as there are at least two honest parties in a ring (represented by i_0^*, i_1^*), even if all other parties in the ring are corrupted by an adversary, the adversary cannot tell which of the honest parties produced a signature. One can also consider an even stronger definition where the adversary may corrupt *all but one* or even *all* of the parties in the ring, as in Definition 6.

Definition 6 (Anonymity against full key exposure). *A ring signature scheme $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ satisfies* anonymity against full key exposure *if it is $(\{\mathsf{OSign}\}, \varnothing, 2)$-anonymous.*

Remark 4. Adaptive variants of anonymity were not discussed in prior work. In this paper, we refer primarily to *adaptive anonymity against adversarially chosen keys*: this is the strongest notion compatible with repudiability and claimability. Definition 6 does not include an adaptive version because adaptivity does not give the adversary any additional power when he can corrupt all the keys.

2.2 Unforgeability

The first unforgeability definition that follows is parametrized by an oracle set, taking a similar approach to our anonymity definitions above. In this section, we only give one instantiation of the parametrized definition of unforgeability. We will give other instantiations of Definition 7 in Sections 3.1 and 3.3.

Definition 7 ($\mathcal{O}$-unforgeability). *Let $\mathcal{O}$ be a set of oracles, where each oracle in the set is parametrized by a list of key-pairs. A ring signature scheme $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is* $\mathcal{O}$-unforgeable *if for any PPT $\mathcal{A}$ and any $N = \mathsf{poly}(k)$, there is a negligible function ε such that*

$$\Pr\left[\begin{array}{l}(vk_1,sk_1),\ldots,(vk_N,sk_N) \leftarrow \mathsf{Gen}(1^k)\\ (R^*,m^*,\sigma^*) \leftarrow \mathcal{A}^{\mathcal{O},\mathsf{OSign},\mathsf{Corr}}(vk_1,\ldots,vk_N)\\ b \leftarrow \mathsf{Verify}(R^*,\sigma^*,m^*)\end{array} : \begin{array}{l} b = 1 \wedge R^* \subseteq \{vk_1,\ldots,vk_N\} \setminus I \\ \wedge\, Q \cap \{(\cdot,m^*,R^*)\} = \varnothing\end{array}\right] < \varepsilon(k)\ ,$$

where the notation $\mathcal{A}^{\mathcal{O},\mathsf{OSign},\mathsf{Corr}}$ is defined as in Definition 3, and I and Q are the sets of queries made to the corruption and signing oracles respectively.

We refer to the event that the conditions on the right-hand side of the colon in the above probability expression are met as a "successful forgery."

Definition 8 (Unforgeability of ring signatures). *A ring signature scheme* $\mathsf{RS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* unforgeable *if it is $\varnothing$-unforgeable.*

3 New definitions: (un)repudiability and (un)claimability

3.1 Repudiable ring signatures

Repudiability addresses the question of whether ring members can prove that they did *not* sign a particular message (when they in fact did not sign it).

Definition 9 (Repudiable ring signature). *A* repudiable ring signature scheme *is a ring signature scheme with an additional pair of algorithms* $(\mathsf{Repudiate}, \mathsf{VerRepud})$, *satisfying the four properties of* correctness *(Definition 2),* repudiability *(Definition 11),* anonymity *(Definition 12), and* unforgeability *(Definition 13). The syntax of* $\mathsf{Repudiate}$ *and* $\mathsf{VerRepud}$ *follows.*

- $\mathsf{Repudiate}(R, sk, \sigma)$ *takes as input a signing key sk, a ring signature σ, and a set of verification keys $R = \{vk_1, \dots, vk_N\}$, and outputs a repudiation ξ.*
- $\mathsf{VerRepud}(R, vk, \sigma, \xi)$ *takes as input a set R of verification keys, a signature σ, a repudiation ξ, and an identity vk, and outputs a single bit indicating whether or not ξ is a valid repudiation of signature σ for identity vk.*

Definition 10 (Repudiation oracle ORpd). *For a repudiable ring signature scheme* RS, *the oracle* $\mathsf{ORpd}_{(vk_1,sk_1),\dots,(vk_N,sk_N)}$ *is defined to take as input $i \in [n]$, a signature σ, and a set R, and output* $\mathsf{RS}.\mathsf{Repudiate}(R \cup \{vk_i\}, sk_i, \sigma)$. *When the oracle is invoked with respect to a single key pair (i.e.,* $\mathsf{ORpd}_{(vk,sk)}$*), we treat the oracle as taking only two inputs, σ and R, since i is superfluous in this case.*

Additionally, we define the oracle $\mathsf{ORpd}^{\langle\sigma^*\rangle}_{(vk_1,sk_1),\dots,(vk_N,sk_N)}$ *to output $\perp$ when it receives the signature σ^* as input, and otherwise to give the same response as* $\mathsf{ORpd}_{(vk_1,sk_1),\dots,(vk_N,sk_N)}$.

Repudiability requires two conditions, expressed by equations (3) and (4) below. Intuitively, (3) captures the requirement "good people can repudiate," i.e., that for any (possibly maliciously generated) signature, an honest party who did not produce it should be able to successfully repudiate. (4) captures the requirements that "bad people cannot repudiate a signature they produced," i.e., addressing the case where the malicious signature and repudiation are both produced using the key being verified, and thus we want the signer to be unable to produce a valid repudiation.

Definition 11 (Repudiability). *A ring signature scheme* $\varSigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfies* repudiability *if equipped with algorithms* $(\mathsf{Repudiate}, \mathsf{VerRepud})$ *such that the following conditions hold.*

1. (Non-signers can repudiate) *Let* $\mathcal{O} = \{\mathsf{OSign}\}$. *For any (possibly adversarial) PPT signing algorithm* $\mathcal{A}_{\mathsf{Sign}}$*, there exists a negligible function* ε *such that*

$$\Pr\left[\begin{array}{l} (vk, sk) \leftarrow \mathsf{Gen}(1^k) \\ (\sigma, m, R') \leftarrow \mathcal{A}_{\mathsf{Sign}}^{\mathcal{O},\mathsf{ORpd}_{(vk,sk)}}(vk) \\ \xi \leftarrow \mathsf{Repudiate}(R', sk, \sigma) \\ b \leftarrow \mathsf{VerRepud}(R', vk, \sigma, \xi) \\ b' \leftarrow \mathsf{Verify}(R', \sigma, m) \end{array} : \begin{array}{l} b = 1 \vee b' = 0 \\ \vee Q \cap \{(\cdot, m, R')\} \neq \varnothing \end{array}\right] > 1 - \varepsilon(k) \ . \tag{3}$$

2. (Signer cannot repudiate) *For any (possibly adversarial) sign-and-repudiate algorithm* $\mathcal{A}_{\mathsf{S\&R}}$*, there is a negligible function* ε *such that for any* $N = \mathsf{poly}(k)$*,*

$$\Pr\left[\begin{array}{l} (vk_1, sk_1), \ldots, (vk_N, sk_N) \leftarrow \mathsf{Gen}(1^k) \\ (\sigma, R', m, \{\xi_{vk}\}_{vk \in R' \setminus R}) \leftarrow \mathcal{A}_{\mathsf{S\&R}}^{\mathcal{O}}(R) \\ \forall vk \in R' \setminus R,\ b_{vk} \leftarrow \mathsf{VerRepud}(R', vk, \sigma, \xi_{vk}) \\ b' \leftarrow \mathsf{Verify}(R', \sigma, m) \end{array} : \begin{array}{l} R' \cap R = \varnothing \vee \displaystyle\bigvee_{vk \in R' \setminus R} b_{vk} = 0 \\ \vee b' = 0 \vee Q \cap \{(\cdot, m, R')\} \neq \varnothing \end{array}\right] > 1 - \varepsilon(k) \ , \tag{4}$$

where $R = \{vk_1, \ldots, vk_N\}$*,* $\mathcal{O} = \{\mathsf{OSign}, \mathsf{ORpd}\}$*, and* Q *is the set of* OSign *queries.*

Remark 5. Equation 4 guarantees that a party possessing a set of signing keys cannot repudiate under all of these keys, *as long as some key in the ring is honestly generated.* If the adversary generates all keys in the ring, then he may be able to produce a repudiation under every key in the ring. However, this does not undermine the purpose of repudiability: indeed, if presented with repudiations under every key in a ring, one can confidently conclude that all keys in the ring were generated dishonestly, and thus that all parties in the ring effectively colluded to produce each signature under that ring. Similarly, given repudiations for a subset of the identities in a ring, one can conclude that *either* one of the remaining identities in the ring produced the signature *or* all of the remaining identities in the ring colluded maliciously to produce the signature. That is, either way, at least one of the remaining identities is responsible for the signature.

Anonymity and unforgeability of repudiable ring signatures The definitions of anonymity and unforgeability need to be adapted for repudiable ring signature schemes, to incorporate a repudiation oracle as described next.

Definition 12 (Anonymity of repudiable ring signatures). *A repudiable ring signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify}, (\mathsf{Repudiate}, \mathsf{VerRepud}))$ *satisfies* anonymity against adversarially chosen keys *if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\{\mathsf{OSign}, \mathsf{ORpd}\}, \varnothing, 0)$-*anonymous (Definition 3) Moreover, it satisfies* adaptive *anonymity against adversarially chosen keys if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\{\mathsf{OSign}, \mathsf{ORpd}\}, \{\mathsf{OSign}, \mathsf{ORpd}^{\langle\sigma\rangle}\}, 0)$-*anonymous, where* σ *is the challenge signature in Equation 2.*

Recall from Remark 4 that adaptive anonymity against adversarially chosen keys is the strongest anonymity notion compatible with repudiability.

Definition 13 (Unforgeability of repudiable ring signatures). *A repudiable ring signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify}, (\mathsf{Repudiate}, \mathsf{VerRepud}))$ *is unforgeable if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $\{\mathsf{ORpd}\}$-*unforgeable (Definition 7).*

3.2 Unrepudiable ring signatures

We next consider a notion where it is *not* possible for a party to prove to others that he did not produce a particular signature. In fact, though it may not be immediately apparent, a natural formalization of this notion is expressed by the definition of *anonymity against full key exposure* (Definition 6): that is, the strongest of the anonymity definitions given in Section 2. The following paragraphs justify this claim with detailed intuition.

Recall that anonymity against full key exposure (FKE) preserves signer anonymity even against an adversary that obtains all of the secret keys of all members of a ring. A ring signature scheme that satisfies repudiability could not also satisfy anonymity against FKE, because of the following attack: the adversary obtains all secret keys in the ring, attempts to repudiate using each secret key, and identifies as the signer the one secret key with respect to which the repudiation algorithm does not produce a valid repudiation. With overwhelming probability, by definition of repudiability, there is exactly one such secret key.

This informal argument establishes that anonymity against FKE must imply any reasonable notion of unrepudiability. Then are the two notions equivalent? While there arguably exist meaningful definitions of unrepudiability that are weaker than anonymity against FKE, we believe anonymity against FKE is the most reasonable definition of unrepudiability, as explained next.

Any reasonable definition of unrepudiability should capture the intuitive requirement that non-signers cannot behave distinguishably from signers. A little more precisely, for any protocol that could be executed by a non-signer Nancy with respect to a signature σ and her verification key vk', the signer Sigmund of that signature must be able to engage in the same protocol with respect to his own verification key vk and behave indistinguishably from Nancy. In other words, we require that if Nancy's secret key were stolen, the thief would be unable to tell whether σ was produced by Nancy or by someone else. Indeed, if Nancy were stateless and did not remember what signatures she had produced in the past, or simply lent her secret key to someone else who used it to produce signatures, then she herself would not be able to tell. The definition of anonymity against FKE embodies almost exactly this requirement — but instead of requiring anonymity against the thief who steals just Nancy's key, the definition makes the stronger requirement that anonymity must hold even against a thief who has every secret key in the ring corresponding to σ.

Is a weaker definition, which only rules out *unilateral* repudiations by a single party, a meaningful definition of unrepudiability? Perhaps. However, it is more in keeping with the intuitive goals and standard properties of ring signatures to protect against adversaries that may have many or all secret keys in a ring: that is, to rule out even the possibility of multiple ring members *colluding* to produce a repudiation for some ring member. Thus we arrive at the following definition.

Definition 14 (Unrepudiable ring signature scheme). *A ring signature scheme is* unrepudiable *if it satisfies anonymity against full key exposure.*

3.3 Claimable ring signatures

Claimability addresses whether the actual signer can prove later that they were the signer, without remembering the signing randomness.

Definition 15 (Claimable ring signature). *A* claimable ring signature scheme *is a ring signature scheme with an additional pair of algorithms* $(\mathsf{Claim}, \mathsf{VerClaim})$*, satisfying the four properties of* correctness *(Definition 2),* claimability *(Definition 17),* anonymity *(Definition 18), and* unforgeability *(Definition 19). The syntax of* Claim *and* $\mathsf{VerClaim}$ *follows.*

- $\mathsf{Claim}(R, sk, \sigma)$ *takes as input a signing key* sk*, a ring signature* σ*, and a set of verification keys* $R = \{vk_1, \ldots, vk_N\}$*, and outputs a claim* ζ*.*
- $\mathsf{VerClaim}(R, vk, \sigma, \zeta)$ *takes as input a set* R *of verification keys, a signature* σ*, a claim* ζ*, and an identity* vk*, and outputs a single bit indicating whether or not* ζ *is a valid claim of signature* σ *for identity* vk*.*

Definition 16 (Claim oracle OClaim). *For a claimable ring signature scheme* RS*, the oracle* $\mathsf{OClaim}_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$ *is defined to take as input* $i \in [n]$*, a set* R*, and a signature* σ*, and output* $\mathsf{RS.Claim}(R, sk, \sigma)$*. When the oracle is invoked with respect to a single key pair (i.e.,* $\mathsf{OClaim}_{(vk,sk)}$*), we treat the oracle as taking only two inputs,* R *and* σ*, since* i *is superfluous in this case.*

Additionally, we define the oracle $\mathsf{OClaim}^{\langle\sigma^*\rangle}_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$ *to output* $\perp$ *when it receives the signature* σ^* *as input, and otherwise to give the same response as* $\mathsf{OClaim}_{(vk_1,sk_1),\ldots,(vk_N,sk_N)}$*.*

Claimability requires three conditions, expressed by equations (5), (6), and (7) below. Informally, (5) requires that honest signers can successfully claim their signatures, (6) requires that adversarial parties cannot successfully claim a signature that they did not produce, and (7) requires that adversarial parties cannot produce a signature along with a claim that appears to be produced by an honest party (that is, falsely framing the honest party as the signer).[7]

Definition 17 (Claimability). *A ring signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* claimable *if equipped with algorithms* $(\mathsf{Claim}, \mathsf{VerClaim})$ *such that the following conditions hold.*

1. (Honest signer can claim) *There exists a negligible function* ε *such that for any* $N = \mathsf{poly}(k)$ *and* $(vk_1, sk_1), \ldots, (vk_N, sk_N) \leftarrow \mathsf{Gen}(1^k)$ *and any* $i \in [N]$*, it holds for any message* m *that*

$$\Pr\left[\sigma \leftarrow \mathsf{Sign}(R, sk_i, m) : \mathsf{VerClaim}(R, vk_i, \sigma, \mathsf{Claim}(R, sk_i, \sigma)) = 1\right] > 1 - \varepsilon(k), \quad (5)$$

where $R = \{vk_1, \ldots, vk_N\}$*.*

[7] Our definition does not guarantee that all signatures that verify (possibly a superset of all honestly generated signatures) can be claimed by someone; requiring this could be a reasonable alternative definition. See the full version [PS19] for more discussion.

2. (Non-signers cannot claim) *Let* $\mathcal{O} = \{\mathsf{OSign}\}$. *For any (possibly adversarial) PPT sampling-and-claiming algorithm* $\mathcal{A}_{\mathsf{Claim}} = (\mathcal{A}_1, \mathcal{A}_2)$, *there exists a negligible function* ε *such that*

$$\Pr\left[\begin{array}{l} (vk, sk) \leftarrow \mathsf{Gen}(1^k) \\ (R', m, \mathfrak{s}) \leftarrow \mathcal{A}_1^{\mathcal{O}, \mathsf{OClaim}_{(vk,sk)}}(vk) \\ \sigma \leftarrow \mathsf{Sign}(R' \cup \{vk\}, sk, m) \\ (\zeta, vk') \leftarrow \mathcal{A}_2^{\mathcal{O}, \mathsf{OClaim}_{(vk,sk)}}(R' \cup \{vk\}, \sigma, \mathfrak{s}) \\ b \leftarrow \mathsf{VerClaim}(R' \cup \{vk\}, vk', \sigma, \zeta) \\ b' \leftarrow \mathsf{Verify}(R' \cup \{vk\}, \sigma, m) \end{array} : \begin{array}{l} b = 1 \wedge b' = 1 \\ \wedge vk' \neq vk \end{array}\right] < \varepsilon(k). \tag{6}$$

3. (Malicious signer cannot frame an honest party) *For any PPT adversary* $\mathcal{A}_{\mathsf{S\&C}}$, *there exists a negligible function* ε *such that*

$$\Pr\left[\begin{array}{l} (vk, sk) \leftarrow \mathsf{Gen}(1^k) \\ (R', m, \sigma, \zeta) \leftarrow \mathcal{A}_{\mathsf{S\&C}}^{\mathcal{O}, \mathsf{OClaim}_{(vk,sk)}}(vk) \\ b \leftarrow \mathsf{VerClaim}(R' \cup \{vk\}, vk, \sigma, \zeta) \\ b' \leftarrow \mathsf{Verify}(R' \cup \{vk\}, \sigma, m) \end{array} : \begin{array}{l} b = 1 \wedge b' = 1 \\ \wedge Q \cap \{(\cdot, \sigma)\} = \varnothing \end{array}\right] < \varepsilon(k). \tag{7}$$

where $\mathcal{O} = \{\mathsf{OSign}\}$ *and* Q *is the set of queries made to oracle* $\mathsf{OClaim}_{vk,sk}$.

Anonymity and unforgeability of claimable ring signatures The definitions of anonymity and unforgeability must be adapted for claimable ring signature schemes, to allow the adversary a claim oracle as described next.

Definition 18 (Anonymity of claimable ring signatures). *A claimable ring signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify}, (\mathsf{Claim}, \mathsf{VerClaim}))$ *satisfies anonymity against adversarially chosen keys if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\{\mathsf{OSign}, \mathsf{OClaim}\}, \varnothing, 0)$-*anonymous (Definition 3). Moreover, the repudiable ring signature satisfies* adaptive *anonymity against adversarially chosen keys if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is*

$$(\{\mathsf{OSign}, \mathsf{OClaim}\}, \{\mathsf{OSign}, \mathsf{OClaim}^{\langle\sigma\rangle}\}, 0)\text{-}\mathit{anonymous}\ ,$$

where σ *is the challenge signature in the anonymity experiment (Equation (2)).*

Recall from Remark 4 that adaptive anonymity against adversarially chosen keys is the strongest anonymity notion compatible with claimability.

Definition 19 (Unforgeability of claimable ring signatures). *A claimable ring signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify}, (\mathsf{Claim}, \mathsf{VerClaim}))$ *is* unforgeable *if* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $\{\mathsf{OClaim}\}$-*unforgeable (Definition 7).*

3.4 Unclaimable ring signatures

An unclaimable ring signature scheme has the property that the signer cannot later convince anyone of her identity. That is, for any function that the true signer can compute given the signing randomness and the secret key, any other member of the ring can compute an indistinguishable function. The result is that even an adversary holding all ring members under duress cannot figure out who produced a given signature. This is true even if the ring members under duress attempt to cooperate with the adversary.

To achieve this, it suffices for any member of the ring to be able to extract signing randomness distributed indistinguishably from true signing randomness, that would produce the given signature under their secret key. More formally, the following guarantee should hold.

Definition 20 (Unclaimable ring signatures). *A* unclaimable *ring signature scheme is a ring signature scheme augmented with an additional algorithm* $\mathsf{ExtractRandomness}$ *as follows.*

- $\mathsf{ExtractRandomness}(R, sk, \sigma, m)$ *takes as input a ring* R*, a secret key* sk*, a signature* σ *and a message* m*. If* sk *is one of the secret keys for ring* R*, and* σ *is a signature on* m *with respect to* R*, then it outputs randomness* ρ*.*

$\mathsf{ExtractRandomness}$ *must satisfy the following condition.*

- (Statistical unclaimability) *Let* $\mathcal{R}$ *be the distribution of signing randomness. For any* $N = \mathsf{poly}(k)$ *there is a negligible function* ϵ *such that the following holds. Let* $(vk_1, sk_1), (vk_2, sk_2) \leftarrow \mathsf{Gen}(1^k)$*. For any message* m *and any* $vk_3, \ldots, vk_N$ *and* $sk_3, \ldots, sk_N$*, let* $R = \{vk_1, \ldots, vk_N\}$ *and* $S = \{(i, vk_i, sk_i)\}_{i \in [N]}$*. Let* $\rho \leftarrow \mathcal{R}$*,* $\sigma_1 \leftarrow \mathsf{Sign}(R, sk_1, m; \rho)$*, and* $\rho_1 \leftarrow \mathsf{ExtractRandomness}(R, sk_2, \sigma_1, m)$*. Let* $\rho_2 \leftarrow \mathcal{R}$ *and* $\sigma_2 \leftarrow \mathsf{Sign}(R, sk_2, m; \rho_2)$*. Then* $(S, \rho_1, \sigma_1) \approx_\epsilon (S, \rho_2, \sigma_2)$*.*

Definition 20 is unusual among the definitions in this paper, in that it gives a statistical rather than a computational guarantee. We opted to give the statistical definition because it is simpler, it is a stronger guarantee, and our construction in this case achieves the statistical guarantee. One could also consider a computational definition.

Remark 6 (Claimability is not the opposite of unclaimability). According to these definitions, unclaimability is *not* technically the opposite of claimability (even when ignoring the fact that the formal definitions give a statistical guarantee for unclaimability but a computational guarantee for claimability). Claimability requires the ability to "voluntarily claim" a signature *without remembering the signing randomness*, whereas unclaimability rules out the ability to "claim under duress" *even given the signing randomness.* For voluntary claims, the natural and stronger definition is to guarantee the ability to claim adaptive, without "planning ahead" and without the storage requirement of remembering the signing randomness. In contrast, when considering attempts to claim under duress, the natural and stronger definition is to rule out the possibility of successful claims even in the presence of the signing randomness.

Remark 7 (Unclaimability protects honest *signers).* An adversarial signer who *wants* to claim can devise ways of credibly later claiming a ring signature, even when using an unclaimable ring signature scheme.[8] This does not decrease the

[8] For example, an adversarial signer might use a PRG output as his signing randomness, or append it to his message, and remember the preimage. If he later revealed the preimage, it would likely serve as a credible claim to authorship of the signature.

utility of an unclaimable ring signaature scheme for honest signers who *want* their signatures to be unclaimable.

Unclaimability implies unrepudiability Any unclaimable ring signature scheme is also unrepudiable. Recall that the definition of unclaimability captures the idea that for any function that the true signer can compute given the signing randomness and the secret key, any other member of the ring can compute an indistinguishable function. Intuitively, the implication follows from the fact that repudiation would require a non-signer to behave in a way that *distinguishable* from any possible behavior of the actual signer.

Theorem 1. *Any unclaimable ring signature scheme is also unrepudiable.*

3.5 Repudiable-and-claimable ring signatures

Suppose that $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is a ring signature scheme, and there are algorithms $\mathsf{Repudiate}$, $\mathsf{VerRepud}$, Claim, and $\mathsf{VerClaim}$ such that, taken together with $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$, they form a repudiable ring signature scheme and a claimable ring signature scheme respectively. The seven algorithms together do *not* necessarily satisfy the natural notion of a "repudiable-and-claimable" scheme. This is not only syntactic: in certain cases, security might in fact not hold in the 7-algorithm scheme. The natural security definition for a repudiable-and-claimable ring signature scheme is to include both repudiation and claim oracles throughout the repudiability, claimability, anonymity, and unforgeability definitions. More discussion and formal definitions are given in the full version.

4 Repudiable construction

Due to space constraints, all proofs are deferred to the full version (attached in supplementary materials). We begin by defining the building blocks.

ZAPs are two-message public coin witness indistinguishable proofs [DN07].

Definition 21 (ZAP). *A ZAP for an NP language L with witness relation $\mathcal{R}_L$ is a triple of algorithms $\mathsf{ZAP}_L = (\mathsf{ZAP.Setup}_L, \mathsf{ZAP.Prove}_L, \mathsf{ZAP.Verify}_L)$, where $\mathsf{ZAP.Setup}$ and $\mathsf{ZAP.Prove}$ are PPT and $\mathsf{ZAP.Verify}$ is polynomial-time and deterministic, satisfying the following properties.*

Public coin. *For some polynomial $\ell = \ell(k)$, $\mathsf{ZAP.Setup}$ is the algorithm that on input 1^k, outputs a uniformly random element of $\{0,1\}^\ell$.*

Completeness. *For any $(x, w) \in \mathcal{R}_L$, $\rho \in \{0,1\}^{\ell(k)}$, we have $\Pr_{\pi \leftarrow \mathsf{ZAP.Prove}(\rho,x,w)}[\mathsf{ZAP.Verify}(\rho, \pi, x) = 1] = 1$.*

Adaptive soundness. *There exists a negligible function ϵ such that $\Pr_{\rho \leftarrow \mathsf{ZAP.Setup}(1^k)}[\exists(x, \pi) : x \notin L \wedge \mathsf{ZAP.Verify}(\rho, \pi, x)] \leq \epsilon(k)$.*

Witness indistinguishability. *For any sequences $\{\rho_k\}_{k\in\mathbb{N}}$, $\{x_k\}_{k\in\mathbb{N}}$, $\{w_{0,k}\}_{k\in\mathbb{N}}$, $\{w_{1,k}\}_{k\in\mathbb{N}}$, where for all k, $\rho_k \in \{0,1\}^{\ell(k)}$, $x_k \in L$ and $(x_k, w_{0,k}), (x_k, w_{1,k}) \in \mathcal{R}_L$, the following pair of ensembles is computationally indistinguishable: $\{\mathsf{ZAP.Prove}(\rho_k, x_k, w_{0,k})\}_{k\in\mathbb{N}} \overset{c}{\approx} \{\mathsf{ZAP.Prove}(\rho_k, x_k, w_{1,k})\}_{k\in\mathbb{N}}$.*

In this work, for simplicity, we will assume use of a ZAP for some NP-complete language L_{NP} (with witness relation $\mathcal{R}_{L_{\mathrm{NP}}}$) and for any $L \in \mathrm{NP}$ with witness relation $\mathcal{R}_L$, we define $\mathsf{ZAP.Prove}_L$ and $\mathsf{ZAP.Verify}_L$ as follows.

- $\mathsf{ZAP.Prove}_L$ takes as input a triple (ρ, x, w). If $(x, w) \notin \mathcal{R}_L$, then output $\perp$. Otherwise, use an NP reduction on (x, w) to get a pair $(x_{\mathrm{NP}}, w_{\mathrm{NP}}) \in \mathcal{R}_{L_{\mathrm{NP}}}$, and output $\mathsf{ZAP.Prove}(\rho, x, w)$.
- $\mathsf{ZAP.Verify}_L$ takes as input a triple (ρ, π, x), uses the same NP reduction to obtain x_{NP} (which is in L_{NP} iff $x \in L$), and outputs $\mathsf{ZAP.Verify}(\rho, \pi, x)$.

Next, we recall the definition of verifiable random functions (VRFs) [MRV99].

Definition 22 (VRF). *A* verifiable random function *(VRF) is a tuple of algorithms* $\mathsf{VRF} = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Prove}, \mathsf{VRF.Verify})$*, where* Gen *and* Verify *are PPT and* Eval *and* Prove *are polynomial time and deterministic, satisfying:*

Complete provability *With probability at least* $1 - 2^{-\Omega(k)}$ *over* $(pk, sk) \leftarrow \mathsf{VRF.Gen}(1^k)$*, we have for all inputs x that* $\Pr[\mathsf{VRF.Verify}(pk, x, \mathsf{VRF.Eval}(sk, x), \mathsf{VRF.Prove}(sk, x)) = 1] > 1 - 2^{-\Omega(k)}$.

Unique provability *For all* $pk, x, y_1, y_2, \tau_1, \tau_2$ *with* $y_1 \neq y_2$*, for either* $i = 1$ *or* $i = 2$ *it holds that* $\Pr[\mathsf{VRF.Verify}(pk, x, y_i, \tau_i) = 1] < 2^{-\Omega(k)}$.

Residual pseudorandomness *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a PPT adversary, where both* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *get oracle access to the VRF evaluation and prove algorithms. Let* $(pk, sk) \leftarrow \mathsf{VRF.Gen}(1^k)$*, and let* $(x, \mathfrak{s}) \leftarrow \mathcal{A}_1^{\mathsf{VRF.Eval}(sk,\cdot),\mathsf{VRF.Prove}(sk,\cdot)}(1^k, pk)$. *Let* $b \leftarrow \{0,1\}$*, and let v be either* $\mathsf{VRF.Eval}(sk, x)$ *or uniformly random, depending on the choice bit b. Let* $b' = \mathcal{A}_2^{\mathsf{VRF.Eval}(sk,\cdot),\mathsf{VRF.Prove}(sk,\cdot)}(1^k, v, \mathfrak{s})$. *Then there is a negligible function ϵ such that* $\Pr[b = b' \text{ and } x \notin Q] < 1/2 + \epsilon(k)$*, where Q is the set of oracle queries made by $\mathcal{A}$ to either oracle.*

For simplicity, we assume that Eval *takes inputs x of any length, i.e.,* $x \in \{0,1\}^*$.

Definition 23. *The* verification failure probability *of a VRF* VRF *is*

$$\Pr\left[\begin{array}{l}(pk, sk) \leftarrow \mathsf{VRF.Gen}(1^k)\\ b \leftarrow \mathsf{VRF.Verify}(pk, x, \mathsf{VRF.Eval}(sk, x), \mathsf{VRF.Prove}(sk, x))\end{array} : b = 0\right] .$$

The residual pseudorandomness property still holds even if the adversary queries many key pairs at once, and may adaptively learn some of the secret keys (then, residual pseudorandomness holds for the uncorrupted keys only).

Lemma 1 (Parallel VRF Game). *Let* VRF *be a a VRF. Then* $\forall$ *PPT* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and all* $N = \mathsf{poly}(k)$*, there is a negligible function ε such that*

$$\Pr\left[\begin{array}{l}(pk_1, sk_1), \ldots, (pk_N, sk_N) \leftarrow \mathsf{VRF.Gen}(1^k)\\ (m^*, \mathfrak{s}) \leftarrow \mathcal{A}_1^{\mathcal{V},\mathsf{Corr}}(vk_1, \ldots, vk_N)\\ \forall i \in [N], y_{i,0} \leftarrow \mathsf{VRF.Eval}(sk_i, m^*)\\ \forall i \in [N], y_{i,1} \leftarrow \$\\ b \leftarrow \{0,1\}\\ b' \leftarrow \mathcal{A}_2(\mathfrak{s}, (y_{i,b})_{i \in [N]\setminus C})\end{array} : b = b' \wedge \forall i \in [N] \setminus C, (i, m^*) \notin Q\right] < 1/2+\varepsilon(k) \,, \tag{8}$$

where oracle $\mathcal{V}$ *maps* (i,m) *to* $(y,\tau) = (\mathsf{VRF.Eval}(sk_i, m), \mathsf{VRF.Prove}(sk_i, m))$, *oracle* Corr *maps* i *to* sk_i, *and* C, Q *are the sets of queries to* Corr, $\mathcal{V}$ *respectively.*

4.1 Construction

Construction 1 *Our construction* R-RS *is parametrized by* ZAP, VRF, *and* M, *where:* ZAP *is a ZAP;* VRF *is a VRF with input domain* $\{0,1\}^*$, *whose* Verify *algorithm takes* ν *bits of randomness and whose verification failure probability (Definition 23) is* ε; *and* M *is a polynomial satisfying* $M \geq (\nu + k)/\log_2(1/\varepsilon)$.[9]

```
R-RS.Gen($1^k$)
  1. $(vk^1_{\mathsf{VRF}}, sk^1_{\mathsf{VRF}}), \ldots, (vk^4_{\mathsf{VRF}}, sk^4_{\mathsf{VRF}}) \leftarrow \mathsf{VRF.Gen}(1^k)$.
     Let $\boldsymbol{vk}_{\mathsf{VRF}} = (vk^1_{\mathsf{VRF}}, \ldots, vk^4_{\mathsf{VRF}})$ and $\boldsymbol{sk}_{\mathsf{VRF}} = (sk^1_{\mathsf{VRF}}, \ldots, sk^4_{\mathsf{VRF}})$.
  2. $\rho \leftarrow \mathsf{ZAP.Setup}(1^k)$.
  3. $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M) \leftarrow (\{0,1\}^\nu)^M$.
  4. Output $vk = (\boldsymbol{vk}_{\mathsf{VRF}}, \rho, \boldsymbol{\alpha})$ and $sk = (\boldsymbol{sk}_{\mathsf{VRF}}, vk)$.
```

Hereafter, we (implicitly) use the following convention to parse a ring R.

Write $R = \{vk_1, \ldots, vk_N\}$.

For each $i \in [N]$, *write*
$$vk_i = (\boldsymbol{vk}^i_{\mathsf{VRF}} = (vk^{i,1}_{\mathsf{VRF}}, \ldots, vk^{i,4}_{\mathsf{VRF}}), \rho_i, \boldsymbol{\alpha}_i = (\alpha^i_1, \ldots, \alpha^i_M)). \quad (9)$$

Definition 24. *Let* L *be the following NP language.*

$$\Big\{\big(R, m, \varphi, (y_1, y_2, y_3, y_4)\big) \;:\; \exists i^*, \tau_1, \tau_2, \tau_3, \tau_4, \gamma \text{ s.t. } (b_1 \vee b_2) \wedge (b_3 \vee b_4)$$
$$\text{where } \forall \eta \in \{1,2,3,4\}, b_\eta = \bigwedge_{i \in [N], j \in [M]} \mathsf{VRF.Verify}(vk^{i^*,\eta}_{\mathsf{VRF}}, (R, m, \varphi), y_\eta, \tau_\eta; \alpha^i_j \oplus \gamma)\Big\} .$$

We now present the Sign *and* Verify *algorithms of our construction.*

```
R-RS.Sign($R, sk, m$)
  1. Parse $R$ as described above and $sk = ((sk^1_{\mathsf{VRF}}, \ldots, sk^4_{\mathsf{VRF}}), vk)$.
  2. If $vk \notin R$ output $\perp$ and halt.
  3. Define $i^* \in [N]$ such that $vk_{i^*} = vk$.
  4. $\gamma \leftarrow \{0,1\}^\nu$. (This is used as part of the ZAP witness in Step 6.)
  5. $\varphi \leftarrow \{0,1\}^k$. (This is used as a salt for the VRF input in Step 7, and
     output in Step 8.)
  6. For $\eta \in \{1,2,3,4\}$, let $y_\eta = \mathsf{VRF.Eval}(sk^\eta_{\mathsf{VRF}}, (R, m, \varphi))$ and $\tau_\eta = \mathsf{VRF.Prove}(sk^\eta_{\mathsf{VRF}}, (R, m, \varphi))$.
     Let $\boldsymbol{y} = (y_1, \ldots, y_4)$.
  7. For each $i \in [N]$, let $\pi_i \leftarrow \mathsf{ZAP.Prove}_L(\rho_i, (R, m, \varphi, \boldsymbol{y}), (i^*, \tau_1, \perp, \tau_3, \perp, \gamma))$.
     Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$.
  8. Output $\sigma = (\boldsymbol{\pi}, \boldsymbol{y}, \varphi)$.
R-RS.Verify($R, \sigma, m$)
```

[9] As explained in the full version, a satisfactory value of M can be set even without knowledge of ε. If ε happens to be known, a smaller value of M can be chosen.

```
  1. Parse $R$ as above and $\sigma = ((\pi_1, \cdots, \pi_N), \boldsymbol{y}, \varphi)$.
  2. Output $\bigwedge_{i\in[N]} \mathsf{ZAP.Verify}_L(\rho_i, \pi_i, (R, m, \varphi, \boldsymbol{y}))$.
```

Next, we describe the repudiation algorithms for R-RS.

Definition 25. *Let L' be the following NP language:*

$$\Bigg\{ \big(R, m, \varphi, (y_1, \ldots, y_4), vk = (\boldsymbol{vk}_{\mathsf{VRF}}, \rho, \boldsymbol{\alpha})\big) \;:\; \exists i^*, y'_1, \ldots, y'_4, \tau'_1, \ldots, \tau'_4, \gamma \text{ s.t.}$$

$$((b'_1 \wedge b'_2) \vee (b'_3 \wedge b'_4)) \wedge vk = vk_{i^*}, \text{ where } \forall \eta \in \{1, 2, 3, 4\},$$

$$b'_\eta = \left(y'_\eta \neq y_\eta \wedge \bigwedge_{i\in[N], j\in[M]} \mathsf{VRF.Verify}(vk^{i^*,\eta}_{\mathsf{VRF}}, (R, m, \varphi), y'_\eta, \tau'_\eta; \alpha^i_j \oplus \gamma) \right) \Bigg\} .$$

```
R-RS.Repudiate($R, sk, \sigma$)
  1. Parse $R$ as above, $sk = ((sk^1_{\mathsf{VRF}}, \ldots, sk^4_{\mathsf{VRF}}), vk)$, and $\sigma = (\boldsymbol{\pi}, \boldsymbol{y}, \varphi)$.
  2. If $vk \notin R$ output $\perp$ and halt.
  3. Define $i^* \in [N]$ such that $vk_{i^*} = vk$.
  4. For $\eta \in \{1, 2\}$: let $y'_\eta = \mathsf{VRF.Eval}(sk^\eta_{\mathsf{VRF}}, (R, m, \varphi))$ and let $\tau'_\eta = \mathsf{VRF.Prove}(sk^\eta_{\mathsf{VRF}}, (R, m, \varphi))$.
  5. $\gamma \leftarrow \{0, 1\}^\nu$. (This is used as part of the ZAP witness in Step 6.)
  6. For each $i \in [N]$, let $\xi_i \leftarrow \mathsf{ZAP.Prove}_{L'}(\rho_i, (R, m, \varphi, \boldsymbol{y}, vk), (i^*, y'_1, y'_2, \perp, \perp, \tau'_1, \tau'_2, \perp, \perp, \gamma))$.
  7. Output $\xi = (\xi_1, \ldots, \xi_N)$.
R-RS.VerRepud($R, vk, \sigma, \xi$)
  1. Parse $R$ as above. If $vk \notin R$, output 1 and halt.
  2. Parse $\sigma = (\boldsymbol{\pi}, \boldsymbol{y}, \varphi)$, and $\xi = (\xi_1, \ldots, \xi_N)$.
  3. Output $\bigwedge_{i\in[N]} \mathsf{ZAP.Verify}_{L'}(\rho_i, \xi_i, (R, m, \varphi, \boldsymbol{y}, vk))$.
```

Remark 8. As written, the size of the VRF input (R, m, φ) scales with the size of R, and we have assumed that the VRF can take variable-length inputs. When this is not the case, or when a smaller-input VRF is desirable for efficiency reasons, the scheme can be straightforwardly modified using a collision-resistant hash function h, and evaluating the VRF on $h(R, m, \varphi)$.

Theorem 2. *Let* VRF *be a VRF and* ZAP *be a ZAP. Then* R-RS *is a repudiable ring signature scheme.*

Proofs are deferred to the full version ([PS19]).

5 Claimable transformation

In this section, we give a simple black-box transformation from any ring signature to a claimable ring signature scheme. If the original scheme is repudiable, the resulting scheme is moreover *claimable-and-repudiable.* We assume familiarity with the standard notions of commitments, standard signatures, and PRFs. We use standard syntax for these; the full version gives detailed syntax definitions.

5.1 The transformation

Construction 2 *Our transformation* $\mathsf{C\text{-}RS}$ *is parametrized by the following:* RS*, a ring signature scheme;* Σ*, a standard signature scheme;* Com*, a commitment scheme; and* PRF*, a PRF. For convenience, and w.l.o.g., we assume that the randomness of* Com *and* Σ *and the output of* $\mathsf{PRF.Eval}$ *are all in* $\{0,1\}^{\nu}$*.*

```
$\mathsf{C\text{-}RS.Gen}(1^k)$
    1. Let $(vk_{\mathsf{RS}}, sk_{\mathsf{RS}}) \leftarrow \mathsf{RS.Gen}(1^k)$.
    2. Let $(vk_{\Sigma}, sk_{\Sigma}) \leftarrow \Sigma.\mathsf{Gen}(1^k)$.
    3. Let $sk_{\mathsf{PRF}} \leftarrow \mathsf{PRF.Gen}(1^k)$.
    4. Output $vk = (vk_{\mathsf{RS}}, vk_{\Sigma})$ and $sk = (vk, sk_{\mathsf{RS}}, sk_{\Sigma}, sk_{\mathsf{PRF}})$.
```

Hereafter, we implicitly parse verification and signing keys of $\mathsf{C\text{-}RS}$ *as* $vk = (vk_{\mathsf{RS}}, vk_{\Sigma})$ *and* $sk = (vk, sk_{\mathsf{RS}}, sk_{\Sigma}, sk_{\mathsf{PRF}})$ *respectively. Also, for a ring* $R = \big(vk_1 = (vk^1_{\mathsf{RS}}, vk^1_{\Sigma}), \ldots, vk_N = (vk^N_{\mathsf{RS}}, vk^N_{\Sigma})\big)$ *, we write* $\mathsf{RS}(R)$ *to denote* $(vk^1_{\mathsf{RS}}, \ldots, vk^N_{\mathsf{RS}})$*.*

```
$\mathsf{C\text{-}RS.Sign}(R, sk, m)$
    1. Let $\sigma_{\mathsf{RS}} \leftarrow \mathsf{RS.Sign}(\mathsf{RS}(R), sk_{\mathsf{RS}}, m)$.
    2. Let $r_{\Sigma} = \mathsf{PRF.Eval}(sk_{\mathsf{PRF}}, (vk, \sigma_{\mathsf{RS}}, 0))$.
    3. Let $\sigma_{\Sigma} = \Sigma.\mathsf{Sign}(sk_{\Sigma}, (vk, \sigma_{\mathsf{RS}}); r_{\Sigma})$.
    4. Let $r_{\mathsf{Com}} = \mathsf{PRF.Eval}(sk_{\mathsf{PRF}}, (vk, \sigma_{\mathsf{RS}}, 1))$.
    5. Let $c = \mathsf{Com}((vk, \sigma_{\Sigma}); r_{\mathsf{Com}})$.
    6. Let $\sigma = (\sigma_{\mathsf{RS}}, c)$.
    7. If $\mathsf{C\text{-}RS.VerClaim}(R, vk, \sigma, \mathsf{C\text{-}RS.Claim}(R, sk, \sigma)) = 1$, output $\sigma$.
    8. Otherwise, output $(\perp, \perp)$.
$\mathsf{C\text{-}RS.Verify}(R, \sigma = (\sigma_{\mathsf{RS}}, c), m)$
    1. If $\sigma_{\mathsf{RS}} = \perp$, output $0$.
    2. Otherwise, output $\mathsf{RS.Verify}(\mathsf{RS}(R), \sigma_{\mathsf{RS}}, m)$.
$\mathsf{C\text{-}RS.Claim}(R, sk, \sigma = (\sigma_{\mathsf{RS}}, c))$
    1. Let $r'_{\Sigma} = \mathsf{PRF.Eval}(sk_{\mathsf{PRF}}, (vk, \sigma_{\mathsf{RS}}, 0))$.
    2. Let $r'_{\mathsf{Com}} = \mathsf{PRF.Eval}(sk_{\mathsf{PRF}}, (vk, \sigma_{\mathsf{RS}}, 1))$.
    3. Let $\sigma'_{\Sigma} = \Sigma.\mathsf{Sign}(sk_{\Sigma}, (vk, \sigma_{\mathsf{RS}}); r'_{\Sigma})$.
    4. If $c \neq \mathsf{Com}(\sigma'_{\Sigma}, r'_{\mathsf{Com}})$, output $\zeta = \perp$.
    5. Otherwise, output $\zeta = (r'_{\mathsf{Com}}, \sigma'_{\Sigma})$.
$\mathsf{C\text{-}RS.VerClaim}(R, vk, \sigma = (\sigma_{\mathsf{RS}}, c), \zeta = (r'_{\mathsf{Com}}, \sigma'_{\Sigma}))$
    1. Let $c' = \mathsf{Com}((vk, \sigma'_{\Sigma}); r'_{\mathsf{Com}})$.
    2. Output $(c = c') \wedge \Sigma.\mathsf{Verify}(vk_{\Sigma}, \sigma'_{\Sigma}, (vk, \sigma_{\mathsf{RS}}))$.
```

If RS *is a* repudiable *ring signature scheme then we additionally define* $\mathsf{C\text{-}RS.Repudiate}$ *and* $\mathsf{C\text{-}RS.VerRepud}$ *as follows.*

```
$\mathsf{C\text{-}RS.Repudiate}(R, sk, \sigma = (\sigma_{\mathsf{RS}}, c))$
    1. Output $\mathsf{RS.Repudiate}(\mathsf{RS}(R), sk, \sigma_{\mathsf{RS}})$.
$\mathsf{C\text{-}RS.VerRepud}(R, vk, \sigma = (\sigma_{\mathsf{RS}}, c), \xi)$
    1. Output $\mathsf{RS.VerRepud}(\mathsf{RS}(R), sk, \sigma_{\mathsf{RS}}, \xi)$.
```

Theorem 3. $\mathsf{C\text{-}RS}$ *is a claimable ring signature scheme. Moreover, if* RS *is a repudiable ring signature scheme, then* $\mathsf{C\text{-}RS}$ *is repudiable-and-claimable.*

6 Unclaimable construction

In this section we show how to construct unclaimable ring signatures from lattice assumptions. The scheme is exactly the SIS-based ring signature scheme of Brakerski and Kalai [BK10], augmented with an additional algorithm ExtractRandomness.

6.1 Lattice trapdoor sampling

We first give a very brief summary of necessary background on lattice trapdoors; see [GPV08] and the full version [PS19] for details Let $q \in \mathbb{N}$, $m' \in \mathbb{N}$, and $\beta \in \mathbb{Z}$ be functions of security parameter n. The (inhomogeneous, average-case) *short integer solution* ($\mathsf{SIS}_{q,m,\beta}$) assumption states that given $A \leftarrow \mathbb{Z}_q^{n \times m'}$, $v \leftarrow \mathbb{Z}_q^n$, it is computationally hard to find $x \in \mathbb{Z}_q^{m'}$ such that $Ax = v$ and $\|x\| \leq \beta$. For polynomial m', β and prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the SIS problem is known to be as hard as approximating worst-case lattice problems, in particular the Shortest Independent Vectors Problem (SIVP), to within a factor of $\beta \cdot \tilde{O}(\sqrt{n})$ [MR07,GPV08].

Let $D_{\Lambda,s,c}$ denote the discrete Gaussian distribution over n-dimensional lattice Λ, centered at $c \in \mathbb{R}^n$ and with parameter s. We note the existence of the following algorithms, described in [GPV08].

- There is an algorithm TrapdoorSamp that on input a security parameter 1^n produces a matrix $A \in \mathbb{Z}_q^n$ and a trapdoor T, where A is statistically close to uniform and T is a short basis for the lattice $\Lambda^\perp(A)$.
- There is an algorithm SampleDist sampling from the discrete Gaussian $D_{\mathbb{Z}^{m'},s,0}$.
- There is an algorithm SampleCond that on input a matrix A, trapdoor T, parameter s and vector u, produces a sample x distributed statistically close to the discrete Gaussian distribution $D_{\mathbb{Z}^{m'},s,0}$ conditioned on $Ax = u$. We have that $\|x\|_2 \leq s\sqrt{n}$ with probability 1.

We will also require additional algorithms that given output values of the algorithms SampleDist and SampleCond, respectively, sample randomness under which the algorithm produces the desired output.

- There is an algorithm ExplainDist that on input an image vector x and parameter s, samples from the distribution $\{\rho | \mathsf{SampleDist}(s;\rho) = x\}$.
- There is an algorithm ExplainCond that on input matrix A, trapdoor T, parameter s, vector u and image vector x, samples randomness ρ that yields output x under algorithm SampleCond with inputs (A, T, s, u), i.e. samples from the distribution $\{\rho | \mathsf{SampleCond}(A, T, s, u; \rho) = x\}$.

We describe the algorithms ExplainDist and ExplainCond in the full version. We will use a slight modification of the SampleCond algorithm of [GPV08] that uses the basis randomization technique of [CHKP10]. We need the following lemma.

Lemma 2. *Let (A_1, T_1) and (A_2, T_2) be sampled from* TrapdoorSamp*, let $y \in \mathbb{Z}_q^n$, and let $s \geq \max(\|\tilde{T_1}\|, \|\tilde{T_2}\|) \cdot \omega(\sqrt{\log n})$, where the tilde denotes Gram-Schmidt orthogonalization. Sample vectors x_1 and x_2' from* SampleDist*. Let $x_2 \leftarrow$* $\mathsf{SampleCond}(A_2, T_2, s, y - A_1 x_1)$*, and let $x_1' \leftarrow$* $\mathsf{SampleCond}(A_1, T_1, s, y - A_2 x_2')$*. Then $(A_1, T_1, A_2, T_2, x_1, x_2)$ and $(A_1, T_1, A_2, T_2, x_1', x_2')$ are statistically close.*

Intuitively, this lemma says that the two trapdoors induce the same distribution on sampled vectors. This follows immediately from Lemma 3.3 of [CHKP10]

6.2 The basic construction of [BK10]

We now describe the construction of [BK10], which first constructs a "basic" scheme, then augments it to fully secure ring signatures in a series of steps.

Let the message space be $\{0,1\}^\ell$, and let $X = \{x \in \mathbb{Z}_q^{m'} : \|x\|_2 \leq s\sqrt{m'}\}$ for some $s = \omega(\sqrt{n \log n \log q})$ be the set of "short" vectors.

The key generation algorithm samples a matrix with an SIS trapdoor, and an additional set of 2ℓ matrices, two corresponding to each bit of the message. It additionally samples a target vector y, and outputs the matrices and target vector as the verification key and the trapdoor as the signing key.

```
BK-RS.Gen$(1^k)$
    1. Let $(A, T) \leftarrow \mathsf{TrapdoorSamp}(1^k)$.
    2. For $(i, b) \in [\ell] \times \{0,1\}$, let $A_{i,b} \leftarrow \mathbb{Z}_q^{n \times m'}$.
    3. Let $y \leftarrow \mathbb{Z}_q^n$.
    4. Output $vk = (A, (A_{j,b})_{(j,b) \in [\ell] \times \{0,1\}}, y)$ and $sk = (vk, T)$.
```

The signing algorithm proceeds as follows. A target vector y is selected from the lexicographically first verification key. For each identity in the ring, short vectors are sampled for matrices corresponding to each bit of the message to be signed, as well as the additional matrix. Finally, the trapdoor is used to obtain a short vector sampled from the same distribution conditioned on Equation 10. The signature consists of the list of short vectors.

```
BK-RS.Sign$(R, sk, m; \rho)$
    1. Parse $R = (vk_1, \ldots, vk_N)$ and $sk = (vk, T)$.
    2. For $i \in [N]$, parse $vk_i = (A_i, (A_{j,b}^{(i)})_{(j,b) \in [\ell \times \{0,1\}}, y_i)$.
    3. Let $y = y_i$, where $i \in [N]$ is such that $vk_i$ is lexicographically first.
    4. If $vk \notin R$, output $\perp$ and halt.
    5. Define $i^* \in [N]$ be such that $vk_{i^*} = vk$.
    6. Using  trapdoor $T_A$ for $A_{i^*}$, we can sample $(x_j^{(i)})_{i \in [N], j \in \{0\} \cup [\ell]}$ such that
           $$\sum_{i \in [N]} A_i x_0^{(i)} + \sum_{\substack{i \in [N] \\ j \in [\ell]}} A_{j,m_j}^{(i)} x_j^{(i)} = y. \qquad (10)$$
       That is, for $(i, j) \in [N] \times \{0\} \cup [\ell]$ other than the pair $(i^*, 0)$, we invoke algorithm $\mathsf{SampleDist}$ to sample $x_j^{(i)} \in$ independently from the discrete Gaussian distribution $\mathcal{X}$. Finally, we invoke algorithm $\mathsf{SampleCond}$ use the trapdoor $T$ for $A_{i^*}$ to sample $x_0^{(i^*)}$ from a distribution statistically close to the distribution $\mathcal{X}$ conditioned on Equation 10 being satisfied.
    7. Output $\sigma = (x_j^{(i)})_{i \in [N], j \in \{0\} \cup [\ell]}$.
```

The verification procedure simply checks that each vector in the signature has short entries and that Equation 10 is satisfied.

```
BK-RS.Verify$(R, \sigma, m)$
  1. Parse $R = (vk_1, \ldots, vk_N)$.
  2. For $i \in [N]$, parse $vk_i = (A_i, (A^{(i)}_{j,b})_{(j,b)\in[\ell\times\{0,1\}}, y_i)$.
  3. Parse $\sigma = (x^{(i)}_j)_{i\in[N], j\in\{0\}\cup[\ell]}$.
  4. For each $x^{(i)}_j$ for $i \in [N], j \in \{0\}\cup[\ell]$, if $x^{(i)}_j \notin X$ then immediately reject.
  5. Let $y = y_i$, where $i \in [N]$ is such that $A_{i^*}$ is lexicographically first.
  6. Accept if Equation 10 above is satisfied, and otherwise reject.
```

We now augment the basic [BK10] ring signature scheme with additional algorithm ExtractRandomness that produces "explaining randomness." The algorithms ExplainDist and ExplainCond referenced below are described in the full version.

```
BK-RS.ExtractRandomness$(R, sk, \sigma, m)$
  1. Parse $R = (vk_1, \ldots, vk_N)$ and $sk = (vk, T)$.
  2. For $i \in [N]$, parse $vk_i = (A_i, (A^{(i)}_{j,b})_{(j,b)\in[\ell\times\{0,1\}}, y_i)$.
  3. Parse $\sigma = (x^{(i)}_j)_{i\in[N], j\in\{0\}\cup[\ell]}$.
  4. If $vk \notin R$, output $\perp$ and halt.
  5. Define $i^* \in [N]$ be such that $vk_{i^*} = vk$.
  6. For $(i, j) \in [N] \times \{0\} \cup [\ell]$ s.t. $(i, j) \neq (i^*, 0)$, run ExplainDist to sample
     randomness $\rho^{(i)}_j$ giving output $x^{(i)}_j$ from discrete Gaussian sampling.
  7. Run ExplainCond to sample random coins $\rho^{(i^*)}_0$ that produce output $x^{(i^*)}_0$
     under the conditional random sampling algorithm using trapdoor $T$.
  8. Output $(\rho^{(i)}_j)$.
```

Theorem 4. *Under the* $\mathsf{SIS}_{q,m',\beta}$ *assumption,* BK-RS *is a unclaimable ring signature scheme satisfying a weak notion of unforgeability in which the challenge is sampled at random at the beginning of the experiment.*

6.3 Unclaimability for the full ring signature scheme of [BK10]

The ring signature scheme above satisfies a weak notion of unforgeability, in which the forgery message is sampled at random by the challenger and sent to the forger in the beginning of the experiment. To achieve full unforgeability, [BK10] provide a sequence of four reductions to construct schemes satsifying successively stronger notions of unforgeability. We give a brief overview of these reductions and the corresponding modifications of the ExtractRandomness algorithm.

The first modified scheme appends a description of the ring to the message to be signed, so ExtractRandomness is simply invoked on a different message.

The second modification is the most complicated, and introduces a variant of chameleon hash functions. A chameleon hash function h is sampled during Gen and is included in the verification key vk. During Sign, randomness r is sampled from a certain distribution, and a value $y = h(m, r)$ is computed, where m is the message to be signed and h is the hash function corresponding to the lexicographically first identity in the ring. The previous signature scheme is invoked on $y = h(m, r)$, where m is the message and h is the hash function for the lexicographically first identity in the ring; then, r is appended to the resulting signature. Now the only randomness to explain is r and the previous signature scheme's randomness. So the only change to $\mathsf{ExtractRandomness}$ is that it now also gives random coins resulting in a particular r, which is straightforward.

The third modification simply computes a signature under the previous scheme of every prefix of the message, and outputs these $|m|$ signatures as its signature. The final modification has Gen additionally output a random pad α, and computes a signature on $m \oplus \alpha_1$ where α_1 is the pad for the lexicographically first identity in the ring. For each of these we simply invoke the previous $\mathsf{ExtractRandomness}$ algorithm on a different message. This yields the following.

Theorem 5. *Assuming* $\mathsf{SIS}_{q,m',\beta}$, *[BK10] ring signatures augmented with the above* $\mathsf{ExtractRandomness}$ *algorithm is an unclaimable ring signature scheme.*

Acknowledgements. We thank Yael Tauman Kalai for advice on an earlier draft, and anonymous reviewers for their comments. Both authors' research was supported by the following grants: NSF MACS (CNS-1413920), DARPA IBM (W911NF-15-C-0236), Simons Investigator award agreement dated June 5th, 2012, and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370. Sunoo Park was additionally supported by the MIT Media Lab's Digital Currency Initiative. Adam Sealfon was additionally supported by a DOE CSGF fellowship, DARPA/NJIT Palisade 491512803, Sloan/NJIT 996698, and MIT/IBM W1771646.

References

Ajt99. Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP*, 1999.

BCC$^+$15. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *ESORICS*, 2015.

Bit17. Nir Bitansky. Verifiable random functions from non-interactive witness-indistinguishable proofs. In *TCC*, 2017.

BK10. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. *IACR Cryptology ePrint Archive 2010/086*, 2010.

BKM09. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *J. Cryptology*, 22(1):114–138, 2009.

BLO18. Carsten Baum, Huang Lin, and Sabine Oechsner. Towards practical lattice-based one-time linkable ring signatures. Cryptology ePrint Archive 2018/107, 2018.

CDNO97. Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *CRYPTO*, 1997.

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, 2010.

CPP18. Ran Canetti, Sunoo Park, and Oxana Poburinnaya. Fully bideniable interactive encryption. *IACR Cryptology ePrint Archive*, 2018:1244, 2018.

CvH91. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, 1991.

DN07. Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6), 2007.

FS07. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *PKC*, 2007.

GHKW17. Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In *TCC*, 2017.

GO92. Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In *CRYPTO*, 1992.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

IEH$^+$16. Ai Ishida, Keita Emura, Goichiro Hanaoka, Yusuke Sakai, and Keisuke Tanaka. Group signature with deniability: How to disavow a signature. In *CANS*, 2016.

LNWX17. San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *ACNS*, 2017.

LSW06. Joseph K. Liu, Willy Susilo, and Duncan S. Wong. Ring signature with designated linkability. In *IWSEC*, 2006.

LWW04. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP*, 2004.

Man00. R. Mankiewicz. *The story of mathematics*. The story of mathematics. Princeton University Press, 2000.

MBB$^+$13. Carlos Aguilar Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit. Adapting lyubashevsky's signature schemes to the ring signature setting. In *AFRICACRYPT*, 2013.

Mon. Monero. Monero: Private digital currency. https://www.getmonero.org.

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

MRV99. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS*, 1999.

Ngu05. Lan Nguyen. Accumulators from bilinear pairings and applications. In *Topics in Cryptology - CT-RSA*, 2005.

PS19. Sunoo Park and Adam Sealfon. It wasn't me! repudiability and unclaimability of ring signatures. *IACR Cryptology ePrint Archive*, 2019:135, 2019.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, 2001.

SS10. Sven Schäge and Jörg Schwenk. A cdh-based ring signature scheme with short signatures and public keys. In *FC*, 2010.

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484, 2014.

XY04. Shouhuai Xu and Moti Yung. Accountable ring signatures: A smart card approach. In *CARDIS*, pages 271–286, 2004.