Logarithmic-Size (Linkable) Threshold Ring Signatures in the Plain Model*

Abida Haque[1], Stephan Krenn[2],
Daniel Slamanig[2], and Christoph Striecks[2]

[1] North Carolina State University, Raleigh, USA
ahaque3@ncsu.edu
[2] AIT Austrian Institute of Technology, Vienna, Austria
firstname.lastname@ait.ac.at

Abstract. A 1-out-of-N ring signature scheme, introduced by Rivest, Shamir, and Tauman-Kalai (ASIACRYPT '01), allows a signer to sign a message as part of a set of size N (the so-called "ring") which are anonymous to any verifier, including other members of the ring. Threshold ring (or "thring") signatures generalize ring signatures to t-out-of-N parties, with $t \geq 1$, who anonymously sign messages and show that they are distinct signers (Bresson et al., CRYPTO'02).

Until recently, there was no construction of ring signatures that both (i) had logarithmic signature size in N, and (ii) was secure in the plain model. The work of Backes et al. (EUROCRYPT'19) resolved both these issues. However, threshold ring signatures have their own particular problem: with a threshold $t \geq 1$, signers must often reveal their identities to the other signers as part of the signing process. This is an issue in situations where a ring member has something controversial to sign; he may feel uncomfortable requesting that other members join the threshold, as this reveals his identity.

Building on the Backes et al. template, in this work we present the first construction of a thring signature that is logarithmic-sized in N, in the plain model, and does not require signers to interact with each other to produce the thring signature.

We also present a linkable counterpart to our construction, which supports a fine-grained control of linkability. Moreover, our thring signatures can easily be adapted to achieve the recent notions of claimability and repudiability (Park and Sealfon, CRYPTO'19).

1 Introduction

Ring signatures, first introduced by Rivest, Shamir, and Tauman-Kalai [RST01], allow a member of a set (known as the *ring*) to anonymously sign on behalf of the ring. A verifier can check that a signature comes from one of the ring

*This is the full version of a paper which appears in the proceedings of the 25th IACR International Conference on Practice and Theory of Public Key Cryptography - PKC 2022, LNCS, Springer.

members, but cannot learn who the actual signer is, a property known as *(signer) anonymity*. Bresson, Stern, and Szydlo [BSS02] generalized ring signatures to t-out-of-N ring signatures (aka threshold ring signatures or *thring signatures*), in which $t \geq 1$ distinct members of an ad-hoc set participate to produce a signature.

In a (th)ring signature, the ring should be set-up free, i.e., members can join at will by publishing a public key. Anyone can then sign with respect to a ring assembled from available public keys. Despite this open setting, many ring signature schemes need a trusted setup and rely on heuristic assumptions (e.g,. the random oracle). However, the most desirable setting for an ad-hoc primitive like ring signatures is the plain model. In the plain model, security is based on standard and falsifiable hardness assumptions and no trusted setup is allowed. Also, most ring signature schemes are linear in the size of the ring, which is an issue when ring sizes are large. Recently, Backes et al. (BDH^+ for short) [BDH^+19b] presented an elegant construction of the first logarithmic-sized ring signatures in the plain model.

While the issues of model and signature size appear in ring signatures (cf. Section 1.3), thring signatures with $t > 1$ have another issue. In a thring signature, t-out-of-N signers compute a signature with the property that any verifier can check that t distinct parties signed the message without revealing exactly which t members signed. While the t signers are anonymous to anyone *outside their set*, these signers may need to interact to create the signature. Thus, signers are not necessarily anonymous *to each other*. Importantly, concatenating t instances of 1-out-of-N ring signatures does not guarantee distinct signers – the same signer may have signed t times in a row.

Two works avoid interaction among the signers but have other drawbacks. First, Okamoto et al. [OTYO18] designed a linear-sized scheme in the random oracle model. Here, ring members can create a 1-out-of-N ring signature themselves, while also showing that they are a new signer. Thus, a list of 1-out-of-N ring signatures forms a threshold ring signature. However, their solution requires a fully trusted party who issues short-term keys to all signers. This is a strong assumption for such an ad-hoc distributed primitive. Second, Liu, Wei, and Wong [LWW04] introduced linkable ring signatures, which allow a verifier to publicly check whether two signatures were produced by the same signer. This could be extended to produce threshold ring signatures. With a list of 1-out-of-N *linkable* ring signatures on a message, the signature verification algorithm checks pairwise that no two signatures in the list are linked to the same ring member without learning the identity of the signers. This approach is generic, but only yields a *one-time* thring signature scheme.

1.1 Our Contribution

In this work we construct thring signatures which are: *(i)* logarithmic-sized in the number of ring members; *(ii)* in the plain model from standard assumptions; *(iii)* and non-interactive, where specifically signers need not know each other.

In more detail:

- We present and prove the first construction of thring signatures where the signature size is *logarithmic* in the number of ring members and *in the plain model.* Our construction is instantiable from falsifiable standard assumptions without the need for the random oracle heuristic or trusted setup assumptions. Our construction is inspired by the recent results by BDH$^+$ [BDH$^+$19b] but requires novel ideas and techniques.
- We create a thring signature scheme in a setting where there is no interaction among the mutually anonymous signers. Signers need not know the other signers that participate in a threshold signing, meaning our scheme achieves *strong inter-signer anonymity.* Every signer locally computes a signature and the thring signature is just the collection of the individual signatures. Additionally, our thring signature scheme allows each signer to select their own threshold. We will discuss our solution in Section 1.2.

We also adapt the current model of *linkability* of ring signatures and make this model more flexible and fine-grained by using the concept of a *scope* to support *scoped linkability.* We describe scoped linkability in more detail in Section 5. We discuss a potential post-quantum instantiation and future directions in Section 6.

1.2 Overview of Our Techniques

To give context to our approach and techniques, we describe the approach used by BDH$^+$ [BDH$^+$19b], which is inspired by the construction of linear-size ring signatures in the plain model due to Bender, Katz, and Morselli [BKM09].

Outline of BDH$^+$ approach. In BDH$^+$, the ring is $P = (VK^1, \dots, VK^N)$. To join the ring, a user $s \in [N]$ generates key pairs $(vk_\sigma^s, sk_\sigma^s)$ and (pk^s, sk^s) of a signature scheme and a public-key encryption (PKE) scheme, respectively, and sets the verification and signing key to $VK^s := (vk_\sigma^s, pk^s)$ and $SK := (sk_\sigma^s, sk^s)$. To produce a signature for a message m with respect to R, a signer s computes a signature σ on m using sk_σ^s and encrypts σ under pk^s resulting in a ciphertext ct. The signer samples a random ciphertext ct' (representing another user i of the ring) and generates two hashing keys hk and hk' of a somewhere perfectly binding (SPB) hashing scheme [OPWW15] that are *perfectly binding* at position s and i respectively. It computes the hash of the ring R under both hk and hk', obtaining hash values h and h'. SPB hashing allows the signer to collapse a ring R of N verification keys into a ring of just two keys and membership witnesses are of size $\mathcal{O}(\log(N))$. Finally, signer s computes a perfectly sound NIWI proof π using an OR-statement which proves that either (hk, h) binds to a key VK^s and that ct encrypts a signature of m for VK^s or (hk', h') binds to a key VK^i and that ct' encrypts a signature of m for VK^i. A signature has the form $\Sigma = (ct, ct', hk, hk', \pi)$ and verification is straightforward.

For a non-interactive threshold variant, one needs to guarantee that a specific signer cannot contribute more than one signature to a thring signature, but at the same time keep other signatures from the same signer unlinkable. As BDH$^+$ encrypt the conventional signatures (which would identify the actual signer) for anonymity, one signer can sign repeatedly on the same message. While BDH$^+$

do have a linkable version, but as soon as a signer issues two signatures, even on different messages, they can be linked together, which contradicts the anonymity of thring signatures.

Outline of our approach. To achieve inter-signer anonymous thring signatures, we follow the BDH^+ template, but our approach requires novel ideas. First, instead of using a signature scheme, we use a verifiable random function (VRF) [MRV99], inspired by the recent work by Park and Sealfon [PS19].[3] A VRF is a function which outputs a pseudorandom value v and a proof p so that given the input m, the values (v, p) and the corresponding verification key vk everyone can check correctness of the evaluation. However, an output v is still pseudorandom if the proof p is not known. Because the VRF yields a deterministic value v, using v in the signature ensures distinctness of the signers. Meanwhile, we encrypt the proof p. Our approach now enables non-interactive thring signatures, where the signatures are a collection of single 1-out-of-N ring signatures. A verifier can inspect the VRF values for inequality to determine if the signers are distinct. We need an assumption called *key collision resistance* on the VRF, which requires that if the VRF is evaluated under different (honestly generated) verification keys and the same message, the evaluations will not collide. This is a reasonable assumption. Indeed, VRF candidates such as the Dodis-Yampolskiy VRF [DY05] satisfy this assumption (where key-collision can be seen to be unconditional).

Suppose the only change we make to BDH^+ is replacing a ciphertext with a VRF evaluation v and encrypted proof, and the other one with a random value $\tilde{v}$ in the VRF range with an encryption of a random value. Intuitively, anonymity holds as the values v and $\tilde{v}$ are (pseudo)random and do not leak the signer's identity. Meanwhile, unforgeability is based on the unpredictability of the VRF. However, we cannot use the same proof technique as BDH^+. In BDH^+, the proof of anonymity goes by hybrid game and indistinguishably hops between two OR-clauses to switch from an encrypted signature from user s to an encryption signature from user i. If we use this strategy in our hybrids, we end up at a point where the VRF evaluations of users s and i are at the same time present as values v and $\tilde{v}$ in one signature. Unfortunately, this immediately gives a distinguisher as the adversary can query signatures for the same message and ring both s and i and in the real game one "evaluation" in each signature will be a random string, but here it finds a pair that contains two values from queried signatures at the same time. Such an event is negligible in the real game.

Thus, our second change to change the NIWI to include a third OR clause. This third clause allows us in the anonymity proof to simulate the first two clauses of the OR language and switch the witnesses in the hybrids of the anonymity proof.

[3] We note that in a concurrent and independent work in [LW19], Lin and Wang propose a modification of BDH^+ that use VRFs instead of signatures to achieve repudiability. We note that their ideas do not extend to thring signatures and thus their approach cannot be directly compared to our work.

Being in the plain model precludes us from using a common reference string (CRS), which would allow us to embed a simulation trapdoor for the anonymity proof. To avoid a CRS, we use the following trick. Each signer s adds an extra secret key sk_{F}^{s} into her overall secret key and encrypts it, i.e., $E \leftarrow \mathsf{Enc}(pk^{s}, sk_{\mathsf{F}}^{s}; r)$, and the ciphertext is added to the public key VK^{s}. Our third clause in the OR language now proves that for two non-revealed users s and i (i.e., s and i from the first two OR-clauses) in the ring, it holds that $\mathsf{F}(sk_{\mathsf{F}}^{s}) = sk_{\mathsf{F}}^{i}$, where F is a one-way function (OWF)[4] , i.e., the clause shows that one of the two keys is the image of the other key under F. For honestly generated keys this relation will never be satisfied. However, in the simulation we can now set up user-keys in a way that they satisfy this relationship (without requiring a CRS). We can then use the witness for this clause of the OR proof to switch out the VRF witnesses to random.

Our approach to linkable thring signatures. With scoped linkability, one may control linking in a fine-grained way. An arbitrary string (the scope) used for signing allows one to link multiple signatures issued with respect to the same scope. While using the compiler by Liu et al. [LWW04] on the linkable version of the BDH^{+} scheme yields linkable thring signatures, it is not clear how to extend this to scoped linkability. One would need to fix the scopes beforehand and make the public keys linear in the number of used scopes. Thus, it would not be possible to support a potential unbounded number of scopes. Besides, the "tagging trick" in BDH^{+} makes their linkable version rather involved.[5] Our linkable thring signatures support an unbounded number of scopes and are a simple modular extension of our basic thring signatures.

We get scoped linkability by adding another VRF key pair to the user's keys and use the evaluation of the VRF on the scope for linking purposes (and fixing the scope in the scheme yields the conventional notion of linkability). We extend the language of the NIWI used for the OR proof to account for this additional VRF.

We use a variant of the folklore technique of extending the language of the proof system to obtain simulation-sound NIZKs [Sah01, Gro06], but use VRFs instead of PRFs or signatures. The additional VRF "signs" a verification key of a strongly unforgeable one-time signature scheme and the corresponding one-time signing key signs the respective partial signature. The signature and the verification key are attached to the respective 1-out-of-N ring signature.

Claimability and repudiability. Recently, Park and Sealfon in [PS19] introduced the notions of (un-)repudiability and (un-)claimability for ring signatures and are the first to formalize such definitions. Our constructions satisfy both notions of repudiability and claimability. Details are discussed in Appendix C.1.

[4] The restriction is that the domain and range of F is the same.

[5] The evaluation of their $\mathsf{JointVerify}$ algorithm which they need to prove with their NIWI, when unrolled gives 480 clauses, where each clause is a conjunction of 5 verification statements of a commitment scheme.

Flexibility. Okamoto et al. [OTYO18] introduced the notion of flexibility. Flexibility means ring members can sign a message themselves and add themselves to a previously computed ring signature if they wish to sign on the same message and ring. However, in [OTYO18] the new signers must cooperate with a trusted dealer to achieve this. The way we construct our threshold ring signatures also allows us to achieve flexibility in that new signers can add themselves to an already-created threshold ring signature at any time and thus the threshold t can be extended dynamically (cf. Appendix C.2.

Applications. We briefly describe some potential applications for non-interactive thring signatures with inter-signer anonymity and scoped linkability. One interesting practical application for thring signatures is to share cryptocurrency wallets that require no setup and that allow users to have a single key (even if they have multiple wallets) as discussed in [MHOY20].

Secondly, linkable ring signatures are a solution to e-voting [TW05]. Our scheme features scoped linkability, so signers can use the same verification key to vote for candidates in different offices. For example, votes cast under the scope 'mayor' are linkable, so that nobody can double vote for mayor. Meanwhile, votes cast for different scopes remain unlinkable, such as between scopes 'governor' and 'mayor'. As a result, thrings with scoped linkability might be a valuable tool for e-voting.

Finally, one can consider an extension of the whistleblower example from Rivest et al. [RST01] to the "parliament's problem". Suppose that a member of a national parliament (an MP) would like to submit a controversial bill for a law. The bill is controversial enough that the MP could lose his standing among his own party. However, if enough other members agree to the bill, it will be submitted for an official law. The MP cannot use a ring signature because another MP, wishing to attach their name, can neither add themselves nor submit a new ring signature while still showing that they are a distinct member. It would not be easy for this MP to discover other interested parties. Otherwise, a thring signature with interaction would do. The solution, then, is for the first MP to publish their bill using a thring signature with strong inter-signer anonymity. Now, he need not interact with other members, and any other MP can add themselves by contributing to the thring signature.

1.3 Related Work

Ring signatures. Many recent works in ring signatures focus on (1) obtaining efficient logarithmic-size ring signatures in the random oracle model in discrete-logarithm hard groups [GK15, BCC$^+$15, LPQ18], (2) sublinear-size post-quantum constructions in the (quantum) random oracle model from different assumptions, e.g., [LLNW16, DRS18, ESS$^+$19, ESLL19, DAZ19], (3) schemes from standard assumptions without random oracles (but a trusted setup) [BK10, DS19], and (4) constructions in the plain model. For the latter, Bender et al. [BKM09] present linear-sized signatures, but for a long time, improvements were made only when assuming a CRS (but without random oracles) towards signature size $\mathcal{O}(\sqrt{N})$ [CGS07] and $\Theta(\sqrt[3]{N})$ in [Gon19]. However, no progress in the plain model was made. Only recently Malavolta and Schröder [MS17] presented the first practical linear-size ring signatures in bilinear groups under knowledge-of-exponent assumptions. Backes et al. [BHKS18] improved the size to $\mathcal{O}(\sqrt{N})$ under falsifiable assumptions, culminating in the first logarithmic-size $\mathcal{O}(\log(N)\mathsf{poly}(\lambda))$ ring signatures in [BDH$^+$19b]. A recent work by Chatterjee et al. [CGH$^+$21] shows that by replacing the NIWI in the former approach with a particular ZAP allows a construction solely under the LWE assumption.

Threshold ring signatures. Currently, threshold ring signature schemes exist from general [AOS02, CCL07], classical [TWC$^+$04, LW05, YLA$^+$11, OTYO18], and post-quantum assumptions [DV09, CLRS10, MCGL11, BS13, PBB13, HS, BKP20]. However, all schemes so far are (1) at least linear in the size of the ring, (2) in the random oracle model (except for [YLA$^+$11] which uses a CRS), and (3) require interaction between the signers. The exception to the last issue is the approach by Okamoto et al. [OTYO18]. There, ring members can simply create a 1-out-of-N ring signature themselves, while also showing that they are a new signer. But we already have discussed their drawbacks.

Finally, Munch-Hansen et al. [MHOY20] construct threshold ring signatures of size $O(t)$ from a concrete traceable ring signatures with signature size $O(1)$. Like our approach, their construction is also non-interactive and achieves an anonymity notion akin to our *strong inter-signer anonymity*. But unlike our goal of constructing threshold ring signatures in the plain model, their approach requires NIZK arguments of knowledge. Moreover, their concrete instantiation additionally requires a trusted setup to generate the parameters of an RSA accumulator.

2 Preliminaries

We denote the main security parameter by λ. We write $[N] = \{1, \ldots, N\}$, and $\vec{a} = (a_1, \ldots, a_N)$. $\sqcup$ denotes disjoint union and $\triangle$ denotes symmetric difference. We denote algorithms by, e.g., A, and write $out \leftarrow \mathsf{A}(in)$ to denote that out is assigned the output of the probabilistic algorithm A with input in; Sometimes we make the used random coins r explicit and write $out \leftarrow \mathsf{A}(in; r)$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is negligible if $\forall k \in \mathbb{N}\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ :\ \mathsf{negl}(n) \leq n^{-k}$.

Non-Interactive Witness-Indistinguishable Proof Systems. Feige and Shamir [FS90] first introduced witness-indistinguishable proof systems. We recap the basic notions of non-interactive witness-indistinguishable proofs (NIWIs).

Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ be an *effective* relation, i.e., $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{R}$ are all efficiently computable. For $(x, \mathsf{w}) \in \mathcal{R}$, x is a *statement*, and w is the *witness*. The language $\mathcal{L}_{\mathcal{R}}$ is defined as all statements that have a valid witness in $\mathcal{R}$, i.e., $\mathcal{L}_{\mathcal{R}} := \{x \mid \exists \mathsf{w} : (x, \mathsf{w}) \in \mathcal{R}\}$.

Definition 1 (Non-interactive Proof System). *Let $\mathcal{R}$ be an effective relation and $\mathcal{L}_{\mathcal{R}}$ be the language accepted by $\mathcal{R}$. A* non-interactive proof system *for $\mathcal{L}_{\mathcal{R}}$ is a pair of algorithms* $(\mathsf{Prove}, \mathsf{Vfy})$ *where:*

- $\pi \leftarrow \mathsf{Prove}(1^\lambda, x, \mathsf{w})$. *On input a statement x and a witness* w*, outputs a proof π or $\perp$.*
- $b \leftarrow \mathsf{Vfy}(x, \pi)$. *Given a statement x and a proof π, outputs a bit b.*

NIWIs must satisfy the following three properties. First, *perfect completeness* guarantees that correct statements can always be successfully proven. Second, *perfect soundness* ensures that it is impossible to generate valid proofs for false statements. Finally, *witness indistinguishability* says that, given two valid witnesses for a statement, no efficient adversary can decide which witness was used to compute a proof. We refer to App. A.1 for formal definitions.

Following BDH$^+$ [BDH$^+$19b], we only consider NIWIs with bounded *proof-size*. That is, if we require that for any valid proof π generated by $\mathsf{Prove}(1^\lambda, x, \mathsf{w})$, it holds that $|\pi| \leq |C_x|\mathsf{poly}(\lambda)$ for a fixed polynomial $\mathsf{poly}(\cdot)$, where C_x is the verification circuit for the statement x, i.e., $(x, \mathsf{w}) \in \mathcal{R}$ iff $C_x(\mathsf{w}) = 1$.

Verifiable Random Functions. A verifiable random function (VRF) is a pseudo-random function that enables the owner of the secret key to compute a non-interactively verifiable proof for the correctness of its output [MRV99].

Definition 2 (Verifiable Random Function (VRF)). *A verifiable random function is 4-tuple* $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Prove}, \mathsf{Vfy})$ *where:*

- $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$. *On input the security parameter λ in unary, this PPT algorithm outputs a public verification key vk and corresponding secret key sk.*
- $v \leftarrow \mathsf{Eval}(sk, x)$. *On input the secret key sk and an input value $x \in \{0,1\}^{a(\lambda)}$, this deterministic algorithm outputs a value $v \in \{0,1\}^{b(\lambda)}$.*
- $p \leftarrow \mathsf{Prove}(sk, x)$. *On input the secret key sk and an input value x, this PPT algorithm outputs a proof p.*
- $b \leftarrow \mathsf{Vfy}(vk, x, v, p)$. *On input a verification key vk, an input value x, a value v, and a proof p, this deterministic algorithm outputs a single bit b.*

Here, $a(\lambda)$ and $b(\lambda)$ are polynomially bounded and efficiently computable functions in λ.

VRFs must satisfy the following six properties. First, *complete provability* guarantees that, if an output v and a proof p have been honestly computed on consistent inputs, then p will verify for v. Second, *unique provability* ensures that for all inputs x, a valid proof can only be computed for a unique output value v. Third, *residual pseudorandomness* says that no efficient adversary that sees arbitrarily many VRF evaluations can distinguish outputs on fresh inputs from uniform. Fourth, *residual unpredictability* requires that no efficient adversary that sees arbitrarily many VRF evaluations can compute a correct input and output pair; this is implied by residual pseudorandomness. Fifth, *key privacy* requires that no efficient adversary, only having access to an output but not the corresponding proof, can decide for which public key the output was computed. Finally, we introduce the notion of *key collision resistance* which guarantees that Eval, on input the same message but two different secret keys, will never return the same output value. We note that all required properties are for instance satisfied by the Dodis-Yampolskiy VRF [DY05]. Other instantiations in the standard model have been proposed by Lysyanskaya [Lys02] and Hofheinz and Jager [HJ16]. We refer to App. A.2 for formal definitions.

Somewhere Perfectly Binding Hashing. Somewhere statistically binding hashes were first introduced by Hubáček and Wichs [OPWW15]. Intuitively, such schemes allow one to efficiently commit to a vector (or database). Furthermore, one can generate short openings for individual positions of the vector.

Originally, it was only required that such schemes be *statistically binding* at a single position [OPWW15]. BDH$^+$ [BDH$^+$19b] strengthened this to *perfectly* binding. Furthermore, they introduced private openings to require a secret hashing key to compute a valid opening.

As shown in [OPWW15, BDH$^+$19b] SPB hashes with private local openings in the standard model can be efficiently obtained from any 2-message private information retrieval scheme with fully efficient verifier and perfect correctness. Also, we refer to [BDH$^+$19b] for DCR and DDH based instantiations of SPB based on [OPWW15].

Definition 3 (Somewhere Perfectly Binding (SPB) Hash). *A* somewhere perfectly binding hash with private local opening *is a tuple of algorithms* (Gen, Hash, Open, Vfy) *where:*

- $(hk, shk) \leftarrow \mathsf{Gen}(1^\lambda, n, \mathsf{i})$*. On input the security parameter λ in unary, a maximum database size n, and an index i, this PPT algorithm outputs public hashing key hk and corresponding secret hashing key shk.*
- $h \leftarrow \mathsf{Hash}(hk, db)$*. On input a hashing key hk and a database db of size n, this deterministic algorithm outputs a hash value h.*
- $\tau \leftarrow \mathsf{Open}(hk, shk, db, j)$*. On input a public and private hashing key hk and shk, a database db, and index j, this algorithm outputs witness τ.*
- $b \leftarrow \mathsf{Vfy}(hk, h, j, x, \tau)$*. On input a hash key hk, a hash value h, an index j, a value x and witness τ, this algorithm outputs a single bit b.*

SPBs must satisfy the following three properties. First, *correctness* guarantees that for honestly generated keys, hashes, and openings, verification will allows succeed. Second, *somewhere perfectly binding* ensures that if for a specific index i and value x verification succeeds, all valid openings on this position must open to x. Finally, *index hiding* says that no efficient adversary can infer the index i from the public hashing key.We refer to App. A.3 for formal definitions.

Definition 4 (Public Key Encryption). *A public key encryption scheme is a triple* ($\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$) *of algorithms over a message space $M(\lambda)$, ciphertext space $C(\lambda)$, and randomness space $Rnd(\lambda)$:*

- $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$. *On input the security parameter λ in unary, this PPT algorithm computes a public key pk and a corresponding secret key sk.*
- $ct \leftarrow \mathsf{Enc}(pk, m)$. *On input a public key pk and a message $m \in M(\lambda)$, this PPT algorithm outputs a ciphertext ct.*
- $m \leftarrow \mathsf{Dec}(sk, ct)$. *On input a secret key sk and a ciphertext ct, this deterministic algorithm outputs a message m.*

We require PKE schemes to satisfy the following three properties. First, *perfect correctness* guarantees that for honestly generated keys and ciphertexts, decryption will always yield the original plaintext. Second, *IND-CPA security* ensures that knowing only the public key, it is computationally infeasible to decide which message is contained in a ciphertext. Finally, *key privacy* says that no efficient adversary, not knowing the secret keys, can decide for which public key a ciphertext has been computed. We refer to App. A.4 for formal definitions.

Definition 5 (Strong One-Time Signature Scheme). *A strong one-time signature scheme is a triple* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ *of algorithms over a message space* $M(\lambda)$*:*

- $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$. *On input the security parameter λ in unary, this PPT algorithm computes a verification key vk and a corresponding signing key sk.*
- $\varsigma \leftarrow \mathsf{Sign}(sk, m)$. *On input a signing key sk and a message $m \in M(\lambda)$, this PPT algorithm outputs a signature ς.*
- $b \leftarrow \mathsf{Vfy}(vk, m, \varsigma)$. *On input a verification key vk, a message m and a signature ς, this deterministic algorithm outputs a single bit b.*

sOTS schemes must satisfy these two properties: First, *correctness* guarantees that for honestly generated keys and signatures, verification will always succeed. Second, *strong unforgeability* ensures that no efficient adversary that can obtain one signature for a given key can produce another valid signature on any message. We refer to App. A.5 for formal definitions.

Definition 6 (One-Way Function). *A one-way function* F *is defined such that:*

- $y \leftarrow \mathsf{F}(1^\lambda, x)$. *On input the security parameter λ in unary and an input value $x \in \{0,1\}^\lambda$, this deterministic algorithm computes an output $y \in \{0,1\}^\lambda$.*

OWFs must satisfy the following two properties. First, it must be *efficiently computable*, meaning that there is a polynomial-time algorithm to evaluate the function. Second, it must be *hard to invert*, so that given only an output value y, it is computationally infeasible to find a preimage x^* mapping to this output. Note that there can be multiple x^* for which $\mathsf{F}(x^*) = y$, but it is hard to find any such x^*. One additional requirement that we put on our OWFs is that the range must be a subset of the domain[6]. We refer to App. A.6 for a formal definition.

We define a concept of a *few fixed points* function. We have not seen this particular property in the literature, however a one-way function F is naturally

[6] A one-way permutation where the domain and range are equal can be used here.

a few fixed points function. If not, given $x \leftarrow \{0,1\}^\lambda$, it would be likely that $\mathsf{F}(x) = x$ and an adversary could find a pre-image. This property helps clarity in the unforgeability proof.

Definition 7 (Few fixed points). *A function F is a few fixed points function if $F : \{0,1\}^* \to \{0,1\}^*$ if $\Pr[x \leftarrow \{0,1\}^\lambda, F(x) = x] \leq \mathsf{negl}(\lambda)$.*

3 Framework and Security Definitions

3.1 Syntax

We extend the basic ring signature notation of Bender et al. [BKM09] to a thring signature. The notation is summarized in Table 1. Assuming an ordering of all public keys (e.g., lexicographic), we denote the sequence of all public keys as P as a *ring*. A *subring* is a subsequence $R \subseteq P$. Regardless of which members are part of the subring, we always enumerate the subring as $R = (VK^1, \ldots, VK^N)$. A set of signers is $S \subseteq [N]$, where $R[S] = \{VK^s\}_{s \in S}$. In a thring signature scheme, a set of signers $S \subseteq [N]$ signs a message $\mathsf{msg} \in \mathcal{M}$ with respect to a subring R. The secret keys of signers are denoted as T.

Each signer $s \in S$ chooses the minimum number of total signers t^s they require for a valid signature, and the verifier can choose a threshold as well. The sequence of all individual signer thresholds is denoted as $\vec{t}$. We denote the verification threshold by t_V, e.g., $t_V \leq |\{s : t^s \leq t_V\}|$ for $t^s \in \vec{t}$. The different thresholds are there to make the scheme as general as possible. This allows for different levels of signatures in different contexts (e.g., a signer may not want her signature to be used if there's not enough support, a verifier might be a potential signer who wants to see that there's enough support before adding her own signature to the set).

For generality, our syntax also considers system parameters pp generated by a Setup algorithm (which in our security definitions is always assumed to be honestly executed) allowing one to also model schemes requiring trusted setup in our framework. However, we stress that our instantiations given in Sec. 4 and Sec. 5 *do not require* such a Setup and are in the plain model.

Definition 8 (Threshold Ring Signature Scheme). *A threshold ring signature (thring) scheme is a 4-tuple of algorithms* $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy})$. *A subset of signers S from ring P signs the message* $\mathsf{msg} \in \mathcal{M}$ *with respect to a subring R and thresholds $\vec{t}$.*

- $pp \leftarrow \mathsf{Setup}(1^\lambda)$. *On input the security parameter λ in unary, this PPT algorithm generates public parameters pp. The public parameters are implicit input to all other algorithms and will be omitted when clear from context.*
- $(VK, SK) \leftarrow \mathsf{KGen}(pp)$. *On input the public parameters pp, this PPT algorithm generates a public verification key VK and a corresponding secret key SK for a signer.*

Symbol	Meaning
t^s	Individual threshold of signer s[§]
$\vec{t}$	$= (t^{i_1}, \dots, t^{i_{\lvert S \rvert}})$
t_V	Verification threshold
N	Number of members of the ring. Indexed by s.
P	Ordered list of public keys $P = (VK^1, \dots, VK^N)$.
R	Subring $R \subseteq P$.
S	Set of signers where $S \subseteq [N]$.
T	Secret keys to signers in S, $T = \{sk^s\}_{s \in S}$.
NS	Non-signers where $NS \subseteq [N]$
$\mathcal{M}$	Message space.
λ	Security parameter.

[§] By convention, indices used to distinguish between signers are written as superscripts.

Table 1. Notation used in the algorithms.

- $\sigma \leftarrow \mathsf{Sign}(\mathsf{msg}, T, R, \vec{t})$. *On input a message* msg*, a set of secret keys* T*, a subring* R*, and a vector of individual thresholds* $\vec{t}$*, this potentially interactive PPT procedure outputs a signature* σ *on* msg*.*
- $b \leftarrow \mathsf{Vfy}(\mathsf{msg}, R, \sigma, t)$. *On input a message* msg*, a subring* R*, a signature* σ*, and a verification threshold* t*, this deterministic algorithm outputs a bit* b*.*

3.2 Security Definitions

In this section, we define the security properties for a thring signature scheme: correctness, unforgeability with respect to insider corruption, and inter-signer anonymity with respect to adversarial keys. Our paper is the first feasibility result for non-interactive thrings entirely in the plain model. Because of these requirements, it does not seem easy to achieve the strongest notions of unforgeability and anonymity (as shown in [BKM09]). Namely, we avoid malicious users to satisfy unforgeability and we have anonymity for honest users only.

We first describe a set of oracles. In our security definitions, the adversary may access these oracles in arbitrary interleaf during the corresponding experiments. All oracles have access to the following initially empty sequences or sets: $P, P_{corr}, \mathcal{L}_{\text{signers}}$, and Q. The first sequence P is the ring, and $P_{corr} \subseteq P$ is the subset of corrupted (or malicious) members in the ring. The sequence $\mathcal{L}_{\text{signers}}$ is the triple of the signer, the public key, and the private key. The set Q is the set of signing queries.

- $\mathsf{OKGen}(s)$. On input a signer s, this oracle first checks whether there exists $(s, \cdot, \cdot) \in \mathcal{L}$ and returns $\perp$ if so. Otherwise, it generates a fresh key pair $(VK^s, SK^s) \leftarrow \mathsf{KGen}(pp)$, adds (s, VK^s, SK^s) to $\mathcal{L}_{\text{signers}}$, VK^s to P, and returns VK^s to the adversary.
- $\mathsf{OSign}(\mathsf{msg}, S, R, \vec{t})$. On input a message msg, a list of signers S, a subring R, and a vector of individual thresholds $\vec{t}$, this oracle first checks whether $R \subseteq P$ and returns $\perp$ if this is not the case. The oracle then decomposes S to $S = S_{corr} \sqcup S_{hon}$, where S_{corr} denotes corrupted users (i.e., corrupted or registered by $\mathcal{A}$) and S_{hon} denotes honest users. The oracle then engages in an execution of $\mathsf{Sign}(\mathsf{msg}, T, R, \vec{t})$. The oracle mimics the behavior of honest parties using

the secret keys corresponding to S_{hon}, and the adversary participates using S_{corr}. For all honest signers s, the oracle adds $(\mathsf{msg}, R, s, t^s)$ to Q.

- $\mathsf{OCorr}(s)$. On input a signer s, if there exists $(s, VK^s, sk^s) \in \mathcal{L}_{\text{signers}}$, the oracle returns SK^s to the adversary. The oracle adds VK^s to P_{corr}.
- $\mathsf{OReg}(s, VK^s)$. On input a signer s and a public key VK^s, the oracle checks if there exists $(s, \cdot, \cdot) \in \mathcal{L}_{\text{signers}}$ and returns $\perp$ if so. Otherwise, it adds VK^s to P_{corr} and $(s, VK^s, \cdot)$ to $\mathcal{L}_{\text{signers}}$.

Correctness. Correctness guarantees that a signature generated by sufficiently many honest users will always pass the verification algorithm. In our definition, the verification algorithm will check whether the individual thresholds are less than or equal to the verification threshold. This supports the concept of flexibility (cf. Appendix C.2).

Definition 9 (Correctness). *A thring signature scheme is* correct *if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for every* $\mathsf{msg} \in \mathcal{M}$*, any subring and ring such that $R \subseteq P$ (with $|P|$ being polynomially bounded in λ), any set of signers $S \subseteq R$, any vector of individual thresholds $\vec{t} = (t^1, \ldots, t^N)$, and any verification threshold t such that $t \leq |\{i : t^i \leq t\}|$, it holds that:*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda) \\ \{(VK^s, SK^s) \leftarrow \mathsf{KGen}(pp)\}_{s\in[|P|]} \\ T = \{SK^s\}_{s\in[|P|]} \\ \sigma \leftarrow \mathsf{Sign}(\mathsf{msg}, T, R, \vec{t}) \end{array} : \begin{array}{c} R[S] \subseteq P \implies \\ \mathsf{Vfy}(\mathsf{msg}, R, \sigma, t) = 0 \end{array}\right] = \mathsf{negl}(\lambda)$$

The scheme is called perfectly correct *iff* $\mathsf{negl}(\lambda) = 0$.

Unforgeability. Intuitively, unforgeability guarantees that an adversary who has corrupted up to $t - 1$ signers will not be able to generate a valid signature for threshold t. More precisely, the adversary can adaptively corrupt an arbitrary number of signers and engage in the signing protocol on arbitrary messages with honest users with respect to any thresholds and subrings. The adversary finally outputs a valid message, signature, subring, and threshold msg^*, σ^*, R^*, and t^*. The adversary wins if (1) he did not request OSign for too many honest parties on msg^* and R^* for thresholds less than t^*, and (2) he corrupted fewer than t^* members in R.

Note, we can tolerate corrupted but not *malicious* parties in our scheme. This is since we get inter-signer anonymity by having unique signatures. While this requirement is weaker, it is not unusual among the ring signature definitions (many schemes do not consider malicious parties). The experiment is described in Fig. 1.

Definition 10 (Unforgeability wrt Insider Corruption). *A thring signature scheme satisfies* unforgeability wrt insider corruption *if for all PPT adversaries $\mathcal{A}$ there is a negligible function* $\mathsf{negl}(\lambda)$ *such that* $\Pr[\mathsf{SigForge}^{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

```
**Experiment** $\mathsf{SigForge}^{\mathcal{A}}(\lambda)$
  $pp \leftarrow \mathsf{Setup}(1^\lambda)$
  $(\mathsf{msg}^*, \sigma^*, R^*, t^*) \leftarrow \mathcal{A}^{\mathsf{OKGen},\mathsf{OCorr},\mathsf{OSign}}(pp)$
  return 1 if:
    $\mathsf{Vfy}(\mathsf{msg}^*, \sigma^*, R^*, t^*) = 1$ and
    $R^* \subseteq P$ and
    $|U \cup (R^* \cap P_{corr})| < t^*$
    where $U = \{VK^s | \exists(\mathsf{msg}^*, R^*, s, t^s) \in Q : t^s \leq t^*\}$
  return 0
```

Fig. 1. Unforgeability

```
**Experiment** $\mathsf{Anonymity}^{\mathcal{A}}(\lambda)$
  $pp \leftarrow \mathsf{Setup}(1^\lambda)$
  $(\mathsf{st}, \mathsf{msg}^*, R^*, S_0^*, S_1^*, \vec{t}) \leftarrow \mathcal{A}^{\mathsf{OKGen},\mathsf{OCorr},\mathsf{OSign},\mathsf{OReg}}(pp)$
  $b \leftarrow \{0, 1\}$
  $T^* = \{SK^s\}$ for $s \in S_b^*$
    If any $s \in P_{corr}$ $\mathcal{A}$ will cooperate to create $\sigma_b$:
  $\sigma_b \leftarrow \mathsf{OSign}(\mathsf{msg}^*, T^*, R^*, \vec{t})$
  $b' \leftarrow \mathcal{A}^{\mathsf{OKGen},\mathsf{OCorr},\mathsf{OSign},\mathsf{OReg}}(\mathsf{st}, \sigma_b)$
    where OCorr and OSign ignore queries on $S_0^* \triangle S_1^*$ on $(\mathsf{msg}^*, R^*)$
  return a random bit if:
    $|S_0^*| \neq |S_1^*|$, or
    $S_0^* \cup S_1^* \not\subseteq R^*$, or
    $(S_0^* \cap P_{corr}) \neq (S_1^* \cap P_{corr})$, or
    $(\mathsf{msg}^*, R^*)$ has been signed before
  return $(b = b')$
```

Fig. 2. Inter-signer anonymity

Anonymity. Anonymity says that it is infeasible to infer from a valid signature which honest users contributed to the signature generation, or in general to link a signer across different signatures. We protect honest signers' identities even from other signers (*inter-signer anonymity*). We can tolerate malicious keys, even in the challenge sets, so long as both sets have the same malicious parties.

In the anonymity game, the adversary has access to all the oracles. He then requests a signature on the sets S_0^* or S_1^*. He may continue to make OSign and OCorr requests, but the oracle will not respond to queries in the set difference between S_0^* and S_1^*. The experiment is in Fig. 2.

In our scheme, users signing the same message msg with respect to the same subring R but potentially different thresholds are linkable among these signatures. We ensure the threshold by preventing signers from signing the same (msg, R) twice. Thus, in the challenge phase we require new message-ring pairs.

Moreover, a signer who has already signed with respect to a message/ring cannot sign again because the signature needs to be completely *distinct* for new users, which is not possible due to the deterministic part. Thus, if the adversary requests a signature from a specific user in the training phase, he could pinpoint from whom that signature originated. Thus, in the challenge phase we require either that the signature is different from previous signatures or that it verifies for a different message-ring pair.

In some ring signature schemes, an adversary $\mathcal{A}$ cannot identify which user a signature came from *even with knowledge of their secret key.* However, in our scheme, the signature is an output of the VRF, and $\mathcal{A}$ can learn which signer signed a message if he knows all the keys. Thus, we do not achieve this more robust definition.

We also note that the anonymity security definition does not hide the set sizes, but this is usually the case with threshold ring signatures. Unlike in other threshold ring signatures (where after the signature is created, it is not possible to *remove* signers), here the signature can be modified for a lower threshold by removing signatures from the total concatenation.

Due to how we use the VRF in our construction, we cannot achieve the strongest notion of anonymity from Bender et al. [BKM09] (i.e., anonymity against attribution attacks/full key exposure), where the adversary sees all the random coins for generating all the honest keys. However, we do achieve anonymity with respect to adversarially chosen keys [BKM09], which still allows an adversary to join the ring with maliciously generated keys and corrupt users. Although an adversary can de-anonymize corrupt or malicious users, our definition allows us to protect honest users' identities.

Definition 11 (Anonymity wrt adversarial Keys). *A threshold ring signature scheme satisfies* inter-signer anonymity with respect to adversarial keys *if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left| \Pr[\mathsf{Anonymity}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)\,.$$

4 Our Construction of Threshold Ring Signatures

In this section, we provide an overview of our construction. The formal description of our threshold ring signature scheme TRS is in Fig. 3.
Signing. Suppose that t members of a ring $R = (VK^1, \ldots, VK^N)$ wish to sign a message msg. We identify a signer index $s \in [N]$. Then each signer s locally evaluates the VRF using her private key on the inputs $\mathsf{msg}||R$ and $t^s||\mathsf{msg}||R$. The latter is needed because we allow each signer to choose its own threshold. The signer then encrypts the proofs of these VRF evaluations in ct and ct'. Next, it samples two SPB hashing keys hk^s and hk^i for $i \in [N]$ where $i \neq s$, binding at positions s and i respectively. Next, it calculates $h^i = \mathsf{Hash}_{SPB}(hk^i, R)$ and $h^s = \mathsf{Hash}_{SPB}(hk^s, R)$. Then (hk^s, h^s) and (hk^i, h^i) are commitments to VK^s and VK^i respectively. Finally, the signer computes a NIWI proof (discussed below). Signer s then outputs its signature σ^s as a tuple containing the VRF evaluations, the ciphertexts, hashing keys, the NIWI proof, and its individual threshold. A threshold signature is now a plain concatenation of individual signatures, i.e., $\sigma = (\sigma^1, \ldots, \sigma^t)$.
Verification. To verify a signature for a target threshold t, the verifier on $\sigma = (\sigma^1, \ldots, \sigma^t)$ checks each σ^i for each $1 \leq i \leq t$. It checks to see if the VRF value is different than all previously verified signatures. Then the verifier will check if the NIWI verifies and whether the threshold is less than or equal to his threshold t. The verifier will keep track of how many valid signatures it sees in a list $\mathcal{L}_V$. At the end, if $\mathcal{L}_V$ contains at least t signatures, the verifier will accept.

NIWI. The signer needs to show that one of the following claims is true:

(i) The computations are correct for signer s, i.e., hk^s is binding at position s and commits to R, VK^s and the corresponding secret key was used to evaluate the VRF on $\mathsf{msg}||R$ and $t^s||\mathsf{msg}||R$ resulting in v and v', and the corresponding proofs have been encrypted as ct and ct' under $pk_{\dagger}^{s}$. This is the branch for which an honest signer has all necessary keys; OR

(ii) the same computations have been performed correctly for signer i; OR

(iii) the secret keys sk_{F} of signer s and i satisfy $\mathsf{F}(sk^s_{\mathsf{F}}) = sk^i_{\mathsf{F}}$, that hk^s and hk^i have been computed for positions s and i, and that the sk_{F} are those corresponding to the public keys of s and i. As discussed in Sec. 1.2 this is needed in the anonymity proof (as publishing the VRF evaluations in the plain does no longer work with the technique of [BDH+19b]) but will never be satisfied for honest keys.

More formally, we denote this language as: $\mathcal{L}' := \mathcal{L}_{\mathcal{R}|_{VK}} \vee \mathcal{L}_{\mathcal{R}|_{VK'}} \vee \mathcal{L}_F$, where $\mathcal{R}|_{VK}$ indicates the following relation $\mathcal{R}$ for a specific key $VK^s = (vk^s, pk^s_{\dagger}, pk^s_{\ddagger}, E_s)$. Statements and witnesses have the form:

$$\begin{aligned}
\mathcal{R}|_{VK} : x &= (\mathsf{msg}, R, t^s, v, v', ct, ct', hk^s, h^s)\\
\mathsf{w} &= (VK^s, s, p, p', r_{ct}, r_{ct'}, \tau)\\
\mathcal{R}_F : x &= (R, h^s, h^i, hk^s, hk^i)\\
\mathsf{w} &= (s, i, VK^s, VK^i, \tau^s, \tau^i, sk^s_{\mathsf{F}}, sk^i_{\mathsf{F}}, r_{E_s}, r_{E_i})
\end{aligned}$$

The relations are then defined as follows:

$(x, \mathsf{w}) \in \mathcal{R}|_{VK}$ if and only if:

$$\begin{aligned}
&\mathsf{Vfy}_{SPB}(hk^s, h^s, s, VK^s, \tau) = 1 \wedge\\
&\mathsf{Enc}_{PKE}(pk^s_{\dagger}, p; r_{ct}) = ct \wedge\\
&\mathsf{Enc}_{PKE}(pk^s_{\dagger}, p'; r_{ct'}) = ct' \wedge\\
&\mathsf{Vfy}_{VRF}(vk^s, \mathsf{msg}||R, v, p) = 1 \wedge\\
&\mathsf{Vfy}_{VRF}(vk^s, t^s||\mathsf{msg}||R, v', p') = 1
\end{aligned}$$

$(x, \mathsf{w}) \in \mathcal{R}_F$ if and only if:

$$\begin{aligned}
&\mathsf{F}(sk^s_{\mathsf{F}}) = sk^i_{\mathsf{F}} \wedge\\
&\mathsf{Enc}_{PKE}(pk^s_{\ddagger}, sk^s_{\mathsf{F}}; r_{E_s}) = E_s \wedge\\
&\mathsf{Enc}_{PKE}(pk^i_{\ddagger}, sk^i_{\mathsf{F}}; r_{E_i}) = E_i \wedge\\
&\mathsf{Vfy}_{SPB}(hk^s, h^s, s, VK^s, \tau^s) = 1 \wedge\\
&\mathsf{Vfy}_{SPB}(hk^i, h^i, i, VK^i, \tau^i) = 1
\end{aligned}$$

Note that an honest signer does not use $(pk_{\ddagger}, sk_{\ddagger})$ for the NIWI proof. As mentioned above, our final language $\mathcal{L}'$ is the OR of two $\mathcal{L}_{\mathcal{R}|_{VK}}$ and $\mathcal{L}_F$. Statements and witnesses for $\mathcal{L}'$ are of the form:

$$x = \begin{pmatrix} \mathsf{msg} & R & v & v' & ct & ct' \\ t & h^s & h^i & hk^s & hk^i & \end{pmatrix} \qquad \mathsf{w} = \begin{pmatrix} VK^s & VK^i & s & i & \tau^s & \tau^i \\ p & p' & sk^s_{\mathsf{F}} & sk^i_{\mathsf{F}} & & \\ r_{ct} & r_{ct'} & r_{E_s} & r_{E_i} & & \end{pmatrix}$$

As our instantiation does not rely on any trusted setup, there is no need for Setup to generate joint parameters. The verifier must know a description of the ring to check the signature. The input of the VRF includes the description of the ring. Thus, it seems that the ring signature must always be linear in the size of the ring. However, if the verifier knows the ring beforehand, the signature need not include the ring description. Alternatively, we can change the domain of the VRF to compute on the hash of the ring instead (this was also noted by Park and Sealfon [PS19]). For simplicity, we include the ring as input everywhere in our scheme.

4.1 Security of Our Construction

In this section, we provide the formal proofs of each property correctness, unforgeability, and anonymity for the construction TRS.

```
Key Generation Gen($1^\lambda$):
    $(vk, sk) \leftarrow \mathsf{Gen}_{VRF}(1^\lambda)$;
    $(pk_\dagger, sk_\dagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
    $(pk_\ddagger, sk_\ddagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
    $sk_{\mathsf{F}} \leftarrow \{0,1\}^\lambda$;
    $r_E \leftarrow Rnd_{PKE}$;
    $E \leftarrow \mathsf{Enc}_{PKE}(pk_\ddagger, sk_{\mathsf{F}}; r_E)$;
    $VK := (vk, pk_\dagger, pk_\ddagger, E)$;
    $SK := (sk, sk_\dagger, sk_\ddagger, r_E, VK)$;
    return $(VK, SK)$.

Verification Vfy($\mathsf{msg}, R, \sigma, t_V$):
        // Parse each signature in the list;
    $\sigma = ((v, v', ct, ct', hk^s, hk^i, \pi, t^i))_{i=1}^t$;
    Sort list by $t^i$;
    for $i \in [t]$
        $h' := \mathsf{Hash}_{SPB}(hk^s, R)$;
        $h'' := \mathsf{Hash}_{SPB}(hk^i, R)$;
        $x := (\mathsf{msg}, R, v, v', ct, ct', h', h'', hk^s, hk^i)$;
        $b' \leftarrow \mathsf{Vfy}_{NIWI}(x, \pi)$;
        if   $b' = 1 \ \wedge \ \sigma^i.v \neq \sigma^k.v \ \forall k \in [i-1]$
          $\mathcal{L}_V.\mathsf{append}(t^i)$;
    endfor
    if $\exists i \geq t_V : \ \mathcal{L}_V[i] \leq i$ return 1;
    return 0.
```

```
Threshold Signing Sign($\mathsf{msg}, T, R, \vec{t}$):
    //  Note $T = \{SK^s\}$
    // Every signer $s$ signs by themselves
    // Look at algorithm per each signer $s$.
    $v \leftarrow \mathsf{Eval}_{VRF}(sk^s, \mathsf{msg}||R)$;
    $p \leftarrow \mathsf{Prove}_{VRF}(sk^s, \mathsf{msg}||R)$;
    $v' \leftarrow \mathsf{Eval}_{VRF}(sk^s, t^s||\mathsf{msg}||R)$;
    $p' \leftarrow \mathsf{Prove}_{VRF}(sk^s, t^s||\mathsf{msg}||R)$;
    $r_{ct}, r_{ct'} \leftarrow Rnd_{PKE}$;
    $ct \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p; r_{ct})$;
    $ct' \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p'; r_{ct'})$;
    $(hk^s, shk^s) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, s)$;
    $h^s \leftarrow \mathsf{Hash}_{SPB}(hk^s, R)$;
    $\tau^s \leftarrow \mathsf{Open}_{SPB}(hk^s, shk^s, R, s)$;
    // Pick other ring member $i \neq s$
    $i \in [N] \setminus s$;
    $r_{E_s}, r_{E_i} \leftarrow Rnd_{PKE}$;
    $(hk^i, shk^i) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, i)$;
    $h^i := \mathsf{Hash}_{SPB}(hk^i, R)$;
    $\tau^i \leftarrow \mathsf{Open}_{SPB}(hk^i, shk^i, R, i)$;
    // Call on the NIWI for language $\mathcal{L}'$
    $\pi \leftarrow \mathsf{Prove}_{NIWI}(x, \mathsf{w})$
    $\sigma^s := (v, v', ct, ct', hk^s, hk^i, \pi, t^s)$;
    // Every signer $s$ broadcasts the signature
    broadcast $\sigma^s$;
    // Final threshold ring signature under set $T$
    return $\sigma = \{\sigma^j\}_{j=1}^t$.
```

Fig. 3. Our threshold ring signature scheme. For notation refer to Table 1.

Theorem 1 (Correctness). *If the underlying NIWI, VRF, PKE, and SPB schemes are correct, and the VRF is key collision free,* TRS *is correct.*

Proof. By construction, all individual signatures are valid. Also, we require that $\sigma^i.v \neq \sigma^j.v$ for all $i \neq j$, which follows directly as otherwise we would have that $\mathsf{Eval}_{VRF}(sk^i, \mathsf{msg}||R) = \mathsf{Eval}_{VRF}(sk^j, \mathsf{msg}||R)$ contradicting to the assumed key collision freeness. □

Theorem 2 (Unforgeability). *If* F *is a one-way function,* VRF *has residual unpredictability and unique provability,* NIWI *has perfect soundness, SPB is somewhere perfectly binding and PKE is perfectly correct, then* TRS *is unforgeable.*

To prove unforgeability, we need to show that a forger $\mathcal{F}$ who knows up to $t-1$ secret keys cannot forge a signature that verifies for t signers. At a high level, because $\mathcal{F}$ needs to provide a valid NIWI proof in the signature, and the NIWI is perfectly sound, the claimed statement must indeed be true. We do not give $\mathcal{F}$ access to the OReg oracle so it is not possible for it to produce keys which satisfy $\mathcal{R}_{\mathsf{F}}$. $\mathcal{F}$ can, however, corrupt honest users. One possible winning strategy is for $\mathcal{F}$ to find a pair $sk_{\mathsf{F}}^{i}, sk_{\mathsf{F}}^{j}$ such that $\mathsf{F}(sk_{\mathsf{F}}^{i}) = sk_{\mathsf{F}}^{j}$. Because F is one-way, such a pair will exist with only negligible probability in the set of users (by Lemma 1, cf. Appendix A.6).

Then for the other strategy, as $\mathcal{F}$ needs to hold a witness to either $\mathcal{R}|_{VK}$ (or $\mathcal{R}|_{VK'}$) due to the somewhere perfect binding property of the SPB, the forgery must have used the identity of a signer who is a member of the ring. Due to the perfect correctness of the PKE scheme, ct (or ct') contains a valid proof for the respective VRF. Due to the unique provability of the VRF we know that an uncorrupted signer must have generated such a value. Finally, we can reduce unforgeability to the residual unpredictability of the VRF.

Proof. We prove unforgeability via hybrid arguments and a reduction to residual unforgeability.

$\mathcal{H}_0$ to $\mathcal{H}_1$: $\mathcal{H}_0$ is the unforgeability experiment $\mathsf{SigForge}$ from Figure 1. In $\mathcal{H}_1$, the challenger will pick one index i^* ahead of time. We abort on an OCorr request of i^*. When the adversary provides a forgery, he must have used honest keys (created via OKGen queries) in his chosen ring. Since i^* was picked at random, i^* is in the forgery's ring with probability at least $\frac{1}{q_{KG}}$ with q_{KG} the number of users generated by OKGen. Suppose there is an adversary $\mathcal{F}$ who forges in $\mathcal{H}_0$ with some probability. Then $\mathcal{F}$ can forge in $\mathcal{H}_1$ with the same probability (except with a loss of $\frac{1}{q_{KG}}$).

$\mathcal{H}_1$ to $\mathcal{H}_2$: All keys sk_{F} generated via OKGen are chosen in a way that for none of the pairs $(sk_{\mathsf{F}}^{j}, sk_{\mathsf{F}}^{k})$ it holds that either $sk_{\mathsf{F}}^{j} = \mathsf{F}(sk_{\mathsf{F}}^{k})$ or $sk_{\mathsf{F}}^{k} = \mathsf{F}(sk_{\mathsf{F}}^{j})$.

Such a pair existed in $\mathcal{H}_1$ with only negligible probability (Lemma 1), thus the probability of distinguishing between $\mathcal{H}_1$ and $\mathcal{H}_2$ is negligible as well. As the sk_{F}^{i} are chosen uniformly at random in the original game, here OKGen need only re-sample at random if a collision is found.

Now $\mathcal{R}_{\mathsf{F}}$ can never be satisfied among all the honest keys. In $\mathcal{H}_2$, the forgery $\sigma^* = \{(v, v', ct, ct', hk^s, hk^i, \pi, t^i)\}_{i=1}^{t}$ needs to use a witness for $\mathcal{R}|_{VK}$ or $\mathcal{R}|_{VK'}$. Due to symmetry of these both cases let us w.l.o.g. assume that $\mathcal{F}$ uses a witness for $\mathcal{R}|_{VK}$. Now by the perfect soundness of the NIWI we know that

$$(\mathsf{msg}, R, t, v, v', ct, ct', hk^{i_0}, h^{i_0}) \in \mathcal{L}|_{VK}.$$

As the SPB is somewhere perfectly binding, we have that $h^{i_0} = \mathsf{Hash}(hk^{i_0}, R)$ and $\mathsf{Vfy}_{SPB}(hk^{i_0}, h^{i_0}, i_0, VK^{s_0}, \tau^{i_0}) = 1$ implies that $R[i_0] = VK^{i_0}$. If we have $i_0 = i^*$, due to the perfect correctness of PKE we have that $(pk_{\dagger}^{i^*}, sk_{\dagger}^{i*})$ are correct for all messages. Then for $p := \mathsf{Dec}(sk_{\dagger}^{i*}, ct)$ and $p' := \mathsf{Dec}(sk_{\dagger}^{i*}, ct')$ the VRF verifications $\mathsf{Vfy}_{VRF}(vk^{i^*}, \mathsf{msg}^*||R^*, v, p) = 1$ and $\mathsf{Vfy}_{VRF}(vk^{i^*}, t^*||\mathsf{msg}^*||R^*, v', p') = 1$. Finally, due to the unique provability of the VRF we know that the values (v, p) and (v', p') are the unique pairs under vk^{i^*} corresponding to inputs $\mathsf{msg}^*||R^*$ and $t^*||\mathsf{msg}^*||R^*$.

Reduction to residual unpredictability.

We present a reduction $\mathcal{A}$ to the residual unpredictability of the VRF, which uses $\mathcal{F}$ (as in $\mathcal{H}_2$) as a subroutine.

- $\mathcal{A}$ from challenger $\mathcal{C}_{VRF}$ receives vk which we embed into VK^{i^*}.
- On each request from $\mathcal{F}$: OKGen, OCorr, OSign are as in $\mathcal{H}_2$. But for each VRF evaluation at i^* $\mathcal{A}$ queries $\mathcal{C}_{VRF}^{\mathsf{OEval}(sk,\cdot)}$. Remember that if $\mathcal{F}$ requests OCorr on either i^* then ABORT.
- From a valid forgery σ^* of $\mathcal{F}$, we obtain p and p' for inputs $\mathsf{msg}^*||R^*$ and $t^*||\mathsf{msg}^*||R^*$.
- Output one of $(\mathsf{msg}^*||R^*, v)$ and $(t^*||\mathsf{msg}^*||R^*, v')$ as forgery to $\mathcal{C}_{VRF}$.

If $\mathcal{F}$ made a valid forgery, then with probability $\frac{1}{q_{KG}}$ he picked the index i^*. Because the relationship for F does not hold, the NIWI has perfect soudness, the SPB is somewhere perfectly binding, and the PKE is perfectly correct, $\mathcal{F}$ can make a valid forgery by violating the residual unpredictability of the VRF. Consequently, we can just forward all VRF evaluations for this key in calls to the OSign oracle to the challenger of the VRF and given that the winning condition for the forgery output by the thrings forger are valid, we need to have a fresh evaluation of the VRF, which allows $\mathcal{A}$ to break residual unpredictability. As $\mathcal{A}$ can break residual unpredictability with at most negligible probability, we see that $\mathcal{F}$ can win in $\mathcal{H}_2$ with at most negligible probability as well. By hybrid argument, we see that $\mathcal{F}$ cannot win in $\mathcal{H}_0$ either except with negligible probability.

□

Theorem 3 (Anonymity). *If SPB is index hiding, PKE has key-privacy and CPA-security,* NIWI *is computationally witness-indistinguishable, and* VRF *has residual pseudorandomness and key-privacy then* TRS *is anonymous.*

Recall the anonymity experiment in Figure 2. In the training phase, the adversary $\mathcal{A}_{anon}$ queries on OKGen, OSign, OCorr, and OReg. Then in the challenge phase, $\mathcal{A}_{anon}$ submits a message msg, a subring R, and two signing sets $S_0, S_1 \subset R$. The challenger picks one of the signing sets S_b and computes a signature σ. On σ, $\mathcal{A}_{anon}$ guesses which of S_0, S_1 signed the message.

For us, a t-out-of-N thring signature is a collection of t ring signatures, and signatures are independent of each other. Thus, it suffices to show anonymity for a single signer and one can use a hybrid argument to show anonymity for larger thresholds. The probability of distinguishing between two sets of signatures is negligible if distinguishing between two signatures is negligible. Then for the challenge phase, $\mathcal{A}_{anon}$ will produce two indices s_0, s_1, message msg, and ring R.

Over a sequence of hybrids, we transform the signature element by element from one under s_0 to a signature under s_1. By showing that each hybrid is computationally indistinguishable from its predecessor, we see that signatures under s_0 and s_1 are indistinguishable to $\mathcal{A}_{anon}$.

We make changes over the hybrids and justify them in the proofs by using the following properties: *(i)* Changes to hk: the SPB is index-hiding. *(ii)* Changes to ct: the PKE has key-privacy and CPA-security. *(iii)* Changes to the witness used for π: the NIWI is computationally witness-indistinguishable. *(iv)* Changes to the value v: the VRF has residual pseudorandomness.

Proof. Consider the following hybrids:

$\mathcal{H}_0$ to $\mathcal{H}_1$: $\mathcal{H}_0$ is the anonymity experiment in Figure 2 with challenge bit $b = 0$. The challenger knows ahead of time q_{KG}, the number of queries $\mathcal{A}_{anon}$ will make to OKGen and picks two indices $\mathsf{i}_0, \mathsf{i}_1 \leftarrow [q_{KG}]$ ($\mathsf{i}_0 \neq \mathsf{i}_1$). If on either $\mathsf{i}_0, \mathsf{i}_1$, $\mathcal{A}_{anon}$ requests OCorr (or chooses these for OReg) then ABORT. Finally, we require that $\mathcal{A}_{anon}$ picks indices $\mathsf{i}_0, \mathsf{i}_1$ equal to the two indices the challenger picked ahead of time. Because $\mathsf{i}_0, \mathsf{i}_1$ were picked randomly, with $\frac{1}{q_{KG}}$ probability these will be the right two indices. An adversary playing in $\mathcal{H}_1$ wins with the same probability as in $\mathcal{H}_0$, except for a multiplicative loss of $\frac{1}{(q_{KG})^2}$.

$\mathcal{H}_1$ to $\mathcal{H}_2$: In this step, the challenger always chooses i_1 as the 'other index' when computing the final challenge signature. As i_1 was uniformly random, this is indistinguishable.

$\mathcal{H}_2$ to $\mathcal{H}_3$: In this step, for OKGen on $\mathsf{i}_0, \mathsf{i}_1$ make sure the secret keys $sk_{\mathsf{F}}^{\mathsf{i}_0}$ and $sk_{\mathsf{F}}^{\mathsf{i}_1}$ are such that $\mathsf{F}(sk_{\mathsf{F}}^{\mathsf{i}_0}) = sk_{\mathsf{F}}^{\mathsf{i}_1}$ holds. This change affects only $sk_{\mathsf{F}}^{\mathsf{i}_0}$ and $sk_{\mathsf{F}}^{\mathsf{i}_1}$, which are hidden in E_{i_0} and E_{i_1} and are never revealed.

$\mathcal{H}_3$ to $\mathcal{H}_4$: Calculate $(v^1, p^1) \leftarrow (\mathsf{Eval}_{VRF}(sk^{\mathsf{i}_1}, \mathsf{msg}||R), \mathsf{Prove}_{VRF}(sk^{\mathsf{i}_1}, \mathsf{msg}||R))$ and $(v'^1, p'^1) \leftarrow (\mathsf{Eval}_{VRF}(sk^{\mathsf{i}_1}, t||\mathsf{msg}||R), \mathsf{Prove}_{VRF}(sk^{\mathsf{i}_1}, t||\mathsf{msg}||R))$,
and $\tau^{\mathsf{i}_1} = \mathsf{Open}_{SPB}(hk^{\mathsf{i}_1}, shk^{\mathsf{i}_1}, R, \mathsf{i}_1)$. Then change the witness w:

$$\mathcal{H}_3 \quad \mathsf{w}^0 = (VK^{\mathsf{i}_0}, VK^{\mathsf{i}_1}, \mathsf{i}_0, \mathsf{i}_1, \tau^{\mathsf{i}_0}, \tau', p, p', sk_{\mathsf{F}}^{\mathsf{i}_0}, sk'_{\mathsf{F}}, r_{ct}, r_{ct'}, r_{E_0}, r_{E_1})$$

$$\mathcal{H}_4 \; \widehat{\mathsf{w}}^0 = (VK^{\mathsf{i}_0}, VK^{\mathsf{i}_1}, \mathsf{i}_0, \mathsf{i}_1, \tau^{\mathsf{i}_0}, \tau^{\mathsf{i}_1}, p^1, p'^1, sk_{\mathsf{F}}^{\mathsf{i}_0}, sk_{\mathsf{F}}^{\mathsf{i}_1}, r_{ct}, r_{ct'}, r_{E_0}, r_{E_1})$$

Note that this only makes changes in the *witness* of the NIWI. The values in the signature are only changed in subsequent games. Since the NIWI is witness indistinguishable, these changes are indistinguishable to any adversary. We construct $\mathcal{A}_{WI}$ which uses $\mathcal{A}_{anon}$.

1. $\mathcal{A}_{WI}$ activates $\mathcal{A}_{anon}$. He chooses $\mathsf{i}_0, \mathsf{i}_1$.
2. For each query, he answers as the challenger would.
3. On a challenge $(s_0, s_1, \mathsf{msg}, R)$, if $s_0 = \mathsf{i}_0$ and $s_1 = \mathsf{i}_1$, he calculates $\mathsf{w}^0, \widehat{\mathsf{w}}^0$ as above and sends to the challenger. He gets back π^*.
4. $\mathcal{A}_{WI}$ forwards π^* as part of the signature to $\mathcal{A}_{anon}$.
5. $\mathcal{A}_{WI}$ outputs the same as $\mathcal{A}_{anon}$.

In $\mathcal{H}_3$, the witness is w_0, and in $\mathcal{H}_4$, it is $\widehat{\mathsf{w}}^0$. Then if $\mathcal{A}_{anon}$ wins $\mathcal{H}_3$ and $\mathcal{H}_4$ with different probabilities, then $\mathcal{A}_{WI}$ can win the witness-indistinguishability game with the same probability. Thus, $\mathcal{H}_3$ and $\mathcal{H}_4$ are indistinguishable.

$\mathcal{H}_4$ to $\mathcal{H}_5$: $ct := \mathsf{Enc}_{PKE}(pk_{\dagger}^{\mathsf{i}_0}, p; r_{ct}) \rightarrow ct^1 := \mathsf{Enc}_{PKE}(pk_{\dagger}^{\mathsf{i}_1}, p; r_{ct})$

To show that this change is indistinguishable, we construct an adversary to PKE key privacy $\mathcal{A}_{KP}^{PKE}$ following the experiment in Figure 14.

1. $\mathcal{A}_{KP}^{PKE}$ receives two public keys pk^0, pk^1 from his challenger.
2. $\mathcal{A}_{KP}^{PKE}$ activates $\mathcal{A}_{anon}$. He picks $\mathsf{i}_0, \mathsf{i}_1$. $\mathcal{A}_{KP}^{PKE}$ answers every query as the challenger would have done, except for KGen at i_0 and i_1, where he gives $pk^{\mathsf{i}_0} = pk^0$ and $pk^{\mathsf{i}_1} = pk^1$.
3. Finally, $\mathcal{A}_{anon}$ will request a signature on msg, R.
4. $\mathcal{A}_{KP}^{PKE}$ computes using sk^{i_0}, $p \leftarrow \mathsf{Prove}_{NIWI}(sk^{\mathsf{i}_0}, \mathsf{msg}||R)$. He sends p to his challenger.
5. The challenger will pick $b \leftarrow \{0,1\}$. If $b = 0$, returns $ct^* = ct$ and if $b = 1$, returns $ct^* = ct^1$.
6. $\mathcal{A}_{KP}^{PKE}$ uses ct^* for the signature he gives $\mathcal{A}_{anon}$.
7. Output the same as $\mathcal{A}_{anon}$.

If the challenger picks pk^0, this is the anonymity game as in $\mathcal{H}_4$, but if he picks pk^1 then this is the game as in $\mathcal{H}_5$. Thus, if $\mathcal{A}_{anon}$ wins $\mathcal{H}_4$ and $\mathcal{H}_5$ with different probabilities, then this is the advantage of $\mathcal{A}_{KP}^{PKE}$ winning the PKE key privacy experiment.

$\mathcal{H}_5$ **to** $\mathcal{H}_6$**:** $ct' := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_0}, p'; r_{ct'}) \rightarrow ct'^1 := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_1}, p'; r_{ct'})$

The argument is identical to the transition from $\mathcal{H}_4$ to $\mathcal{H}_5$ with a reduction to PKE key privacy. The only difference is that $p' \leftarrow \mathsf{Prove}_{VRF}(sk^{\mathsf{i}_0}, t||\mathsf{msg}||R)$ and thus we omit details.

$\mathcal{H}_6$ **to** $\mathcal{H}_7$**:** $ct^1 := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_1}, p; r_{ct}) \rightarrow \widehat{ct}^1 := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_1}, p^1; r_{ct})$, where $p^1 \leftarrow \mathsf{Prove}_{VRF}(sk^{\mathsf{i}_1}, \mathsf{msg}||R)$.

We construct $\mathcal{A}_{CPA}$ which uses $\mathcal{A}_{anon}$ as a subroutine to break CPA securityfollowing Figure 13.

1. $\mathcal{A}_{CPA}$ receives pk. $\mathcal{A}_{CPA}$ picks $\mathsf{i}_0, \mathsf{i}_1$, activates $\mathcal{A}_{anon}$.
2. For each query by $\mathcal{A}_{anon}$, $\mathcal{A}_{CPA}$ answers as a challenger would, except for OKGen at i_0, where he gives $pk_\dagger^{\mathsf{i}_0} = pk$.
3. When $\mathcal{A}_{anon}$ queries on $(s_0, s_1, \mathsf{msg}, R)$ then $\mathcal{A}_{CPA}$ calculates both $p \leftarrow \mathsf{Eval}_{VRF}(sk^{\mathsf{i}_0}, \mathsf{msg}||R)$ and $p^1 \leftarrow \mathsf{Eval}_{VRF}(sk^{\mathsf{i}_1}, \mathsf{msg}||R)$. He gives p, p^1 to his challenger.
4. Challenger flips $b \leftarrow \{0,1\}$. If $b = 0$ he encrypts p, if $b = 1$ he encrypts p^1. He returns ct^* to $\mathcal{A}_{CPA}$.
5. $\mathcal{A}_{CPA}$ uses ct^* for the signature he gives to $\mathcal{A}_{anon}$.
6. $\mathcal{A}_{CPA}$ outputs the same as $\mathcal{A}_{anon}$.

If the challenger picks p, we are in $\mathcal{H}_6$, if p^1 then $\mathcal{H}_7$. Thus, $\mathcal{A}_{CPA}$ wins the CPA-security experiment with the same advantage as the difference of $\mathcal{A}_{anon}$ winning in $\mathcal{H}_6$ versus winning in $\mathcal{H}_7$.

$\mathcal{H}_7$ **to** $\mathcal{H}_8$**:** $ct'^1 := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_1}, p'; r_{ct}) \rightarrow \widehat{ct}'^1 := \mathsf{Enc}_{PKE}(pk_\dagger^{\mathsf{i}_1}, p'^1; r_{ct})$, where $p'^1 \leftarrow \mathsf{Prove}_{VRF}(sk^{\mathsf{i}_1}, t||\mathsf{msg}||R)$. The argument is identical to the transition from $\mathcal{H}_6$ to $\mathcal{H}_7$ and thus we omit details.

$\mathcal{H}_8$ **to** $\mathcal{H}_9$**:** $\sigma = (v, v', \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{i}_0}, hk^{\mathsf{i}_1}, \pi, t) \rightarrow$

$$\sigma = (v^1, v', \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{i}_0}, hk^{\mathsf{i}_1}, \pi, t)$$

where $v^1 \leftarrow \mathsf{Eval}_{VRF}(sk^{\mathsf{i}_1}, \mathsf{msg}||P)$.

We show that the change between $\mathcal{H}_8$ and $\mathcal{H}_9$ is indistinguishable using the following reduction to the VRF key privacyexperiment in Figure 11.

1. $\mathcal{A}_{KP}^{VRF}$ gets a vk^0, vk^1 from his challenger.
2. $\mathcal{A}_{KP}^{VRF}$ picks $\mathsf{i}_0, \mathsf{i}_1$ and activates $\mathcal{A}_{anon}$.
3. $\mathcal{A}_{KP}^{VRF}$ answers every query from $\mathcal{A}_{anon}$. At index i_0 he sets the VRF $vk^{\mathsf{i}_0} = vk^0$ and at i_1 he sets $vk^{\mathsf{i}_1} = vk^1$.
4. On an OSign query for msg_i, R_i, at i_0: $\mathcal{A}_{KP}^{VRF}$ asks the challenger to return $v \leftarrow \mathsf{Eval}_{VRF}(sk^b, \mathsf{msg}_i || R_i)$
5. When $\mathcal{A}_{anon}$ makes his challenge, $(s_0, s_1, \mathsf{msg}, R)$, then $\mathcal{A}_{KP}^{VRF}$ submits $\mathsf{msg}||R$ to the challenger as his challenge and gets back v^*. He uses this in the signature $\sigma = (v^*, ct, hk, h, \pi)$.

If $b = 0$, then the $\mathcal{A}_{KP}^{VRF}$ uses vk^{i_0} to answer queries. If $b = 1$, then $\mathcal{A}_{KP}^{VRF}$ uses vk^{i_1}. By the VRF's key privacy property, $\mathcal{A}_{KP}^{VRF}$ cannot distinguish between a v from vk^{i_0} and vk^{i_1}. Thus, $\mathcal{H}_8$ and $\mathcal{H}_9$ are indistinguishable.

$\mathcal{H}_9$ to $\mathcal{H}_{10}$: $\sigma = (v^1, v', \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{i}_0}, hk^{\mathsf{i}_1}, \pi, t) \to (v^1, v'^1, \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{i}_0}, hk^{\mathsf{i}_1}, \pi, t)$. The challenger replaces v^1 by $v'^1 \leftarrow \mathsf{Eval}_{PRF}(sk^{\mathsf{i}_1}, t||\mathsf{msg}||P)$. The argument is as in $\mathcal{H}_8$ to $\mathcal{H}_9$.

$\mathcal{H}_{10}$ to $\mathcal{H}_{11}$: $(hk, shk) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, \mathsf{i}_0) \to hk^1, shk^1 \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, \mathsf{i}_1)$.

Because of the SPB's index-hiding property (see Figure 12)we can next change the index for which hk is generated from i_0 to i_1. We construct $\mathcal{A}_{IH}$ as an adversary against SPB index hiding which uses $\mathcal{A}_{anon}$ as a subroutine.

1. $\mathcal{A}_{IH}$ picks $(N, \mathsf{i}_0, \mathsf{i}_1, \emptyset)$ (where N is the maximum ring size).
2. $\mathcal{A}_{IH}$ activates $\mathcal{A}_{anon}$ as a subroutine. On each query $\mathcal{A}_{IH}$ answers as the challenger would.
3. Eventually, $\mathcal{A}_{anon}$ requests a signature on i_0, i_1.
4. $\mathcal{A}_{IH}$ produces the signature as described in $\mathcal{H}_{10}$, except for he gives $(N, \mathsf{i}_0, \mathsf{i}_1)$ to the challenger. He uses (hk, shk) from the challenger to create the signature for $\mathcal{A}_{anon}$.
5. $\mathcal{A}_{IH}$ outputs same as $\mathcal{A}_{anon}$.

If $\mathcal{A}_{anon}$ wins with non-negligibly different probabilities in $\mathcal{H}_{10}$ and $\mathcal{H}_{11}$, then $\mathcal{A}_{IH}$ could win the index hiding experiment. We see then that $\mathcal{H}_{10}$ and $\mathcal{H}_{11}$ must be indistinguishable.

$\mathcal{H}_{11}$ to $\mathcal{H}_{12}$: Using $\tau^1 \leftarrow \mathsf{Open}_{SPB}(hk^1, shk^1, R, \mathsf{i}_1)$, select $sk_{\mathsf{F}}^{\mathsf{i}_0}, sk_{\mathsf{F}}^{\mathsf{i}_1}$ randomly when requested for OKGen and change the witness:

$$\mathcal{H}_{11}\ \widehat{\mathsf{w}}^0 = (VK^{\mathsf{i}_0}, VK^{\mathsf{i}_1}, \mathsf{i}_0, \mathsf{i}_1, \tau^{\mathsf{i}_0}, \tau^{\mathsf{i}_1}, p^1, p'^1, sk_{\mathsf{F}}^{\mathsf{i}_0}, sk_{\mathsf{F}}^{\mathsf{i}_1}, r_{ct_1}, r_{ct'_1}, , r_{E_0}, r_{E_1})$$

$$\mathcal{H}_{12}\ \mathsf{w}^1 = (VK^{\mathsf{i}_1}, VK^{\mathsf{i}_0}, \mathsf{i}_1, \mathsf{i}_0, \tau^{\mathsf{i}_1}, \tau^{\mathsf{i}_0}, p^1, p'^1, sk_{\mathsf{F}}^{\mathsf{i}_1}, sk_{\mathsf{F}}^{\mathsf{i}_0}, r_{ct_1}, r_{ct'_1}, r_{E_0}, r_{E_1})$$

and use to compute $\pi^1 \leftarrow \mathsf{Prove}_{NIWI}(x, \mathsf{w}^1)$. This change is indistinguishable because NIWI has witness indistinguishability.

In $\mathcal{H}_{12}$, the challenger is returning a signature for i_1. Because each hybrid is computationally indistinguishable from its predecessor, we see that signatures under s_0 and s_1 are indistinguishable to $\mathcal{A}_{anon}$. □

5 (Scoped) Linkable Thring Signatures

We extend the techniques in TRS to create a *linkable* threshold ring signature scheme LTRS. Linkability [LWW04] means that given two thring signatures for any two messages, one can verify whether (at least one of) the same signers contributed to both signatures. The verification is done via a Link algorithm that takes as input two thring signatures and outputs a bit indicating whether the two signatures are linked.

The security framework and construction presented in the following support *scoped* linkability, where two signatures link if they have been produced by related sets of signers for the same scope (e.g., context information)[7]. Scoped linkability is more fine-grained than linkability: two signatures are linkable if they have been produced w.r.t the same scope, but across different scopes signatures cannot be linked. Scope can be an arbitrary string. Using a scope string fixed in the scheme yields the conventional notion of linking. Like BDH$^+$ [BDH$^+$19b] recently did for ring signatures, we present the first construction of a *linkable* thring signature scheme in the plain model by building upon our thring signature scheme.

The standard security requirements for (scope-)linkable threshold ring signatures are correctness, unforgeability, *scoped linkability*, *linkable anonymity*, and *non-frameability*. Scoped linkability requires that even maliciously generated signatures need to link. With linkable anonymity, while it is possible to see that two signatures come from the same signer, it is not possible to determine which signer it is. Finally, non-frameability requires that an adversary, even after seeing many messages and signatures, cannot generate fresh signatures which will link to signatures that have been generated by honest parties.

Syntax. A linkable threshold ring signature scheme is a 5-tuple of algorithms $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy}, \mathsf{Link})$ and an extension of threshold ring signatures, where Sign and Vfy take an extra input sc (the scope). Thus, we do not detail the first four interfaces here. Finally, Link is defined as follows:

– $b \leftarrow \mathsf{Link}(\sigma_1, \sigma_2)$. On input two valid threshold ring signatures for the same scope, this deterministic algorithm outputs a single bit b.

5.1 Properties and Definitions

We now formally define the above mentioned properties.
Scoped Linkability. Intuitively, scoped linkability guarantees that signatures from non-disjoint sets of signers for the same scope will link. This is captured by giving the adversary access to honestly generated keys for a ring of size q (for any q) and have the adversary output $q + 1$ valid signatures for the same scope. The

[7] We note that this concept is not new and has previously been used within anonymous credential systems (cf. [CKL$^+$16]), direct anonymous attestation [BCC04] and also in context of traceable ring signatures [FS07].

```
$\mathbf{Experiment}\ \mathsf{ScopedLinkability}^{\mathcal{A}}(\lambda, q)$
  $pp \leftarrow \mathsf{Setup}(1^\lambda)$
  $(vk^s, sk^s) \leftarrow \mathsf{KGen}(pp)$ for $s \in [q]$
  $(\{(\mathsf{msg}_i, R_i, \sigma_i, t_i)\}_{i\in[q+1]}, \mathsf{sc}) \leftarrow$
                              $\mathcal{A}(\{(vk^s, sk^s)\}_{s\in[q]})$
  return 1 if:
    $\mathsf{Vfy}(\mathsf{msg}_i, R_i, \sigma_i, t_i, \mathsf{sc}) = 1$ for $i \in [q+1]$
    $R_i \subseteq \{vk^1, \ldots, vk^s\}$ for $i \in [q+1]$
    $\forall i \neq j \in [q+1] \ :\ \mathsf{Link}(\sigma^i, \sigma^j) = 0$
  return 0
```

Fig. 4. Scoped Linkability

```
$\mathbf{Experiment}\ \mathsf{Frameability}^{\mathcal{A}}(\lambda, q)$
  $pp \leftarrow \mathsf{Setup}(1^\lambda)$
  $(vk^s, sk^s) \leftarrow \mathsf{KGen}(pp)$ for $s \in [q]$
  $(\mathsf{st}, \mathsf{msg}^*, R^*, \sigma^*, t^*, \mathsf{sc}) \leftarrow \mathcal{A}^{\mathsf{OCorr},\mathsf{OSign}}(\{vk^s\}_{s\in[q]})$
    where $\sigma^* = (\sigma^*_1, \ldots, \sigma^*_{n^*})$
  $(\mathsf{msg}^\dagger, R^\dagger, \sigma^\dagger, t^\dagger) \leftarrow \mathcal{A}(\mathsf{st}, \{sk^s\}_{s\in[q]})$
    where $\sigma^\dagger = (\sigma^\dagger_1, \ldots, \sigma^\dagger_{n^\dagger})$
  return 1 if:
    $\mathsf{Vfy}(\mathsf{msg}^*, R^*, \sigma^*, t^*, \mathsf{sc}) = 1$
    $\mathsf{Vfy}(\mathsf{msg}^\dagger, R^\dagger, \sigma^\dagger, t^\dagger, \mathsf{sc}) = 1$
    $R^* \cup R^\dagger \subseteq P$
    $|R^* \cap P_{corr}| < t^*$
    $\exists i$ such that $\sigma^*_i$ was not obtained for
      $(\mathsf{msg}^*, R^*)$ from $\mathsf{OSign}$
    $R^* \cap R^\dagger \cap P_{corr} = \emptyset$
    $\mathsf{Link}(\sigma^*, \sigma^\dagger) = 1$
  return 0
```

Fig. 5. Non-frameability

adversary wins if none of them link to each other. The definition of linkability is reminiscent of unforgeability (Def. 10): the adversary with only knowledge of q keys cannot create $q+1$ signatures. We have a similar limitation with linkability as we had with unforgeability: our construction has us exclude malicious keys due to our use of $\mathcal{R}_{\mathcal{F}}$. Thus, we miss the situation where the adversary forges signatures using malicious keys, which could be trivially made to link. As before, this is inherent in our scheme due to the use of the VRF and the OWF which allows us to be in the plain model.

We note that the proposed signatures will be linkable for different scopes if the same message/ring is signed.

Definition 12 (Scoped Linkability). *A linkable threshold ring signature scheme satisfies* scoped linkability *if for every PPT adversary $\mathcal{A}$ and every q polynomially bounded in λ, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{ScopedLinkability}^{\mathcal{A}}(\lambda, q) = 1] \leq \mathsf{negl}(\lambda)\,.$$

Non-Frameability. Non-frameability guarantees that no adversary can generate fresh signatures which will link to signatures that have been generated by honest parties. The adversary has access to OCorr and OSign and can receive arbitrarily many signatures, and finally outputs a strong forgery, i.e., a fresh signature to a new message and subring. The adversary then learns all secret keys and wins if it can generate another signature which links to the former one, if no corrupted user was in the subring for both signatures (as this would allow for trivial attacks).

Definition 13 (Non-Frameability). *A linkable threshold ring signature scheme satisfies* non-frameability *if for every PPT adversary $\mathcal{A}$ and every q polynomially bounded in λ, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Frameability}^{\mathcal{A}}(\lambda, q) = 1] \leq \mathsf{negl}(\lambda)\,.$$

In our definition, we only consider signature schemes where the threshold signature consists of a list of individual signatures (as is also the case in the construction). This is because the same message can be signed for the same subring and scope, but by two disjoint sets of signers. Because of non-interactivity and inter-signer anonymity, the individual contributions of the signature cannot depend on the set of signers, and thus an adversary could create a fresh overall signature by combining parts from both signatures. The adversary could then to trivially win the frameability experiment, if the winning condition only excluded that the overall challenge signature σ^* has not been generated by OSign, while not having a real-world impact. To overcome this problem, we either must drop inter-signer anonymity or require an interactive process.

Linkable Anonymity. For linkable anonymity, it is not possible to decide from which signer in the ring the signatures came from, only what signatures are linked together. We capture this concept formally in the experiment in Figure 6.

The adversary picks two signing sets S_0, S_1 such that $|S_0| = |S_1$. We can assume that $S_0 \cap S_1 = \emptyset$ without loss of generality. By ordering members of each set, we have a correspondence from user i_k and j_k in S_0, S_1 respectively. We say a key $VK_0^s \in S_0$ is matched with a key $VK_1^s \in S_1$.

Then in one case, all signature queries with signer i_k are signed with i_k. Otherwise, all signature queries of i_k are signed with j_k (and vice versa). Then $\mathcal{A}$ can use Link for any signature gotten from the challenger. If $\mathcal{A}$ requested two signatures for i_k then these two signatures will always link.

Depending on the bit b the challenger generates, the requested signatures on the sets S_0, S_1 have this form: If $b = 0$ then the signers are the ones $\mathcal{A}$ asks for. If $b = 1$ then each signature on $i_k \in S_0$ is replaced with $j_k \in S_1$ (and vice versa). In the end, $\mathcal{A}$ must decide whether the signatures were signed according to what he requested, or whether they were all flipped.

We note a weakness in our scheme: neither S_0 nor S_1 can have a corrupted member in the set. That is, if user i_k is corrupted, then it will be easy for $\mathcal{A}$ to learn whether i_k's key was used to create a signature. Therefore, we have OCorr ignores any calls to users in $S_0 \cup S_1$. This is a rather weak definition in the context of ring signatures, but seems unavoidable when using a deterministic function such as VRFs.

Definition 14 (Linkable Anonymity). *A linkable threshold ring signature scheme satisfies* scope-exclusive linkable anonymity *if for every PPT adversary $\mathcal{A}$ and every q polynomially bounded in λ, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr[\mathsf{LinkableAnonymity}^{\mathcal{A}}(\lambda, q) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

5.2 Our Construction

Our threshold ring signature TRS is modular in the sense that we need only to add a few elements to TRS to turn it into a linkable thring signature scheme

```
**Experiment** $\mathsf{LinkableAnonymity}^{\mathcal{A}}(\lambda, q)$
  $pp \leftarrow \mathsf{Setup}(1^\lambda)$
  $(vk^s, sk^s) \leftarrow \mathsf{KGen}(pp)$ for $s \in [q]$
  $b \leftarrow \{0,1\}$
  $(S_0, S_1, \mathsf{st}) \leftarrow \mathcal{A}(\{vk^s\}_{s \in [q]})$
    let $S_0 = \{vk^{i_1}, \ldots, vk^{i_m}\}$ and $S_1 = \{vk^{j_1}, \ldots, vk^{j_m}\}$
  $(S_0^*, S_1^*, \mathsf{msg}^*, R^*, \vec{t}^*, \mathsf{sc}^*, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{OSign}, \mathsf{OSign}', \mathsf{OCorr}}(\mathsf{st})$
    where $\mathsf{OCorr}$ ignores any calls to users in $S_0 \cup S_1$
    where $\mathsf{OSign}'$ engages in a signing protocol with $\mathcal{A}$ on the given inputs
      mimicking all uncorrupted users
    where $\mathsf{OSign}$ ignores calls where $\exists m$ with $vk^{i_m} \in S$ or $vk^{j_m} \in S$ but $\{vk^{i_m}, vk^{j_m}\} \nsubseteq S$, and
      otherwise computes $S'$ by replacing all signers from $S_0$ with $S_1$ and vice versa, i.e.,
      with $S' = S \setminus (\{vk^{i_m}\}_{i_m \in S} \cup \{vk^{j_m}\}_{j_m \in S}) \cup (\{vk^{j_m}\}_{i_m \in S} \cup \{vk^{i_m}\}_{j_m \in S})$
      and then engages in signing protocols with $\mathcal{A}$ for sets $S$ and $S'$ $(b = 0)$ or $S'$ and $S$ $(b = 1)$.
  $\sigma^* \leftarrow \mathsf{Sign}(\mathsf{msg}^*, T_b^*, R^*, \vec{t}^*, \mathsf{sc}^*)$
      where $T_b^* = \{SK^s\}$ for $s \in S$ if $b = 0$ and $s \in S'$ if $b = 1$.
  $b' \leftarrow \mathcal{A}(\mathsf{st}, \sigma^*)$
  return a random bit if:
    $(\mathsf{msg}^*, R^*)$ was queried before, or
    $(\mathsf{sc}^*, S_j')$ has been queried before for some $j \in \{0,1\}$, or
    $S_0 \cap S_1 \neq \emptyset$
  return $b = b'$
```

Fig. 6. Linkable Anonymity.

including the concept of a scope. The full LTRS construction is presented in Figure 7.

Besides TRS keys, we include two VRF keys (vk_L, sk_L) (for linking) and (vk_{mal}, sk_{mal}) (for achieving non-malleability). For signing, a signer additionally evaluates the first VRF on the scope sc to get (v_L, p_L) and encrypts p_L into ct_L. The evaluation on scope is necessary to allow for scoped linkability. Then, it creates a key-pair (vk_{sOTS}, sk_{sOTS}) for a strong one-time signature (sOTS) scheme. The signer evaluates another VRF using sk_{mal} on vk_{sOTS} (i.e., "signs" the verification key) and encrypts p_{mal} into ct_{mal}. The purpose of the second VRF is for non-malleability, i.e., sk_{sOTS} is used to sign the partial signature and the final signature also includes the sOTS signature. As before, the signer evaluates a NIWI:

NIWI. The NIWI consists of the OR of three different languages:

$$\mathcal{L}_L' := (\mathcal{L}_{\mathcal{R}|_{VK}} \wedge \mathcal{L}_{\mathcal{R}_{Link}}) \vee (\mathcal{L}_{\mathcal{R}|_{VK'}} \wedge \mathcal{L}_{\mathcal{R}_{Link'}}) \vee \mathcal{L}_{\mathcal{R}_F}$$

The relation $\mathcal{R}_F$ is identical to the one in TRS and $\mathcal{R}|_{VK}$ is a straightforward adaption of the one in TRS. We added a new language for the relationship $\mathcal{R}_{Link}$, which allows a signer to maintain anonymity and non-frameability: $\mathcal{R}_{Link}(x, \mathsf{w}) \iff$

$$ct_L = \mathsf{Enc}_{PKE}(pk_\dagger, p_L; r_L) \ \wedge \ ct_{mal} = \mathsf{Enc}_{PKE}(pk_\dagger, p_{mal}; r_{mal}) \ \wedge$$
$$\mathsf{Vfy}_{VRF}(vk_L, \mathsf{sc}, v_L, p_L) = 1 \ \wedge \ \mathsf{Vfy}_{VRF}(vk_L, vk_{ots}, v_{mal}, p_{mal}) = 1$$

Key Generation $\mathsf{Gen}(1^\lambda)$:

```
$(vk, sk) \leftarrow \mathsf{Gen}_{VRF}(1^\lambda)$;
$(vk_L, sk_L) \leftarrow \mathsf{Gen}_{VRF}(1^\lambda)$ ;
$(vk_{mal}, sk_{mal}) \leftarrow \mathsf{Gen}_{VRF}(1^\lambda)$ ;
$(pk_\dagger, sk_\dagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
$(pk_\ddagger, sk_\ddagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
$sk_{\mathsf{F}} \leftarrow \{0,1\}^\lambda$;
$r_E \leftarrow Rnd_{PKE}$;
$E \leftarrow \mathsf{Enc}_{PKE}(pk_\ddagger, sk_{\mathsf{F}}; r_E)$;
$VK := (vk,\ vk_L, vk_{mal}\ , pk_\dagger, pk_\ddagger, E)$;
$SK := (sk,\ sk_L, sk_{mal}\ , sk_\dagger, sk_\ddagger, r_E, VK)$;
return $(VK, SK)$.
```

Verification $\mathsf{Vfy}(\mathsf{msg}, R, \sigma, t_V, \mathsf{sc})$:

```
  // Parse signature;
$\sigma = (\rho,\ \varsigma, \mathsf{vk}_{sOTS}\ )_{j=1}^{t}$;
$\rho = (v, v',\ v_L, v_{mal}\ , ct, ct', ct_L, ct_{mal},$
        $hk^s, hk^i, \pi, t^j,\ \mathsf{sc}^j\ )$;
Sort list by $t^i$;
for $j \in [t]$
  $h' := \mathsf{Hash}_{SPB}(hk^s, R)$;
  $h'' := \mathsf{Hash}_{SPB}(hk^i, R)$;
  $x := (\mathsf{msg}, R, v, v', ct, ct',\ ct_L,$
    $ct_{mal}\ , h', h'', hk^s, hk^i, \mathsf{sc})$;
  $b \leftarrow \mathsf{Vfy}_{NIWI}(x, \pi)$;
  $b \leftarrow b\ \wedge\ \mathsf{Vfy}_{sOTS}(\mathsf{vk}_{sOTS}, \rho, \varsigma)$;
    if $\ b = 1\ \wedge\ \sigma^j.v \neq \sigma^k.v\ \forall k \in [j-1]$
      $\mathcal{L}_V.\mathsf{append}(t^j)$;
endfor
if $\exists i \geq t_V:\ \mathcal{L}_V[i] \leq i$ return $1$;
return $0$
```

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

```
Let $t_i := |\sigma_i|, i \in \{1,2\}$
for $(j,k) \in [t_1] \times [t_2]$
  if $\sigma_1^j.v_L = \sigma_2^k.v_L$ return $1$
endfor
return $0$
```

Threshold Signing $\mathsf{Sign}(\mathsf{msg}, T, R, \vec{t}, \mathsf{sc})$:

```
// Every signer $s \in S$, $|S| \geq t$
$v \leftarrow \mathsf{Eval}_{VRF}(sk^s, \mathsf{msg}||R)$;
$p \leftarrow \mathsf{Prove}_{VRF}(sk^s, \mathsf{msg}||R)$;
$v' \leftarrow \mathsf{Eval}_{VRF}(sk^s, t^s||\mathsf{msg}||R)$;
$p' \leftarrow \mathsf{Prove}_{VRF}(sk^s, t^s||\mathsf{msg}||R)$;
$v_L \leftarrow \mathsf{Eval}_{VRF}(sk_L^s, \mathsf{sc})$;
$p_L \leftarrow \mathsf{Prove}_{VRF}(sk_L^s, \mathsf{sc})$;
$(vk_{sOTS}, sk_{sOTS}) \leftarrow \mathsf{Gen}_{sOTS}(1^\lambda)$;
$v_{mal} \leftarrow \mathsf{Eval}_{VRF}(sk_{mal}^s, vk_{sOTS})$;
$p_{mal} \leftarrow \mathsf{Prove}_{VRF}(sk_{mal}^s, vk_{sOTS})$;
$r_{ct}, r_{ct'},\ r_L, r_{mal}\ \leftarrow Rnd_{PKE}$;
$ct \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p; r_{ct})$;
$ct' \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p'; r_{ct'})$;
$ct_L \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p_L; r_L)$;
$ct_{mal} \leftarrow \mathsf{Enc}_{PKE}(pk_\dagger^s, p_{mal}; r_{mal})$;
$(hk^s, shk^s) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, s)$;
$h^s \leftarrow \mathsf{Hash}_{SPB}(hk^s, R)$;
$\tau^s \leftarrow \mathsf{Open}_{SPB}(hk^s, shk^s, R, s)$;
// Pick other ring member $i \neq s$
$i \leftarrow [N] \setminus s$;
$r_{E_0}, r_{E_1} \leftarrow Rnd_{PKE}$
$(hk^i, shk^{i_1}) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, i)$;
$h^i := \mathsf{Hash}_{SPB}(hk^i, R)$;
$\tau^i \leftarrow \mathsf{Open}_{SPB}(hk^{i_1}, shk^{i_1}, R, i)$;
// Call on the NIWI for language $\mathcal{L}'_L$
$\pi \leftarrow \mathsf{Prove}_{NIWI}(x, \mathsf{w})$
$\rho := (v, v',\ v_L, v_{mal}\ , ct, ct',\ ct_L, ct_{mal}\ ,$
    $hk^s, hk^i, \pi, t^s,\ \mathsf{sc}\ )$;
$\varsigma \leftarrow \mathsf{Sign}_{sOTS}(\mathsf{sk}_{sOTS}, \rho)$;
$\sigma^s := (\rho,\ \varsigma, \mathsf{vk}_{sOTS}\ )$;
// Every signer $s$ broadcasts the signature
broadcast $\sigma^s$;
// Final threshold ring signature
return $\sigma = \{\sigma^i\}_{i=1}^{t}$.
```

Fig. 7. Linkable threshold ring signature scheme LTRS (changes to TRS highlighted).

Statements x and witnesses w for $\mathcal{L}'_L$ are of the form:

$$x = \begin{pmatrix} \mathsf{msg} & R & v & v_{mal} & v_L \\ h^s & h^i & hk^s & hk^i & vk_{ots} \\ ct & ct_{mal} & \mathsf{sc} & ct_L & \varsigma_{ots} \end{pmatrix} \qquad \mathsf{w} = \begin{pmatrix} VK^s & VK^{s_1} & s & s_1 & \tau^s & \tau^i \\ p & p_{mal} & p_L & sk_{\mathsf{F}}^{i_0} & sk_{\mathsf{F}}^{i_1} & \\ r_{ct} & r_{ct'} & r_{E_0} & r_{E_1} & r_L & r_{mal} \end{pmatrix}$$

5.3 Security of Our Construction

In the following we state the security claims for our (scope) linkable thring signature scheme LTRS. The proofs are along the same lines as those for TRS and therefore omitted; proof sketches can be found inAppendix B.

Theorem 4. *If* F *is a OWF,* VRF *has residual unpredictability and unique provability,* NIWI *has perfect soundness, SPB is somewhere perfectly binding and PKE is perfectly correct, SPB is index hiding then* LTRS *is unforgeable.*

Theorem 5. *If* F *is a one-way function, the NIWI has perfect soundness, PKE is perfectly correct, SPB is somewhere perfectly binding, VRF has residual unpredictability and key collision resistance, and sOTS is strongly unforgeable then* LTRS *is non-frameable.*

Theorem 6. *If the NIWI is computationally witness-indistinguishable, PKE has key-privacy and CPA-security, and VRF has key privacy and residual pseudorandomness, then* LTRS *has linkable anonymity.*

Theorem 7. *If the NIWI has perfect soundness, SPB is somewhere perfectly binding, the VRF has unique provability, and PKE is perfectly correct, then* LTRS *is linkable.*

6 Instantiations and Future Directions

Our construction is generic but we have made some choices for convenience, i.e., the use of PKE instead of commitments as key-privacy is a natural well studied notion for PKEs. Thus, for a concrete instantiation there may be a number of choices and possible optimizations, which are outside the scope of this paper. Also, for asymptotics, the general algebraic circuit for the NIWI will have a polynomial expansion in the size of its input. The input is logarithmic in the number of users of the ring, and therefore the overall size of the proof is *polylogarithmic* in the size of the input. This is a natural limitation when relying on general building blocks, an issue that is also present in BDH$^+$ [BDH$^+$19b]. We discuss later how this could be circumvented.

Towards Post-Quantum Instantiations. If the instantiations of underlying primitives are post-quantum secure, then our thring signature scheme is also post-quantum secure. Post-quantum VRFs that rely on LWE [GHKW17] exist[8]

[8] Though key privacy and key collision resistance seem natural in this approach, a formal treatment is missing.

and so do key-private PKE schemes (e.g., based on LWE [LP11]). Many PKE schemes have key privacy, as this property immediately holds if the ciphertexts are pseudorandom. SPBs can be constructed from somewhere statistically binding hashing (SSBs). BDH$^+$ show in Appendix A.2 of [BDH+19a] how to turn two-to-one SSBs into SPBs. There are SSBs that rely only on the existence of a lossy/injective functions. One natural lossy/injective function is one built from the LWE problem [AKPW13]. The other building blocks we require are a post-quantum strong one-time signature scheme and OWFs. One example for the former is the Winternitz scheme [BDE+11] and for latter there are multiple candidates (e.g, from assumptions such as LWE or SIS or based on symmetric primitives as in Picnic [CDG+17] and related signature schemes).

The last concern is whether one can construct NIWIs that are post-quantum secure. While NIWIs in the post-quantum setting are not known, as discussed in a recent work by Chatterjee et al. [CGH+21] and based on an observation in [BKM09], the NIWI in the BDH$^+$ approach (and also ours) can be replaced with a two-message public coin argument systems (ZAPs [DN00]). This can be done by extending verification keys with the first message of the ZAP (cf. [CGH+21]). While ZAPs are known under the LWE assumption [BFJ+20, GJJM20], one requires to rely on subexponential hardness. To achieve standard polynomial hardness, it though might be possible to adapt the recent approach in [CGH+21], which uses the BDH$^+$ approach along with a novel ZAPs for a limited class of languages to achieve compact ring signatures in the plain model from LWE.

Potential Trade-offs. The most challenging part of our approach is proving that a verification key belongs to the ring of verification keys. Like BDH$^+$ [BDH+19b], our thring signature is asymptotically logarithmic but due to the insistence of being in the plain model, there are technical sticking points that guide our choice of building blocks. In particular, when we want to use NIWIs we require perfect soundness and thus like BDH$^+$ [BDH+19b] rely on SPBs. Clearly, if we move to knowledge sound NIZKs and thus allow a trusted setup (CRS), then we can move to computationally sound versions and in particular accumulators, e.g., Merkle-tree accumulators with log-sized membership witnesses or even ones with constant size. For the latter ones, the accumulators rely on a trusted setup. Using accumulators was already shown to be useful to get compact ring signatures [DKNS04] and in concurrent and independent work also more compact thring signatures [MHOY20]. However, the latter requires accepting trusted setup and the random oracle heuristic. We expect that our core idea could also be combined with these primitives when accepting these additional assumptions. Another direction to reduce the signature size would be replace the NIWI with a zk-SNARG or zk-SNARK. However, this would again require a trusted setup or the random oracle heuristic and additionally non-falsifiable assumptions in the latter case.

Acknowledgments. We thank anonymous reviewers for their comments. This work was in part funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 830929 (CyberSec4Europe) and by the Austrian Science Fund (FWF) and netidee SCIENCE under grant agreement P31621-N38 (PROFET).

References

AKPW13. Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 57–74. Springer, Heidelberg, August 2013.

AOS02. Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT*, 2002.

BBDP01. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, Heidelberg, December 2001.

BCC04. Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004*, pages 132–145. ACM Press, October 2004.

BCC$^+$15. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 243–265. Springer, Heidelberg, September 2015.

BDE$^+$11. Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, and Markus Rückert. On the security of the Winternitz one-time signature scheme. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 363–378. Springer, Heidelberg, July 2011.

BDH$^+$19a. Michael Backes, Nico Döttling, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Ring signatures: Logarithmic-size, no setup — from standard assumptions. Cryptology ePrint Archive, Report 2019/196, 2019. https://eprint.iacr.org/2019/196.

BDH$^+$19b. Michael Backes, Nico Döttling, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Ring signatures: Logarithmic-size, no setup - from standard assumptions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 281–311. Springer, Heidelberg, May 2019.

BFJ$^+$20. Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667. Springer, Heidelberg, May 2020.

BHKS18. Michael Backes, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Signatures with flexible public key: Introducing equivalence classes for public keys. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 405–434. Springer, Heidelberg, December 2018.

BK10. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086, 2010. https://eprint.iacr.org/2010/086.

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

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Heidelberg, December 2020.

BS13. Slim Bettaieb and Julien Schrek. Improved lattice-based threshold ring signature scheme. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 34–51. Springer, Heidelberg, June 2013.

BSS02. Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 465–480. Springer, Heidelberg, August 2002.

CCL07. Ya-Fen Chang, Chin-Chen Chang, and Pei-Yu Lin. A concealed t-out-of-n signer ambiguous signature scheme with variety of keys. *Informatica*, 18(4):535–546, 2007.

CDG$^+$17. Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1825–1842. ACM Press, October / November 2017.

CGH$^+$21. Rohit Chatterjee, Sanjam Garg, Mohammad Hajiabadi, Dakshita Khurana, Xiao Liang, Giulio Malavolta, Omkant Pandey, and Sina Shiehian. Compact ring signatures from learning with errors. LNCS, pages 282–312. Springer, Heidelberg, 2021.

CGS07. Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sublinear size without random oracles. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 423–434. Springer, Heidelberg, July 2007.

CKL$^+$16. Jan Camenisch, Stephan Krenn, Anja Lehmann, Gert Læssøe Mikkelsen, Gregory Neven, and Michael Østergaard Pedersen. Formal treatment of privacy-enhancing credential systems. In Orr Dunkelman and Liam Keliher, editors, *SAC 2015*, volume 9566 of *LNCS*, pages 3–24. Springer, Heidelberg, August 2016.

CLCH. Yu-Shian Chen, Chin-Laung Lei, Yun-Peng Chiu, and Chun-Ying Huang. Confessible threshold ring signatures. In *ICSNC 2006*.

CLRS10. Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. A lattice-based threshold ring signature scheme. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 255–272. Springer, Heidelberg, August 2010.

DAZ19. Dipayan Das, Man Ho Au, and Zhenfei Zhang. Ring signatures based on middle-product learning with errors problems. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje eddine Rachidi, editors, *AFRICACRYPT 19*, volume 11627 of *LNCS*, pages 139–156. Springer, Heidelberg, July 2019.

DKNS04. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In Christian Cachin and Jan

Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626. Springer, Heidelberg, May 2004.

DN00. Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st FOCS*, pages 283–293. IEEE Computer Society Press, November 2000.

DRS18. David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 419–440. Springer, Heidelberg, 2018.

DS19. David Derler and Daniel Slamanig. Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. *Des. Codes Cryptogr.*, 87(6):1373–1413, 2019.

DV09. Léonard Dallot and Damien Vergnaud. Provably secure code-based threshold ring signatures. In Matthew G. Parker, editor, *12th IMA International Conference on Cryptography and Coding*, volume 5921 of *LNCS*, pages 222–235. Springer, Heidelberg, December 2009.

DY05. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 416–431. Springer, Heidelberg, January 2005.

ESLL19. Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 115–146. Springer, Heidelberg, August 2019.

ESS$^+$19. Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 67–88. Springer, Heidelberg, June 2019.

FS90. Uriel Feige and Adi Shamir. Witness Indistinguishable and Witness Hiding Protocols. In Harriet Ortiz, editor, *STOC*, pages 416–426. ACM, 1990.

FS07. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 181–200. Springer, Heidelberg, April 2007.

GHKW17. Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 537–566. Springer, Heidelberg, November 2017.

GJJM20. Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 668–699. Springer, Heidelberg, May 2020.

GK15. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 253–280. Springer, Heidelberg, April 2015.

Gon19. Alonso González. Shorter ring signatures from standard assumptions. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 99–126. Springer, Heidelberg, April 2019.

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, December 2006.

HJ16. Dennis Hofheinz and Tibor Jager. Verifiable random functions from standard assumptions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 336–362. Springer, Heidelberg, January 2016.

HS. Abida Haque and Alessandra Scafuro. Threshold ring signatures:new definitions and post-quantum security. In *PKC 2020*.

LLNW16. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 1–31. Springer, Heidelberg, May 2016.

LP11. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

LPQ18. Benoît Libert, Thomas Peters, and Chen Qian. Logarithmic-size ring signatures with tight security from the DDH assumption. In Javier López, Jianying Zhou, and Miguel Soriano, editors, *ESORICS 2018, Part II*, volume 11099 of *LNCS*, pages 288–308. Springer, Heidelberg, September 2018.

LW. Jiqiang Lv and Xinmei Wang. Verifiable ring signature. In *DMS 2003*.

LW05. Joseph K. Liu and Duncan S. Wong. On the security models of (threshold) ring signature schemes. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04*, volume 3506 of *LNCS*, pages 204–217. Springer, Heidelberg, December 2005.

LW19. Hao Lin and Mingqiang Wang. Repudiable ring signature: Stronger security and logarithmic-size. Cryptology ePrint Archive, Report 2019/1269, 2019. https://eprint.iacr.org/2019/1269.

LWW04. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *ACISP 04*, volume 3108 of *LNCS*, pages 325–335. Springer, Heidelberg, July 2004.

Lys02. Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 597–612. Springer, Heidelberg, August 2002.

MCGL11. Carlos Aguilar Melchor, Pierre-Louis Cayrel, Philippe Gaborit, and Fabien Laguillaumie. A new efficient threshold ring signature scheme based on coding theory. *IEEE Transactions on Information Theory*, 57(7):4833–4842, 2011.

MHOY20. Alexander Munch-Hansen, Claudio Orlandi, and Sophia Yakoubov. Stronger notions and a more efficient construction of threshold ring signatures. Cryptology ePrint Archive 2020/678, 2020.

MRV99. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130. IEEE Computer Society Press, October 1999.

MS17. Giulio Malavolta and Dominique Schröder. Efficient ring signatures in the standard model. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 128–157. Springer, Heidelberg, December 2017.

OPWW15. Tatsuaki Okamoto, Krzysztof Pietrzak, Brent Waters, and Daniel Wichs. New realizations of somewhere statistically binding hashing and positional accumulators. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 121–145. Springer, Heidelberg, November / December 2015.

OTYO18. Takeshi Okamoto, Raylin Tso, Michitomo Yamaguchi, and Eiji Okamoto. A k-out-of-n ring signature with flexible participation for signers. Cryptology ePrint Archive, Report 2018/728, 2018. https://eprint.iacr.org/2018/728.

PBB13. Albrecht Petzoldt, Stanislav Bulygin, and Johannes Buchmann. A multivariate based threshold ring signature scheme. *Applicable Algebra in Engineering, Communication and Computing*, 24(3-4):255–275, 2013.

PS19. Sunoo Park and Adam Sealfon. It wasn't me! - Repudiability and claimability of ring signatures. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 159–190. Springer, Heidelberg, August 2019.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, December 2001.

Sah01. Amit Sahai. Simulation-sound non-interactive zero knowledge. Technical report, IBM RESEARCH REPORT RZ 3076, 2001.

TW05. Patrick P Tsang and Victor K Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ICISP*, 2005.

TWC$^+$04. Patrick P. Tsang, Victor K. Wei, Tony K. Chan, Man Ho Au, Joseph K. Liu, and Duncan S. Wong. Separable linkable threshold ring signatures. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 384–398. Springer, Heidelberg, December 2004.

YLA$^+$11. Tsz Hon Yuen, Joseph K. Liu, Man Ho Au, Willy Susilo, and Jianying Zhou. Threshold ring signature without random oracles. In Bruce S. N. Cheung, Lucas Chi Kwong Hui, Ravi S. Sandhu, and Duncan S. Wong, editors, *ASIACCS 11*, pages 261–267. ACM Press, March 2011.

A Formal Security Definitions for Building Blocks

A.1 Non-Interactive Witness-Indistinguishable Proof Systems

In the following we give provide formal definitions of the security properties that need to be satisfied by a NIWI.

Definition 15 (Perfect Completeness). *For all* $\lambda \in \mathbb{N}$ *and* $(x, \mathsf{w}) \in \mathcal{R}$ *it holds that* $\mathsf{Vfy}(x, \mathsf{Prove}(1^\lambda, x, \mathsf{w})) = 1$.

Definition 16 (Perfect Soundness). *For all* $\lambda \in \mathbb{N}$*, all* $x \notin \mathcal{L}_\mathcal{R}$*, and all* $\pi \in \{0,1\}^*$ *it holds that* $\mathsf{Vfy}(x, \pi) = 0$.

```
**Experiment** $\mathsf{WitnessIndistinguishability}^{\mathcal{A}}(\lambda, \mathcal{R})$
  $(x, \mathsf{w}_0, \mathsf{w}_1, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda, \mathcal{R})$
  $b \leftarrow \{0,1\}$
  $\pi \leftarrow \mathsf{Prove}(1^\lambda, x, \mathsf{w}_b)$
  $b' \leftarrow \mathcal{A}(\mathsf{st}, \pi)$
  return a random bit if:
    $(x, \mathsf{w}_0) \notin \mathcal{R}$ or $(x, \mathsf{w}_1) \notin \mathcal{R}$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 8. The witness-indistinguishability experiment for non-interactive proof systems.

Definition 17 (Witness Indistinguishability). *A proof system for a relation* $\mathcal{R}$ *is* witness indistinguishable *if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left| \Pr[\mathsf{WitnessIndistinguishability}^{\mathcal{A}}(\lambda, \mathcal{R}) = 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)\,.$$

A.2 Verifiable Random Functions

In the following we give provide formal definitions of the security properties that need to be satisfied by a VRF.

Definition 18 (Complete Provability). *A VRF has the* complete provability *property, iff for all* λ *and all* $x \in \{0,1\}^{a(\lambda)}$ *it holds that*

$$\Pr[\mathsf{Vfy}(vk, x, \mathsf{Eval}(sk, x), \mathsf{Prove}(sk, x)) = 1 : (vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)] = 1\,.$$

Definition 19 (Unique provability). *A VRF has the* unique provability *property, iff for all* vk*,* $x \in \{0,1\}^{a(\lambda)}$*,* $v_0, v_1 \in \{0,1\}^{b(\lambda)}$*,* p_0*,* p_1*, where* $v_0 \neq v_1$ *there exists* $i \in \{0,1\}$ *such that* $\mathsf{Vfy}(vk, x, v_i, p_i) = 0$.

```
**Experiment** $\mathsf{ResidualPseudorandomness}^{\mathcal{A}}(\lambda)$
  $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  $b \leftarrow \{0,1\}$
  $(x^*, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{OEval}(sk,\cdot)}(vk)$
    where $\mathsf{OEval}$, upon input $x$, adds $x$ to the initially empty set $\mathcal{Q}$ and
      returns $\mathsf{Eval}(sk, x)$
  $v_0 \leftarrow \mathsf{Eval}(sk, x^*)$
  $v_1 \leftarrow \{0,1\}^{b(\lambda)}$
  $b' \leftarrow \mathcal{A}(\mathsf{st}, v_b)$
  return a random bit if:
    $x^* \in \mathcal{Q}$, or
    $x^* \notin \{0,1\}^{a(\lambda)}$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 9. The residual pseudorandomness experiment for VRFs.

Definition 20 (Residual Pseudorandomness). *A VRF has the* residual pseudorandomness *property, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr[\mathsf{ResidualPseudorandomness}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

Residual pseudorandomness in particular also implies residual unpredictability defined next.

```
**Experiment** $\mathsf{ResidualUnpredictability}^{\mathcal{A}}(\lambda)$
  $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(x^*, v^*) \leftarrow \mathcal{A}^{\mathsf{OEval}(sk,\cdot)}(vk)$
    where $\mathsf{OEval}$, upon input $x$, adds $x$ to the initially empty set $\mathcal{Q}$ and
      returns $\mathsf{Eval}(sk, x)$
  return 1 if:
    $v^* = \mathsf{Eval}(sk, x^*)$, and
    $x^* \notin \mathcal{Q}$
  return 0
```

Fig. 10. The residual unpredictability experiment for VRFs.

Definition 21 (Residual Unpredictability). *A VRF has the* residual unpredictability *property, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{ResidualUnpredictability}^{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)\,.$$

Definition 22 (VRF Key Privacy). *A VRF has the* key privacy *property, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr[\mathsf{VRFKeyPrivacy}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

```
**Experiment** $\mathsf{VRFKeyPrivacy}^{\mathcal{A}}(\lambda)$
  $(vk_0, sk_0) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(vk_1, sk_1) \leftarrow \mathsf{Gen}(1^\lambda)$
  $b \leftarrow \{0,1\}$
  $b' \leftarrow \mathcal{A}(vk_0, vk_1)^{\mathsf{OEval}(sk_b,\cdot)}$
    where $\mathsf{OEval}$, upon input $x$, returns $\mathsf{Eval}(sk_b, x)$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 11. The key privacy experiment for VRFs.

Definition 23 (Key Collision Resistance). *A VRF has the* key collision resistance *property, iff for all λ and all $x \in \{0,1\}^{a(\lambda)}$ it holds that*

$$\begin{aligned}\Pr[sk_0 \neq sk_1 \ \wedge \mathsf{Eval}(sk_0, x) = \mathsf{Eval}(sk_1, x) :\\ (sk_0, vk_0) \leftarrow \mathsf{KGen}(1^\lambda), (sk_1, vk_1) \leftarrow \mathsf{KGen}(1^\lambda)] = 0\,.\end{aligned}$$

Key collision resistance turns out to be sufficient for the main construction given in the main body of this paper. However, for certain extensions a stronger notion also capturing maliciously generated keys needs to be considered.

Definition 24 (Strong Key Collision Resistance). *A VRF has the* strong key collision resistance *property, iff for all λ and all $x \in \{0,1\}^{a(\lambda)}$ it holds that*

$$\begin{aligned}\Pr[vk_0 \neq vk_1 \ \wedge v_0 = v_1 :\\ \mathsf{Vfy}(vk_0, x, v_0, p_0) = \mathsf{Vfy}(vk_1, x, v_1, p_1) = 1] = 0\,.\end{aligned}$$

A.3 Somewhere Perfectly Binding Hashing

In the following we give provide formal definitions of the security properties that need to be satisfied by an SPB hash. The first property ensures that if verification succeeds for a specific index and value, the scheme is perfectly binding at this particular index for this hash value h.

Definition 25 (Somewhere Perfectly Binding). *A SPB hash is* somewhere perfectly binding, *if for all λ, all n polynomially bounded in λ, all public hashing keys hk, all databases db, all indices $1 \leq \mathsf{i} \leq n$, and all witnesses τ it holds that:*

$$\Pr[\mathsf{Vfy}(hk, h, \mathsf{i}, x, \tau) = 1 \ \wedge \ db_{\mathsf{i}} \neq x : h \leftarrow \mathsf{Hash}(hk, db)] = 0\,.$$

This definition also covers the case where the public hashing key has not been honestly generated.

Definition 26 (Index Hiding). *A SPB hash is* index hiding, *if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\left|\Pr[\mathsf{IndexHiding}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

```
Experiment $\mathsf{IndexHiding}^{\mathcal{A}}(\lambda)$
  $(n, \mathsf{i}_0, \mathsf{i}_1, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda)$
  $b \leftarrow \{0,1\}$
  $(hk, shk) \leftarrow \mathsf{Gen}(1^\lambda, n, \mathsf{i}_b)$
  $b' \leftarrow \mathcal{A}(\mathsf{st}, hk)$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 12. The index hiding experiment.

```
Experiment $\mathsf{IND-CPA}^{\mathcal{A}}(\lambda)$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}(pk)$
  $b \leftarrow \{0,1\}$
  $ct \leftarrow \mathsf{Enc}(pk, m_b)$
  $b' \leftarrow \mathcal{A}(\mathsf{st}, ct)$
  return a random bit if:
    $\{m_0, m_1\} \not\subseteq M(\lambda)$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 13. The IND-CPA experiment.

A.4 Public Key Encryption

In the following we give provide formal definitions of the security properties that need to be satisfied by a PKE scheme.

Definition 27 (IND-CPA). *A PKE scheme is* IND-CPA secure*, if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\left|\Pr[\mathsf{IND-CPA}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

```
Experiment $\mathsf{KeyPrivacy}^{\mathcal{A}}(\lambda)$
  $(pk_0, sk_0) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(pk_1, sk_1) \leftarrow \mathsf{Gen}(1^\lambda)$
  $(m, \mathsf{st}) \leftarrow \mathcal{A}(pk)$
  $b \leftarrow \{0,1\}$
  $ct \leftarrow \mathsf{Enc}(pk_b, m)$
  $b' \leftarrow \mathcal{A}(\mathsf{st}, ct)$
  return a random bit if:
    $m \notin M(\lambda)$
  return 1 if:
    $b = b'$
  return 0
```

Fig. 14. The key privacy experiment.

Definition 28 (PKE Key Privacy [BBDP01]). *A PKE scheme is* key private*, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr[\mathsf{KeyPrivacy}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)\,.$$

A.5 Strong One-Time Signatures

In the following we give provide a formal definition of the security that need to be satisfied by a strong one-time signature scheme.

Definition 29 (Strong Unforgeability). *A one-time signature scheme is strongly unforgeable (strongly unforgeable under adaptively chosen message attacks), if for every PPT adversary $\mathcal{A}$ that makes a single query to* OSign' *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that* $\Pr[\mathsf{sUF\text{-}CMA}_{\mathcal{A}}^{\mathsf{sOTS}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$*.*

```
Experiment $\mathsf{sUF\text{-}CMA}_{\mathcal{A}}^{\mathsf{sOTS}}(\lambda)$
  $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \varsigma^*) \leftarrow \mathcal{A}^{\mathsf{OSign}'(sk,\cdot)}(vk)$
    where oracle $\mathsf{OSign}'$ on input $m$:
      $\varsigma \leftarrow \mathsf{OSign}(sk, m)$
      $\mathcal{Q} := \mathcal{Q} \cup \{(m, \varsigma)\}$
      return $\varsigma$
  return 1 if:
    $\mathsf{Vfy}(vk, m^*, \varsigma^*) = 1$ and
    $(m^*, \varsigma^*) \notin \mathcal{Q}$
  return 0
```

Fig. 15. The strong unforgeability experiment.

A.6 One-Way Functions

In the following we give provide formal definitions of the security properties that need to be satisfied by a one-way function.

```
Experiment $\mathsf{Invert}^{\mathcal{A}}(\lambda)$
  $x \leftarrow \{0,1\}^\lambda$
  $y \leftarrow \mathsf{F}(1^\lambda, x)$
  $x^* \leftarrow \mathcal{A}(1^\lambda, y)$
  return 1 if:
    $y = \mathsf{F}(1^\lambda, x^*)$
  return 0
```

Fig. 16. The one-wayness experiment.

Definition 30 (One Way). *A one-way function is* hard to invert*, if for every PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Invert}^{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)\,.$$

We provide a lemma that shows that the probability of finding random values that happen to be pre-images in a polynomially-sized list is negligible.

Lemma 1. *If* $\mathsf{F} : \{0,1\}^* \to \{0,1\}^*$ *is a one-way function, then the probability that the process:* $s = 1, \ldots, N$ $x_s \leftarrow \{0,1\}^\lambda$ *generates a pair* (x_i, x_j) *such that* $\mathsf{F}(x_i) = x_j$ *is negligible in* λ *(where* N *is at most* $\mathsf{poly}(\lambda)$*).*

Proof. Given a list $x_1, \ldots, x_N$, where each $x_s \leftarrow \{0,1\}^\lambda$ (with replacement) we want to find the probability that there exists $i, j \in [N]$ such that $F(x_i) = x_j$.

Because F is a function, for each value x_i, there is one image: $F(x_i) = y$. Thus, given two values (x_i, x_j), which are chosen uniformly at random, the probability that $x_j = y$ is $\frac{1}{2^\lambda}$.

There are $N(N-1)$ pairs where $i \neq j$.

$$\Pr[x_i, x_j \leftarrow D, F(x_i) = x_j] \leq \mathsf{negl}'(\lambda).$$

Where if $x_i = x_j$, from the few fixed points property we know it's less than $\mathsf{negl}(\lambda)$ and if $x_i \neq x_j$ it is $\frac{1}{2^\lambda}$, which is still a negligible function in λ.

Then we look at the case where $i = j$. There are N such pairs and by definition of few fixed points:

$$\Pr[x_i \leftarrow \{0,1\}^\lambda, F(x_i) = x_j] \leq \mathsf{negl}(\lambda).$$

Adding together, he overall probability of success is $T(\lambda) = \frac{N}{\mathsf{negl}'(\lambda)} + \frac{N(N-1)}{2^\lambda}$. Because N is polynomial in λ, it is much smaller than 2^λ. So we conclude that $T(\lambda)$ is negligible in λ as well.

B Security Proofs for of (Scoped) Linkable Thrings

Proof (of Theorem 5, sketch). At a high level, the adversary $\mathcal{A}$ must produce a signature that links to one of the honest keys in order to win. Suppose that there is an adversary who can create two signatures that link. Then let σ^* be the signature that he outputs in the first part of the experiment, and $\sigma^\dagger$ the signature that he outputs in the second half. It is sufficient to consider a single signature in each set, σ_i^* and $\sigma_j^\dagger$ and denote the corresponding signers as i and j. By an argument identical to the proof in Theorem 2, we can assume that $\mathcal{R}_F$ is not satisfied. Then, due to the fact that the respective keys have been generated honestly, the perfect soundness of the NIWI, the somewhere perfectly binding property of the VRF and the perfect correctness of PKE, we know that we can obtain $(p^i, p'^i, p_L^i, p_{mal}^i)$ and $(p^j, p'^j, p_L^j, p_{mal}^j)$ for the values contained in the signatures. Now, due to the key collision resistance of the VRF, we know that $i = j$ and due to the residual unpredictability of the VRF we know that

σ_i^* must have been generated by querying the signing oracle in the first phase of the experiment. Since these two signatures, however, are different, and due to the residual unpredictability of the VRF, we know that p_{mal}^j corresponds to an already queried one-time signing key. As we now, however, have two signatures under some vk_{sOTS} this contradicts the strong unforgeability of sOTS. □

Proof (of Theorem 6, sketch). In linkable anonymity, the challenger always responds correctly or with the opposite signer. In $b = 0$, the first signature is the "normal" one and the second is the "flipped" one. We morph the response from $b = 0$ to $b = 1$ by switching the signers chosen element by element via a sequence of hybrids, This proof is similar to the one for anonymity of TRS, but we need to account for the two extra VRF evaluations (by keeping track of what we decided v_L should be for the signers $\mathsf{i}_0, \mathsf{i}_1$ on the various scopes for consistency). □

Proof (of Theorem 7, sketch). The challenger creates a ring P of q keys. Suppose $\mathcal{A}$ produces $q+1$ signatures. All of these signatures verify and $R^s \in P$ for all $s \in [q+1]$ and $\forall i \neq j$ we have $\mathsf{Link}(\sigma_i, \sigma_j) = 0$.

By an argument identical to the proof in Theorem 2, we can assume that $\mathcal{R}_F$ is not satisfied. Then due to the perfect soundness of the NIWI we have that $\mathcal{L}_{\mathcal{R}_{|VK}} \wedge \mathcal{L}_{\mathcal{R}_{Link}}$ must be true. Thus, for any vk_L^s, $s \in [q+1]$, we have that $VK^s \in P$ because $\mathcal{L}_{\mathcal{R}_{|VK}}$ requires that $\mathsf{Vfy}(hk, h, s, VK^s, \tau) = 1$ for some location s in the ring. Because the SPB is somewhere perfectly binding, and there are only q positions, (by the pigeonhole principle) there must be a position i in P that $\mathcal{A}$ has bound to twice.

Second, $\mathcal{L}_{\mathcal{R}_{Link}}$ says $\mathsf{Vfy}(vk_L^i, \mathsf{sc}, v_L^s, \pi_L^s) = 1$. From $\mathcal{L}_{\mathcal{R}_{|VK}}$, we see that some sc and vk_L^i was used twice to create some v_L^i as well as v_L^*. However, this contradicts the unique provability of the VRF. □

C Additional Features and Extensions

In the following we briefly discuss how our threshold ring signature scheme relates to the recently introduced notions of (un)claimability and (un)repudiability [PS19] as well as flexibility [OTYO18].

C.1 Claimability and Repudiability

The concepts of (un)claimability and (un)repudiability were recently introduced by Park and Sealfon (PS) [PS19] (where claimability was already introduced under different names, e.g., in [LW, CLCH]). PS show that previous ring signature schemes do not give an answer either way about their claimability or repudiability. Repudiability means a user can prove that she did not generate a specific signature, while claimability means that a user can prove that she is accountable for a specific signature (without requiring her to keep state). The authors show that previous ring signature schemes do not give an answer either way about their claimability or repudiability. They are the first to formalize these notions and also provide three constructions: (1) a compiler to obtain claimable ring signatures in a generic black-box way from any standard ring signature scheme, (2) a construction of repudiable ring signatures from VRFs and ZAPs, i.e., two-message public coin witness indistinguishable proofs [DN00], and, (3) an unclaimable scheme based on the scheme of Brakerski and Kalai [BK10]. Below we discuss how our thring signatures can be viewed in the light of claimability and repudiability.

Claimability. A claimable ring signature allows a signer to de-anonymize himself by claiming he produced a signature. While the original ring signature is anonymous, here a signer must be able to show that he was one of the signing set by revealing some information, e.g., some randomness that only he could have used. A ring signature is unclaimable if anyone in the ring can also produce credible signing randomness for any signature.

Claimability of our thring signatures. Our thring signatures, due to the use of a VRF, naturally achieve claimability. For a given thring signature a signer can simply reveal (v, p) and any verifier can verify this and due to the unique provability and the (strong) key collision resistance will be convinced of the fact that the signer participated in producing the thring signature. This makes our thring signature scheme claimable for 'free'.

Repudiability. A repudiable ring signature allows ring members to prove that they did not sign a particular message and a scheme is unrepudiable if it is not possible to do so.

Repudiability of our thring signatures. We can achieve repudiability within our thring signatures by providing a proof that none of the pairs (v, p) from the VRF in a thring signature are generated by using the own VRF secret key without leaking anything beyond, i.e., demonstrating that the VRF evaluated on $\mathsf{msg}||R$ is different from every v value in the thring signature. Since we do not reveal the VRF output, we include a different NIZK (with a CRS here, which is not required for the actual thring signatures and only repudiability proofs).

C.2 Flexibility

Okamoto et al. in [OTYO18] introduce the concept of flexibility for threshold ring signature schemes. It allows signers to gradually upgrade an existing t-out-of-N ring signature to some $(t + \alpha)$-out-of-N ring signature. The α new signers engage in an interactive protocol with a trusted dealer to compute the updated signature, and the t previous signers do not need to be involved.

Flexibility of our thring signatures. Our construction and security framework directly support a non-interactive version of flexibility without requiring a trusted dealer in that new signers that are members of the ring (but did not sign previously) can add themselves to an already-created t-out-of-N threshold ring signature at any time by adding their own 1-out-of-N ring signature including their acceptable threshold. By merging the signatures, one achieves a $(t+1)$-out-of-N threshold signature, which can also be gradually updated.