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. Ring signatures are a cryptographic primitive that allow a signer to anonymously sign messages on behalf of an ad-hoc group of N potential signers (the so-called ring). This primitive has attracted significant research since its introduction by Rivest et al. (ASIACRYPT'01), but until recently, no construction was known that was both (i) compact, i.e., the signature size is sub-linear in N, and (ii) in the plain model, i.e., secure under standard hardness assumptions without requiring heuristic or setup assumptions. The first construction in this most desirable setting, where reducing trust in external parties is the primary goal, was only recently presented by Backes et al. (EUROCRYPT'19).

An interesting generalization of ring signatures are t-out-of-N ring signatures for $t \geq 1$, also known as threshold ring (thring) signatures (Bresson et al., CRYPTO'02). For threshold ring signatures, non-linkable sub-linear-size constructions are not even known under heuristic or setup assumptions.

In this work, we propose the first sub-linear thring signatures and prove them secure in the plain model. While our constructions are inspired by the template underlying the Backes et al. construction, they require novel ideas and techniques. Our scheme is non-interactive, and has strong inter-signer anonymity, meaning that signers do not need to know the other signers that participate in a threshold signing. We then present a linkable counterpart to our non-linkable construction. Our thring signatures can easily be adapted to achieve the recently introduced notions of flexibility (Okamoto et al., EPRINT'18) as well as claimability and repudiability (Park and Sealfon, CRYPTO'19).

(Th)Ring signatures and, in particular, their linkable versions have recently drawn significant attention in the field of privacy-friendly cryptocurrencies. We discuss applications that are enabled by our strong inter-signer anonymity, demonstrating that thring signatures are interesting from a practical perspective also.

* Work partly done while visiting AIT Austrian Institute of Technology, Vienna, Austria.

1 Introduction

Ring signatures, first introduced by Rivest, Shamir, and Tauman [47], 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 members, but cannot learn who the actual signer is, a property known as *(signer) anonymity*. Bresson, Stern, and Szydlo [10] generalized ring signatures to t-out-of-N ring signatures (aka threshold ring signatures or *thring signatures*), in which $t > 1$ distinct members of an ad-hoc set need to participate to produce a signature.

In a ring signature, there is no prescribed method to distribute keys among members. The ring can be set-up free, where members can join at will by publishing their public key. Members can sign at will with respect to a ring. Despite this setting, many ring signature schemes rely on the random oracle model or need a trusted setup, i.e., a common reference string. The plain model is the most desirable model for such an ad-hoc primitive as ring signatures, as it avoids any setup or heuristic assumptions and is based on standard and falsifiable hardness assumptions. Furthermore, most ring signature schemes are linear in the size of the ring, being problematic when ring sizes can be large. Recently, BDH$^+$ presented an elegant construction of the first sub-linear ring signatures in the plain model in [2].

While the issues of model and signature size appear in ring signatures (and we detail these in Section 1.3), thring signatures with $t > 1$ have another issue. While signers need not interact to create a 1-out-of-N ring signature, this is not immediately obvious for thring signatures with $t > 1$. For example, in a list of 1-out-of-N ring signatures, a single signer may sign repeatedly. As such, it is not easy to show that the signers are distinct can be complicated for existing ring signatures. Almost all thring signature schemes require that the signers know each other and that they interact. This raises the practical question of how potential signers can discover who the others are, or produce a thring signature without having to reveal their identity.

There are two exceptions that remove interactivity but have other drawbacks: First, Okamoto, Tso, Yamaguchi, and Okamoto [43] design 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, which is a strong assumption for such an ad-hoc distributed primitive. Second, Liu, Wei, and Wong [35] introduced the linkable ring signatures, which allows a verifier to publicly check whether two signatures were produced by the same signer (while still preserving the anonymity property). The authors note that this could be extended to produce threshold ring signatures. With a list of 1-out-of-N linkable ring signatures on some message, the signature verification algorithm checks pairwise that no two signatures in the list are linked to the same ring member. This approach is generic, but only works for *linkable* thring signatures, as any signer that participates in two threshold ring signatures becomes linkable. Using this approach with the recent sub-linear scheme of BDH$^+$

in the plain model [2], one obtains a sub-linear sized threshold ring signature scheme which is linkable; yet, when aiming for non-linkable thring signatures, it only yields a *one-time* scheme. Also, it is not obvious how to obtain *scoped* linkability.

Relevance of such schemes. Recently, (linkable) ring signatures have drawn significant attention in the field of privacy-friendly cryptocurrencies, particularly in Monero[3]. With a market capitalization of $1.47B[4] (at the time of writing), Monero is the largest privacy-friendly cryptocurrency. In contrast to other approaches such as Zcash[5], Monero focuses on avoiding any kind of trusted setup. This is closer to the original spirit of cryptocurrencies, whose main goal is to avoid centralization. Monero recently announced[6] that they are planning to introduce threshold ring confidential transactions, for which (linkable) thring signatures are an attractive building block. We expect a boost in the popularity of threshold ring signatures, as Monero consider them and are actively working on them [26].

We extend the original whistleblower example from Rivest et al. [47] 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 even 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 signing a new thring signature.

Another potential application is voting. Linkable ring signatures are a possible solution to e-voting, as shown by Liu and Wei [50]. In voting, nobody should know for whom a voter cast his vote, but the voter must not be allowed to vote more than once. A voter registers using their verification key. To cast a vote, she signs a ring signature. This signature does not reveal the voter's identity. A verifier can take all signatures on the ring and tally votes for each candidate. Any verifier can see valid distinct signatures (and thus valid, distinct votes) without any auditor needing to carry out additional checks. Because we have the feature of scoped linkability, it is possible for signers to use the same verification key to vote for other candidates. For example, say a voter wishes to vote for both mayor and governor. Votes cast under the scope 'mayor' are linkable, so that nobody can double vote for mayor. The same is true for the scope 'governor'. A voter can now vote for both offices separately using the same verification key. Fur-

[3] https://getmonero.org/
[4] https://coinmarketcap.com/currencies/monero/
[5] https://z.cash/
[6] https://forum.getmonero.org/73b46fdf

thermore, because anyone can join the ring, anyone who wants to vote can. As a result, scoped linkable thring signatures might be a valuable tool for e-voting schemes in the future.

Besides the applications as described above, we expect threshold ring signatures that have inter-signer anonymity and our new feature of scopable linkability will extend the scope of applications.

1.1 Our Contribution

Given the limitation of existing thring signatures and also the increasing real world interest, we construct a thring signature which is simultaneously:

(i) sub-linear in size of the ring,
(ii) in the plain model (i.e., without requiring random oracles or trusted setup, e.g., a common reference string) from standard assumptions,
(iii) and non-interactive (in particular, where signers need not know each other).

None of those were previously achieved (even in isolation). We elaborate on the contributions here:

- We present and prove the first construction of thring signatures that both have *sub-linear* signature sizes and are *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$^+$ [2]. However, their approach does not allow for a thring signature scheme as a straightforward extension. We require novel ideas and techniques, which we review in Section 1.2.
- We create a thring signature scheme in a setting where there is no interaction among the mutually anonymous signers. Our scheme also achieves *strong inter-signer anonymity*. Every signer locally computes a signature and the thring signature is just the collection of the individual signatures. Signers need not know the other signers that participate in a threshold signing. Previous constructions of thring signatures require rounds of interaction via reliable broadcast channels, which seems hard to achieve in practice for such an ad-hoc privacy-preserving primitive. We will discuss our solution in Section 1.2.
- We adapt the current model of *linkability* of thring signatures and make this model more flexible and fine-grained by introducing the concept of a *scope* to support *scoped linkability*.[7]
  To build upon our thring signature scheme, as was also done recently by BDH$^+$ [2] for ring signatures, we present the first construction of a logarithmic-size *linkable* thring signature scheme in the plain model. While linkability means that *any two* signatures produced by the same signer are linkable, scope is more fine-grained, i.e., two signatures are linkable if they have been

[7] This is in the vein of scope-exclusive pseudonyms in the context of attribute-based credential systems (cf. [11]).

produced on the same scope, but across different scopes signatures cannot be linked. Scope thereby can be an arbitrary string and can include context information. Using a scope string fixed in the scheme yields the conventional notion of linking.

1.2 Overview of Our Techniques

Before we start describing our approach and techniques, we discuss the approach used by Backes et al. [2] (BDH$^+$ for short). BDH$^+$ is in turn inspired by the approach to construct linear-size ring signatures in the plain model due to Bender, Katz, and Morselli [5].

In BDH$^+$, a user $i \in [N]$ generates key pairs $(vk_\sigma^i, sk_\sigma^i)$ and (pk^i, sk^i) of a signature scheme and a public-key encryption scheme, respectively, and sets the verification and signing key to $VK^i := (vk_\sigma^i, pk^i)$ and $SK := (sk_\sigma^i, sk^i)$. To produce a signature for message m with respect to ring $R = (VK^1, \ldots, VK^N)$, a signer j computes a signature σ on m using sk_σ^j and encrypts σ under pk^j resulting in a ciphertext ct. The signer samples another random ciphertext ct' (representing a user j') and generates two hashing keys hk and hk' of a somewhere perfectly binding (SPB) hashing scheme [44] that are binding at position j and j' respectively, and computes the hash of the ring $R = (VK^1, \ldots, VK^N)$ under both hk and hk', obtaining hash values h and h'. SPB hashing allows to collapse a ring R of N verification keys into a ring of just two keys (and membership witnesses are of size $\mathcal{O}(\log(N))$. This is a means to reduce the size of the witness for the membership proofs. Finally, signer j computes a perfectly sound NIWI proof π using an OR-statement which proves that either (hk, h) bind to a key VK^j and that ct encrypts a signature of m for VK^j or (hk', h') bind to a key $VK^{j'}$ and that ct' encrypts a signature of m for $VK^{j'}$. With a NIWI (instead of a NIZK), we can avoid a common reference string (CRS). Then a signature has the form $\Sigma = (ct, ct', hk, hk', \pi)$ and verification is straightforward.

The approach of BDH$^+$ [2] does not allow for a straightforward extension to logarithmic-size thring signatures in the plain model with non-interactive signing. 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. Note that when using the aforementioned compiler from linkable ring signatures, as soon as a signer issues two signatures, even on different messages, they can be linked together. This feature contradicts the idea behind real thring signatures. BDH$^+$ encrypt the conventional signatures (which link to the actual signer) to allow for anonymity. Therefore, one requires a proof that all the respective signers are distinct. The signers could additionally prove that they are pairwise distinct, but the additional quadratic sized proof breaks the logarithmic size requirement. Also, here at least one signer needs to compute this proof, and as such needs to collect all other signatures. Such a scheme would not achieve inter-signer anonymity nor non-interactivity.

Succinct non-interactive zero-knowledge arguments (zk-SNARKs) [28] could be used to remove this quadratic blow-up to a constant-size proof. However, zk-

SNARKs again requires setup (CRS) or heuristic assumptions (ROM) as well as non-falsifiable assumptions and thus would not allow to stay in the plain model.

Our approach. We follow the BDH$^+$ template, but our approach has major modifications and novel ideas. First, instead of using a signature scheme, we use a verifiable random function (VRF) [41], inspired by the recent work on (un-)repudiability and (un-)claimability of ring signatures by Park and Sealfon [45].[8] A VRF is a function which outputs a pseudorandom value v and a proof p so that given the input m, values (v, p) and the corresponding verification key vk everyone can check correctness of the evaluation. But without knowing a proof p for some output v, the output is still pseudorandom. So, we reveal v but encrypt the proof p for our thring construction. Also, for our setting, we need an assumption called *key collision resistance* on the VRF, which requires that if the VRF is evaluated under different (honestly generated) public keys and the same message, the evaluations will not collide. This is a reasonable assumption is satisfied by natural VRF candidates such as the Dodis-Yampolskiy VRF [20]. 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.

Suppose that we replace the encrypted signature in BDH$^+$ with a plain VRF evaluation and the respective encrypted VRF proof and the rest follows the BDH$^+$ template outlined above.[9] Intuitively, anonymity holds as the value v does not leak the signer (it is pseudorandom) and unforgeability is based on the unpredictability of the VRF. Now the anonymity proof in BDH$^+$ works because signatures are encrypted. However, as within the OR language both evaluations of the VRF are available in plain in the thring signature, a proof strategy along the lines of BDH$^+$ always runs into a circular problem and thus fails.

We point out that in order for the verifier to check the signature, they need to know a description of the ring. The input of the VRF includes the description of the ring. This would indicate that the ring signature must always be linear in the size of the ring. However, if the ring is known beforehand or has a short public description then it is not necessary to send the ring with the signature. Alternatively, it is possible to change the domain of the VRF so as to compute on the hash of the ring instead (this was also noted by Park and Sealfon [45]). For simplicity, we include the ring as input everywhere in our scheme.

Thus, our second change is to add another ingredient to the setup. Finally, we will change the NIWI to include a third OR clause. The latter will allow the challenger in the anonymity proof to simulate the first two clauses of the OR language, which then allows to switch the witnesses to random in the anonymity

[8] We note that in a concurrent and independent work in [34], 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.

[9] In the concrete construction we add another evaluation of the VRF in order to account for the threshold, but this does not change our intuition given below.

proof. We are in the plain model and cannot use a common reference string (CRS) which would allow us to embed a simulation trapdoor that we could use in the anonymity proof. To avoid a CRS and thus still be able to use only NIWI proofs, we use the following trick. Each signer i adds an extra secret key sk^i_{F} into her overall secret key and a commitment to it, i.e., $E \leftarrow \mathsf{Enc}(pk^i, sk^i_{\mathsf{F}}; r)$ to the public key VK^i. Our third clause in the OR language now proves that for two users i_0 and i_i in the ring, it holds that $\mathsf{F}(sk^{i_0}_{\mathsf{F}}) = sk^{i_1}_{\mathsf{F}}$, where F is a one-way permutation (OWP), i.e., the clause shows that one of the two keys is the image of the other key under OWP 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) such that we can then use the witness for this clause of the OR proof to switch out the VRF witnesses to random. Due to how we use the VRF in our construction, we cannot achieve the strongest notion of anonymity from Bender et al. [5, 6] (i.e., anonymity against attribution attacks/full key exposure), where the adversary sees all the random coins for generating all the honest keys. We achieve anonymity with respect to adversarially chosen keys [5, 6], which is still a strong notion and allows our proof strategy to work.

Our approach to linkable thring signatures. As mentioned before, we are interested in scoped linkability so that it is possible to control linking in a fine-grained way. While using the compiler by Liu et al. [35] on the linkable version of the BDH$^+$ 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 scopes. Thus, it would not be possible to support a potential unbounded number of scopes. Apart from these issues, the "tagging trick" in BDH$^+$ adds significant overhead, as when they unroll their required $\mathsf{JointVerify}$ algorithm in the proof, they obtain 480 clauses, where each clause is a conjunction of 5 verification statements of a commitment scheme.

Our linkable thring signatures support an unbounded number of scopes and are a modular extension of our basic thring signatures. We get linkability on a scope by adding an additional VRF key pair to the user's keys and use the evaluation of the VRF on the scope for linking purposes (recall that fixing the scope in the scheme yields the conventional notion of linkability). We need to extend the language of the NIWI used for the OR proof to account for this additional VRF. For technical reasons, to achieve non-frameability, we need to guarantee non-malleability for the partial signatures. We use a variant of the folklore technique of extending the language of the proof system to obtain simulation-sound NIZKs [48, 49, 27], but use VRFs instead of PRFs or signatures. In particular, we use an additional VRF to "sign" a verification key of a strongly unforgeable one-time signature and use the corresponding one-time signing key to sign the respective partial signature.

Claimability and repudiability. Recently, Park and Sealfon in [45] introduced the notions of (un-)repudiability and (un-)claimability for ring signatures. Here, claimability means that a user can prove that she is accountable for a specific

signature (without requiring her to keep state), while repudiability means that she can prove that she did not generate a specific signature. The authors show that previous ring signature schemes do not give an answer either way about their claimability or repudiability. As such, they are the first to formalize these definitions. Our constructions satisfy both notions of repudiability and claimability, and we discuss this in more detail in Section C.1.

Flexibility. Okamoto et al. [43] introduced the notion of flexibility, which allows to update an existing t-out-of-N ring signature to some $(t + \alpha)$-out-of-N ring signature, where the α new signers cooperate with a trusted dealer to achieve this. The way how we construct our threshold ring signatures also allows us to achieve some kind of 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. Section C.2).

1.3 Related Work

Ring signatures. Ring signatures have been extensively studied and the most recent work focuses on (1) obtaining efficient logarithmic-size ring signatures in the random oracle model in discrete-logarithm hard groups [29, 8, 33], (2) sublinear-size post-quantum constructions in the (quantum) random oracle model from different assumptions, e.g., [32, 18, 23, 22, 17], (3) schemes from standard assumptions without random oracles (but a trusted setup) [9, 19], and (4) constructions in the plain model, i.e., without assuming random oracles or common reference string (i.e., without trusted setup). A first construction of linear-size signatures appears in [5], but for a long time, improvements were made only when assuming a common reference string (without random oracles) towards signature size $\mathcal{O}(\sqrt{N})$ [13], which was recently further improved to $\Theta(\sqrt[3]{N})$ in [25]. However, no progress in the plain model was made. Only recently has there been a series of works improving on ring signatures in the plain model. Malavolta and Schröder [39] presented the first practical linear-size ring signatures in the standard model which can instantiated in bilinear groups under knowledge-of-exponent assumptions. Backes et al. [3] 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 the plain model in [2].

Threshold ring signatures. Currently, there exist constructions from general assumptions [1, 14], classical assumptions [51, 36, 52, 43], and post-quantum assumptions [16, 12, 40, 7, 46, 30]. However, all thring signatures schemes so far are (1) at least linear in the size of the ring, (2) in the random oracle model (except for [52] which uses a common reference string), and (3) require interaction between the signers (meaning that signers must know each other). As already mentioned in the introduction, an exception to the last issue is the recent *flexible* scheme of Okamoto, Tso, Yamaguchi, and Okamoto [43], in which ring members can simply create a 1-out-of-N ring signature themselves, while also showing that they are a new signer. To reach a threshold of t, there needs to t of these 1-out-of-N ring signatures. Their solution introduces an additional

party, the *dealer*, who is fully trusted by everybody and issues short-term keys to potential signers. It enforces that no signer will be issued more than one of these short-term keys.

Finally, a very recent, concurrent and independent work in [42] constructs threshold ring signatures of size $O(t)$ from a concrete traceable ring signatures with signature size $O(1)$. We note that 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, i.e., without requiring random oracles or trusted setup and from standard assumptions, their construction requires non-interactive zero-knowledge arguments of knowledge and thus either a common reference string or the random oracle heuristic[10].

2 Preliminaries

Throughout this paper, we denote the main security parameter by λ. We write $[N] = \{1, \ldots, N\}$, and $\boldsymbol{a} = (a_1, \ldots, a_N)$. We denote algorithms by A, B, ..., and write $out \leftarrow \mathsf{A}(in)$ to denote that out is assigned the output of the potentially probabilistic algorithm A with input in; sometimes, we will 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 called negligible, iff it vanishes faster than any inverse polynomial, i.e., $\forall k \in \mathbb{N}\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ :\ \mathsf{negl}(n) \leq n^{-k}$.

Non-Interactive Witness-Indistinguishable Proof Systems. Feige and Shamir [24] 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 candidate w, this algorithm outputs a proof π or $\perp$.*
- $b \leftarrow \mathsf{Vfy}(x, \pi)$*. Given a statement x and a proof π, this algorithm outputs either 0 or 1.*

We require NIWIs to 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

[10] They also inherently rely on random oracles in other parts of their construction.

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.

Finally, following BDH$^+$ [2], 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 $\mathsf{poly}(.)$, 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. VRFs were first introduced by Micali et al. [41], and instantiations in the standard model have, e.g., been proposed by Lysyanskaya [38], Dodis and Yampolskiy [20] as well as Hofheinz and Jager [31].

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 public key pk, 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.

We require VRFs to 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 [20]. We refer to App. A.2 for formal definitions.

Somewhere Perfectly Binding Hashing. Somewhere statistically binding

hashes were first introduced by Hubáček and Wichs [44]. 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.

In the original work [44] it was only required that such schemes are *statistically binding* at a single position. BDH$^+$ [2] strengthened this to *perfectly* binding. Furthermore, they introduced private openings to require a secret hashing key to compute a valid opening.

As shown in [44, 2] 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 [2] for DCR and DDH based instantiations of SPB based on [44].

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

- $(hk, shk) \leftarrow \mathsf{Gen}(1^\lambda, n, \mathsf{ind})$. *On input the security parameter λ in unary, a maximum database size n, and an index* ind*, 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 h, an index j, a value x and witness τ, this algorithm outputs a single bit b.*

We require SPBs to 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 ind 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 ind 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 $R(\lambda)$:*

- $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$. *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 corresponding keys, can decide for which public key a ciphertext has been computed. We refer to App. A.4 for formal definitions.

Strong One-Time Signatures. In the following we briefly recall the definition of strong one-time signature (sOTS) schemes.

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* pk *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*.*

We require sOTS schemes to satisfy the following 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 come up with another valid signature on any message. We refer to App. A.5 for formal definitions.

Definition 6 (One-Way Permutation). *A one-way permutation* 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* $y \in \{0,1\}^\lambda$*.*

We require OWPs to satisfy the following two properties. First, it must be *easy to compute*, meaning that there is a polynomial-time algorithm to evaluate the function. Second, the need to be *hard to invert*, guaranteeing that given only an output value, it is computationally infeasible to find the preimage mapping to this output. For F to be a permutation, it must be that any y has a unique preimage $x = \mathsf{F}^{-1}(y)$. We refer to App. A.6 for formal definition.

3 Framework and Security Definitions

In this section, we define the syntax for thring signature schemes. Next, we describe the oracles needed for the security definitions, and finally, the formal properties and definitions themselves.

Symbol	Meaning
t^s	Individual threshold of signer s[11]
$\boldsymbol{t}$	$= (t^{i_1}, \ldots, t^{i_{\lvert S \rvert}})$
N	Number of members of the ring. Indexed by s.
P	Ordered list of public keys $P = (VK^1, \ldots, VK^N)$.
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.

Table 1: Notation used in algorithm.

3.1 Syntax

While our notation is similar to that of Bender et al. [5], we need to extend the basic ring signature notation 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 $P = (vk^1, vk^2, \ldots)$ 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 in the message space $\mathsf{msg} \in \mathcal{M}$ with respect to a subring R. The secret keys of signers are denoted as T. Each signer $s \in S$ may choose an individual threshold t^s, with the sequence of all individual thresholds denoted as $\boldsymbol{t}$. Each signer chooses the minimum number of total signers they require for a valid signature. These thresholds may be arbitrary, i.e. t does not need to be fixed in the system parameters. Furthermore, in case of a non-interactive scheme, each signer may individually decide how many other signers are needed to let her signature become valid. For the sake of 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. 4.1 and Sec. 5 *do not require* such a Setup and are in the plain model.

Definition 7 (Threshold Ring Signature Scheme). *A threshold ring signature (thring) scheme consists of 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 $\boldsymbol{t}$.*

- $pp \leftarrow \mathsf{Setup}(1^\lambda)$. *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.*

[11] Here and in the following we use the convention that indices used to distinguish between signers are written as superscripts.

- $(vk, sk) \leftarrow \mathsf{KGen}(pp)$. *On input the public parameters pp, this PPT algorithm generates a public verification key vk and a corresponding secret key sk for a signer.*
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{msg}, T, R, \boldsymbol{t})$. *On input a message* msg*, a set of secret keys T, a subring R, and a vector of individual thresholds $\boldsymbol{t}$, 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 single 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. However, we must 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: $P, P_{corr}, \mathcal{L}_{\text{signers}}, Q$. 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 $\mathcal{L}$ 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 $\mathcal{L}$, vk^s to P, and returns vk^s to the adversary.
- $\mathsf{OSign}(\mathsf{msg}, S, R, \boldsymbol{t})$. On input a message msg, a list of signers S, a subring R, and a vector of individual thresholds $\boldsymbol{t}$, this oracle first checks whether $R \subseteq P$ and returns $\perp$ if this is not the case. The oracle then decomposes R to $S = S_{corr} \sqcup S_{hon}$[12], 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 $\mathsf{Sign}(\mathsf{msg}, T, R, \boldsymbol{t})$. 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.
- $\mathsf{OCorrupt}(s)$. 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}.
- $\mathsf{ORegister}(s, vk)$. 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. We note that 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 (ref. Sec. C.2).

[12] By $\sqcup$ we denote disjoint union.

Experiment $\mathsf{SigForge}^{\mathcal{A}}(\lambda)$
$pp \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{msg}^*, \sigma^*, R^*, t^*) \leftarrow \mathcal{A}^{\mathsf{OKGen},\mathsf{OCorrupt},\mathsf{OSign}}(pp)$
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 $\mathsf{Anonymity}^{\mathcal{A}}(\lambda)$
$pp \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{st}, \mathsf{msg}^*, R^*, S_0^*, S_1^*, \boldsymbol{t}) \leftarrow \mathcal{A}^{\mathsf{OKGen},\mathsf{OCorrupt},\mathsf{OSign},\mathsf{ORegister}}(pp)$
$b \leftarrow \{0,1\}$
$\sigma_b \leftarrow \mathsf{OSign}(\mathsf{msg}^*, S_b^* \setminus P_{corr}, \boldsymbol{t})$
$b' \leftarrow \mathcal{A}^{R^*,\boldsymbol{t}),\mathsf{OKGen},\mathsf{OCorrupt},\mathsf{OSign},\mathsf{ORegister}}(\mathsf{st}, \sigma_b)$
where $\mathsf{OCorrupt}$ and OSign ignore queries involving users in the set difference of S_0^* and S_1^*, i.e., in $(S_0^* \cup S_1^*) \setminus (S_0^* \cap S_1^*)$.
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
(msg^*, R^*) has been signed before
return 1 if:
$b = b'$
return 0

Fig. 2: Inter-signer anonymity

Definition 8 (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* $\boldsymbol{t} = (t^1, \ldots, t^N)$*, and any verification threshold* t *such that* $t \leq |\{i : t^i \leq t\}|$*, it holds that:*

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

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.

We note that we can tolerate corrupted parties in our scheme, but not malicious parties. This is due to the fact that we obtain 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 9 (Unforgeability wrt Insider Corruption). *A thring signature scheme satisfies* unforgeability wrt insider corruption *if for all PPT adversaries*

$\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that* $\Pr[\mathsf{SigForge}^{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

Anonymity. Anonymity says that it is infeasible to infer from a valid signature which users contributed to the generation of the signature, or in general to link a signer across different signatures. In our anonymity notion, we protect honest signers' identities even from other signers (*inter-signer anonymity*). This is true even if some of the signers in the challenge set use maliciously generated keys. 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^*.

We can tolerate malicious keys, even in the challenge sets, so long as both sets have the same malicious parties. Then the adversary interacts with the oracle using his malicious parties (if needed) and finally receives a signature. He may continue to make OSign and $\mathsf{OCorrupt}$ 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. As our scheme is set up to have inter-signer anonymity, if a message-ring pair has been signed before, it is possible to pinpoint who signed it. We ensure the threshold by preventing signers from signing twice on the same (msg, R). Thus, in the challenge phase of the anonymity experiment we require new message-ring pairs.

Definition 10 (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 TRS

4.1 Overview of TRS

In this section, we provide an overview of our construction. The details of the algorithms are in Fig. 3.

Our construction uses the following building blocks: *(i)* a verifiable random function VRF, *(ii)* a public key encryption scheme PKE, *(iii)* a somewhere perfectly binding hash function SPB, *(iv)* a one-way permutation F, and *(v)* a non-interactive witness indistinguishable proof system NIWI.

Signing. Suppose that t members of a ring R wish to sign a message msg. The ring is a sequence of public keys which we denote as $R = (VK^1, \ldots, VK^N)$ and 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 $h^i = \mathsf{Hash}(hk^i, R)$ and $h^s = \mathsf{Hash}(hk^s, R)$. Then (hk^s, h^s) and (hk^i, h^i) are commitments to VK^i and VK^s respectively. As in BDH$^+$ [2], this allows us to collapse a ring of size N into two keys. Finally, the signer calculates a proof on NIWI. 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 many individual signatures, i.e., $\sigma = (\sigma^{\mathbf{1}}, \ldots, \sigma^{\mathbf{t}})$. *Verification.* To verify a signature for a target threshold t_V, the verifier on $\sigma = (\sigma^{\mathbf{1}}, \ldots, \sigma^{\mathbf{t}})$ 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_V. 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_V signatures, the verifier will accept.

NIWI. To calculate a proof for the 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 $pk^s_\dagger$. 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 allow to use the proof technique of [2], but will never be satisfied for honest keys.

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, where $VK^s = (vk^s, pk^s_\dagger, pk^s_\ddagger, E)$. Statements and witnesses for the two relations have the form:

$$\begin{aligned}
\mathcal{R}|_{VK} : x &= (\mathsf{msg}, R, t, v, v', ct, ct', hk, h) \\
\mathsf{w} &= (VK^s, s, p, p', r_{ct}, r_{ct'}, \tau) \\
\mathcal{R}_F : x &= (R, h^{i_0}, h^{i_1}, hk^{i_0}, hk^{i_1}) \\
\mathsf{w} &= (i_0, i_1, VK^{i_0}, VK^{i_1}, \tau^{i_0}, \tau^{i_1}, sk^{i_0}_\mathsf{F}, sk^{i_1}_\mathsf{F}, r_{E_0}, r_{E_1})
\end{aligned}$$

The relations are then defined as follows:

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

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

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

$$
\begin{aligned}
&\mathsf{F}(sk^{i_0}_{\mathsf{F}}) = sk^{i_1}_{\mathsf{F}} \wedge \\
&\mathsf{Enc}(pk_\ddagger, sk^{i_0}_{\mathsf{F}}; r_{E_0}) = E_0 \wedge \\
&\mathsf{Enc}(pk_\ddagger, sk^{i_1}_{\mathsf{F}}; r_{E_1}) = E_1 \wedge \\
&\mathsf{Vfy}_{SPB}(hk^{i_0}, h^{i_0}, i_0, VK^{i_0}, \tau^{i_0}) = 1 \wedge \\
&\mathsf{Vfy}_{SPB}(hk^{i_1}, h^{i_1}, i_1, VK^{i_1}, \tau^{i_1}) = 1
\end{aligned}
$$

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

$$
x = \begin{pmatrix} \mathsf{msg} & R & v & v' & ct & ct' \\ t & h^{i_0} & h^{i_1} & hk^{i_0} & hk^{i_1} & \end{pmatrix} \qquad \mathsf{w} = \begin{pmatrix} VK^{i_0} & VK^{i_1} & i_0 & i_1 & \tau^{i_0} & \tau^{i_1} \\ \tau & p & p' & sk^{i_0}_{\mathsf{F}} & sk^{i_1}_{\mathsf{F}} & \\ r_{ct} & r_{ct'} & r_{E_0} & r_{E_1} & & \end{pmatrix}
$$

Having said this, the formal description of our threshold ring signature scheme TRS is now given in Fig. 3. Note that, as our instantiation does not rely on any trusted setup, there is no need for Setup to generate joint parameters.

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

$(vk, sk) \leftarrow \mathsf{Gen}_{VRF}(1^\lambda)$;
$(pk_\dagger, sk_\dagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
$(pk_\ddagger, sk_\ddagger) \leftarrow \mathsf{Gen}_{PKE}(1^\lambda)$;
$sk_{\mathsf{F}} \leftarrow \{0,1\}^{2\lambda}$;
$r_E \leftarrow PKE.R$;
$E \leftarrow \mathsf{Enc}(pk_\ddagger, sk_{\mathsf{F}}; r_E)$;
$VK := (vk, pk_\dagger, pk_\ddagger, E)$;
$SK := (sk, sk_\dagger, sk_\ddagger, r_E, VK)$;
return (VK, SK).

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

// 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 $i \in [t]$
  $h' := \mathsf{Hash}(hk^s, R)$;
  $h'' := \mathsf{Hash}(hk^i, R)$;
  $x := (\mathsf{msg}, R, v, v', ct, ct', h', h'', hk^s, hk^i)$;
  $b' \leftarrow \mathsf{Vfy}_{NIWI}(x, \pi)$;
  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 $\mathsf{Sign}(\mathsf{msg}, T, R, \boldsymbol{t})$:

// Every signer $s \in S$, $|S| \geq t$
$v \leftarrow \mathsf{Eval}(sk^s, \mathsf{msg}||R)$;
$p \leftarrow \mathsf{Prove}(sk^s, \mathsf{msg}||R)$;
$v' \leftarrow \mathsf{Eval}(sk^s, t^s||\mathsf{msg}||R)$;
$p' \leftarrow \mathsf{Prove}(sk^s, t^s||\mathsf{msg}||R)$;
$r_{ct}, r_{ct'} \leftarrow PKE.R$;
$ct \leftarrow \mathsf{Enc}(pk^s_\dagger, p; r_{ct})$;
$ct' \leftarrow \mathsf{Enc}(pk^s_\dagger, p'; r_{ct'})$;
$(hk^s, shk^s) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, s)$;
$h^s \leftarrow \mathsf{Hash}(hk^s, R)$;
$\tau^s \leftarrow \mathsf{Open}(hk^s, shk^s, R, s)$;
// Pick other ring member $i \neq s$
$i \leftarrow [N] \setminus s$;
$r_{E_0}, r_{E_1} \leftarrow PKE.R$;
$(hk^i, shk^i) \leftarrow \mathsf{Gen}_{SPB}(1^\lambda, N, i)$;
$h^i := \mathsf{Hash}(hk^i, R)$;
$\tau^i \leftarrow \mathsf{Open}(hk^i, shk^i, R, i)$;
// 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 σ^s;
// Final threshold ring signature
return $\sigma = \{\sigma^j\}_{j=1}^t$.

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

4.2 Security of Our Construction

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

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. It can easily be seen that by construction all individual signatures are valid. What remains to show is that $\sigma^i.v \neq \sigma^j.v$ for all $i \neq j$, which follows directly as otherwise we would have that $\mathsf{Eval}(sk^i, \mathsf{msg}||R) = \mathsf{Eval}(sk^j, \mathsf{msg}||R)$ in contradiction to the assumed key collision freeness. □

Theorem 2 (Unforgeability). *If* F *is a one-way permutation,* VRF *has residual unpredictability and unique provability,* NIWI *has perfect soundness, SPB is somewhere perfectly binding and PKE is perfectly correct, then the thring signature scheme* 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 as part of the signature, and the NIWI is perfectly sound, we know that the claimed statement must indeed be true. We can thereby exclude that $\mathcal{F}$ knows the witness to $\mathcal{R}_{\mathsf{F}}$ as no such one will exist unconditionally in our hybrids. This is why we do not give $\mathcal{F}$ access to the $\mathsf{ORegister}$ oracle. Then, as it needs to hold a witness to either $\mathcal{R}|_{VK}$ or $\mathcal{R}|_{VK'}$ due to the somewhere perfect binding property of the SPB, we know that 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 we know that ct (or ct') contains a valid proof for the respective VRF. Due to the unique provability of the VRF we know that such a value must have been generated by an uncorrupted signer, and then we can give a reduction 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 real 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 $\mathsf{OCorrupt}$ request of i^*. When the adversary provides a forgery, it must be the case that he used honest keys (created by OKGen queries) in his chosen ring. Since i^* was picked at random, i^* will be chosen to be part of the forgery's ring with at least probability $\frac{1}{q_{KG}}$ where q_{KG} is the number of users generated by OKGen. Suppose that there is an adversary $\mathcal{F}$ who who forges in $\mathcal{H}_0$ with some probability. Then the adversary will be able to 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 $(sk^j_{\mathsf{F}}, sk^k_{\mathsf{F}})$ it holds that either $sk^j_{\mathsf{F}} = \mathsf{F}(sk^k_{\mathsf{F}})$ or $sk^k_{\mathsf{F}} = \mathsf{F}(sk^j_{\mathsf{F}})$.

Note that $\mathcal{R}_{\mathsf{F}}$ can now never be satisfied among all the honest keys, in $\mathcal{H}_2$ we are at a point where 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. Finally, we present a reduction to the residual unpredictability $\mathcal{A}$ of the VRF, which uses $\mathcal{F}$ (as in $\mathcal{H}_2$) as a subroutine.

- $\mathcal{A}$ engages with a challenger $\mathcal{C}_{VRF}$ and receives vk which we embed into VK^{i^*}.
- On each request from $\mathcal{F}$: OKGen, $\mathsf{OCorrupt}$, OSign do as in $\mathcal{H}_2$. The difference is for each VRF evaluation at i^* $\mathcal{A}$ queries $\mathcal{C}_{VRF}^{\mathsf{OEval}(sk,\cdot)}$. Remember that if $\mathcal{F}$ requests $\mathsf{OCorrupt}$ 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 $\mathcal{F}$ created a valid forgery, then with probability $\frac{1}{q_{KG}}$ he picked the index i^*. 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 game 2. In the training phase, the adversary $\mathcal{A}_{anon}$ queries on OKGen, OSign, $\mathsf{OCorrupt}$, and $\mathsf{ORegister}$. 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, t-out-of-N 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 the various types of 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 real anonymity experiment 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 $\mathsf{ind}_0, \mathsf{ind}_1 \leftarrow [q_{KG}]$ ($\mathsf{ind}_0 \neq \mathsf{ind}_1$). If on either $\mathsf{ind}_0, \mathsf{ind}_1$, $\mathcal{A}_{anon}$ requests $\mathsf{OCorrupt}$ (or chooses these for $\mathsf{ORegister}$) then ABORT. Finally, we require that the adversary $\mathcal{A}_{anon}$ picks $\mathsf{ind}_0, \mathsf{ind}_1$ must be the two indices the challenger picked ahead of time. Because $\mathsf{ind}_0, \mathsf{ind}_1$ were picked randomly, there is a $\frac{1}{q_{KG}}$ probability that 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 $\frac{1}{(q_{KG})^2}$. **AH:** *TODO: need to say what the multiplicative loss is of.*

$\mathcal{H}_1$ to $\mathcal{H}_2$: In this step, the challenger always chooses ind_1 as the 'other index' when computing the final challenge signature. As ind_1 was uniformly random, this is indistinguishable.

$\mathcal{H}_2$ to $\mathcal{H}_3$: In this step, for OKGen on $\mathsf{ind}_0, \mathsf{ind}_1$ make sure the secret keys $sk_{\mathsf{F}}^{\mathsf{ind}_0}$ and $sk_{\mathsf{F}}^{\mathsf{ind}_1}$ are such that $\mathsf{F}(sk_{\mathsf{F}}^{\mathsf{ind}_0}) = sk_{\mathsf{F}}^{\mathsf{ind}_1}$ holds. This change only affects $sk_{\mathsf{F}}^{\mathsf{ind}_0}$ and $sk_{\mathsf{F}}^{\mathsf{ind}_1}$. This change affects only $sk_{\mathsf{F}}^{\mathsf{ind}_0}$ and $sk_{\mathsf{F}}^{\mathsf{ind}_1}$, which are hidden in E_{ind_0} and E_{ind_1}, and are never revealed.

$\mathcal{H}_3$ to $\mathcal{H}_4$: Calculate $(v^1, p^1) \leftarrow (\mathsf{Eval}(sk^{\mathsf{ind}_1}, \mathsf{msg}||R), \mathsf{Prove}(sk^{\mathsf{ind}_1}, \mathsf{msg}||R))$ and $(v'^1, p'^1) \leftarrow (\mathsf{Eval}(sk^{\mathsf{ind}_1}, t||\mathsf{msg}||R), \mathsf{Prove}(sk^{\mathsf{ind}_1}, t||\mathsf{msg}||R))$,

and $\tau^{\mathsf{ind}_1} = \mathsf{Open}(hk^{\mathsf{ind}_1}, shk^{\mathsf{ind}_1}, R, \mathsf{ind}_i)$. Then change the witness w:

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

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

Note that this only makes changes in the witness of the NIWI. 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 $\mathsf{ind}_0, \mathsf{ind}_1$.
2. For each query, he answers as the challenger would.
3. On a challenge $(s_0, s_1, \mathsf{msg}, R)$, if $s_0 = \mathsf{ind}_0$ and $s_1 = \mathsf{ind}_1$, 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 $\mathcal{A}_{anon}$.

We see w_0 is the same as $\mathcal{H}_2$ and $\widehat{\mathsf{w}}^0$ is the same as in $\mathcal{H}_3$. If $\mathcal{A}_{anon}$ wins $\mathcal{H}_2$ and $\mathcal{H}_3$ with different probabilities, then $\mathcal{A}_{WI}$ can win the witness-indistinguishability game with the same probability. Thus, $\mathcal{H}_2$ and $\mathcal{H}_3$ are indistinguishable.

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

To show that this change is indistinguishable, we construct an adversary to PKE key privacy $\mathcal{A}_{KP}^{PKE}$ following the game in 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 $\mathsf{ind}_0, \mathsf{ind}_1$. $\mathcal{A}_{KP}^{PKE}$ answers every query as the challenger would have done, except for KGen at ind_0 and ind_1, where he gives $pk^{\mathsf{ind}_0} = pk^0$ and $pk^{\mathsf{ind}_1} = pk^1$.
3. Finally, $\mathcal{A}_{anon}$ will request a signature on msg, R.
4. $\mathcal{A}_{KP}^{PKE}$ computes using sk^{ind_0}, $p \leftarrow \mathsf{Prove}(sk^{\mathsf{ind}_0}, \mathsf{msg}||R)$. 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 $\mathcal{H}_3$, but if he picks pk^1 then this is the game as in $\mathcal{H}_4$. Thus, if $\mathcal{A}_{anon}$ wins $\mathcal{H}_3$ and $\mathcal{H}_4$ with different probabilities, then this is the advantage of $\mathcal{A}_{KP}^{PKE}$ winning the PKE key privacy game.

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

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 $p' \leftarrow \mathsf{Prove}(sk^{\mathsf{ind}_0}, t||\mathsf{msg}||R)$ and thus we omit details.

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

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

1. $\mathcal{A}_{CPA}$ receives pk. $\mathcal{A}_{CPA}$ picks $\mathsf{ind}_0, \mathsf{ind}_1$, activates $\mathcal{A}_{anon}$.
2. For each query by $\mathcal{A}_{anon}$, $\mathcal{A}_{CPA}$ answers as a challenger would, except for OKGen at ind_0, where he gives $pk_{\dagger}^{\mathsf{ind}_0} = pk$.
3. When $\mathcal{A}_{anon}$ queries on $(s_0, s_1, \mathsf{msg}, R)$ then $\mathcal{A}_{CPA}$ calculates both $p \leftarrow \mathsf{Eval}(sk^{\mathsf{ind}_0}, \mathsf{msg}||R)$ and $p^1 \leftarrow \mathsf{Eval}(sk^{\mathsf{ind}_1}, \mathsf{msg}||R)$. 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. $\mathcal{A}_{CPA}$ gives the signature $\sigma = (v, ct^*, hk^{\mathsf{ind}_0}, hk^{\mathsf{ind}_1}, \pi)$.
6. $\mathcal{A}_{CPA}$ outputs the same as $\mathcal{A}_{anon}$.

If the challenger picks p, we are in $\mathcal{H}_4$, if p^1 then $\mathcal{H}_5$. Thus, $\mathcal{A}_{CPA}$ wins the CPA-security game with the same advantage as the difference of $\mathcal{A}_{anon}$ winning in $\mathcal{H}_4$ versus winning in $\mathcal{H}_5$.

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

$\mathcal{H}_8$ to $\mathcal{H}_9$: $\sigma = (v, v', \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{ind}_0}, hk^{\mathsf{ind}_1}, \pi, t) \rightarrow$
$\sigma = (v^1, v', \widehat{ct}^1, \widehat{ct}'^1, hk^{\mathsf{ind}_0}, hk^{\mathsf{ind}_1}, \pi, t)$
where $v^1 \leftarrow \mathsf{Prove}(sk^{\mathsf{ind}_1}, \mathsf{msg}||P)$.

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

1. $\mathcal{A}_{KP}^{VRF}$ gets a vk^0, vk^1 from his challenger.
2. $\mathcal{A}_{KP}^{VRF}$ activates $\mathcal{A}_{anon}$. $\mathcal{A}_{KP}$ picks $\mathsf{ind}_0, \mathsf{ind}_1$.
3. $\mathcal{A}_{KP}^{VRF}$ answers every query from $\mathcal{A}_{anon}$. At index ind_0 he sets the VRF $vk^{\mathsf{ind}_0} = vk^0$ and at ind_1 he sets $vk^{\mathsf{ind}_1} = vk^1$.
4. On an OSign query for msg_i, R_i, at ind_0: $\mathcal{A}_{KP}^{VRF}$ asks the challenger to return $v \leftarrow \mathsf{Eval}(vk^b, \mathsf{msg}_i||R_i)$
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 $\mathcal{A}_{KP}^{VRF}$ is answering queries with vk^{ind_0}. If $b = 1$, then $\mathcal{A}_{KP}^{VRF}$ is answering with vk^{ind_1}. Due the key privacy property of the VRF, $\mathcal{A}_{KP}^{VRF}$ cannot distinguish between a v from vk^{ind_0} and vk^{ind_1}. Thus, $\mathcal{H}_8$ and $\mathcal{H}_9$ are indistinguishable.

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

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

Because of the index-hiding property of SPB 12 we can next change the index for which hk is generated from ind_0 to ind_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 $(N, \mathsf{ind}_0, \mathsf{ind}_1, \emptyset)$ (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 ind_0, ind_1.
4. $\mathcal{A}_{IH}$ produces the signature as described in $\mathcal{H}_{10}$, except for he gives $(N, \mathsf{ind}_0, \mathsf{ind}_1)$ 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 $\tau^1 \leftarrow \mathsf{Open}(hk^1, shk^1, R, \mathsf{ind}_1)$, select $sk_{\mathsf{F}}^{\mathsf{ind}_0}, sk_{\mathsf{F}}^{\mathsf{ind}_1}$ randomly

when requested for OKGen and change the witness:

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

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

and use to compute $\pi^2 \leftarrow \mathsf{Prove}_{NIWI}(x, \mathsf{w}^1)$. This change is indistinguishable because NIWI has witness indistinguishability.

In $\mathcal{H}_{12}$, the challenger is returning a signature for ind_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 technique that we used to construct TRS to create a *linkable* threshold ring signature scheme LTRS. As introduced by Liu et al. [35], linkability means that given multiple thring signatures for different messages, it is possible to verify whether there was (at least) one signer contributing to both signatures. To achieve this, there is 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 are linkable if they have been produced by related sets of signers for the same scope (e.g., context information), thus achieving a more fine-grained notion than plain linkability (which can still be done by setting the scope to the empty string).

Besides correctness and unforgeability, the standard security requirements for (scope-)linkable threshold ring signatures are *scoped linkability*, *linkable anonymity*, and *non-frameability*. Scoped linkability requires that even maliciously generated signatures need to link. With linkable anonymity, while 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 LTRS is a 5-tuple of algorithms $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy}, \mathsf{Link})$. As LTRS is an extension of TRS, 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

In the following we now formally define scoped linkability, linkable anonymity, and non-frameability.

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

Experiment $\mathsf{Frameability}^{\mathcal{A}}(\lambda, q)$
$pp \leftarrow \mathsf{Setup}(1^\lambda)$
$(vk^s, sk^s) \leftarrow \mathsf{KGen}(pp)$ for $s \in [q]$
$(\mathsf{st}, \mathsf{msg}^*, R^*, \sigma^*, t^*, \mathsf{sc}) \leftarrow \mathcal{A}^{\mathsf{OCorrupt},\mathsf{OSign}}(\{vk^s\}_{s\in[q]})$
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 σ^*_i was not obtained for (msg^*, R^*) from OSign
$R^* \cap R^\dagger \cap P_{corr} = \emptyset$
$\mathsf{Link}(\sigma^*, \sigma^\dagger) = 1$
return 0

Fig. 5: Non-frameability

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 request the adversary to output $q+1$ valid signatures for the same scope. The adversary wins, if none of them link to each other.

Definition 11 (Scoped Linkability). *A 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 $\mathsf{OCorrupt}$ 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 is able to generate another signature which links to the former one, as long as no corrupted user was in the subring for both signatures (as this would allow for trivial attacks).

Definition 12 (Non-Frameability). *A 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 it is inherently possible to combine the individual contributions to a fresh overall signature by combining parts from both signatures. This would allow the adversary 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 would either have to drop inter-signer anonymity or require an interactive process.

Linkable Anonymity. Anonymity is maintained even though the signatures are linkable. It is not possible to decide which signer in the ring the signatures came from, only what signatures are linked together. We capture this concept formally in the Figure 6.

The adversary picks two signing sets S_0, S_1 that are the same cardinality and disjoint. 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, so this is consistent. The challenger creates two signatures: σ_0: where everything is done as expected, and σ_1: where everyone is flipped. In the end, $\mathcal{A}$ must decide whether all signatures were signed according to what he requested, or whether they were all flipped.

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 $S_1 = \{vk^{j_1}, \ldots, vk^{j_m}\}$
$(S_0^*, S_1^*, \mathsf{msg}^*, R^*, \boldsymbol{t}^*, \mathsf{sc}^*, j, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{OSign},\mathsf{OSign}',\mathsf{OCorrupt}}(\mathsf{st})$
where $\mathsf{OCorrupt}$ ignores any calls to users in $S_0 \cup S_1$
where OSign' engages in a signing protocol with the adversary on the given inputs, thereby mimicking all uncorrupted users
where OSign ignores calls where $\exists m$ with $vk^{i_m} \in S$ or $vk^{j_m} \in S$ but $\{vk^{i_m}, vk^{j_m}\} \not\subseteq S$, and otherwise computes S' by replacing all signers from S_0 by S_1 and vice versa, i.e., $S' = S \setminus (\{vk^{i_m}\}_{i_m \in S} \cup \{vk^{j_m}\}_{j_m \in S}) \cup (\{vk^{j_m}\}_{i_m \in S} \cup \{vk^{i_m}\}_{j_m \in S})$, and then engages in signing protocols with $\mathcal{A}$ for signing sets S and S' ($b = 0$) or S' and S ($b = 1$).
$\sigma^* \leftarrow \mathsf{Sign}(\mathsf{msg}^*, T_j^*, R^*, \boldsymbol{t}^*, \mathsf{sc}^*)$
$b' \leftarrow \mathcal{A}(\mathsf{st}, \sigma^*)$
return a random bit if:
(msg^*, R^*) was queried before, or
(sc^*, S'_j) has been queried before for some $j \in \{0,1\}$, or
$S_0 \cap S_1 \neq \emptyset$
return 1 if:
$b = b'$
return 0

Fig. 6: Linkable Anonymity

Definition 13 (Linkable Anonymity). *A 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

We modify our threshold ring signature TRS to be linkable. Our scheme 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 including the newly introduced concept of a scope. The full construction can be seen in the 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 as in TRS 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. He 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, which is discussed subsequently.

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 of our thring signatures and $\mathcal{R}|_{VK}$ is a straightforward adaption of the one for thring signatures. We added a new language for the relationship $\mathcal{R}_{Link}$, which allows a signer to maintain anonymity and non-frameability, and as described in the following: $\mathcal{R}_{Link}(x, \mathsf{w}) \iff$

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

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 threshold ring signature scheme. The proofs are along the same lines as those for the main construction and are therefore omitted; proof sketches can be found in Appendix B.

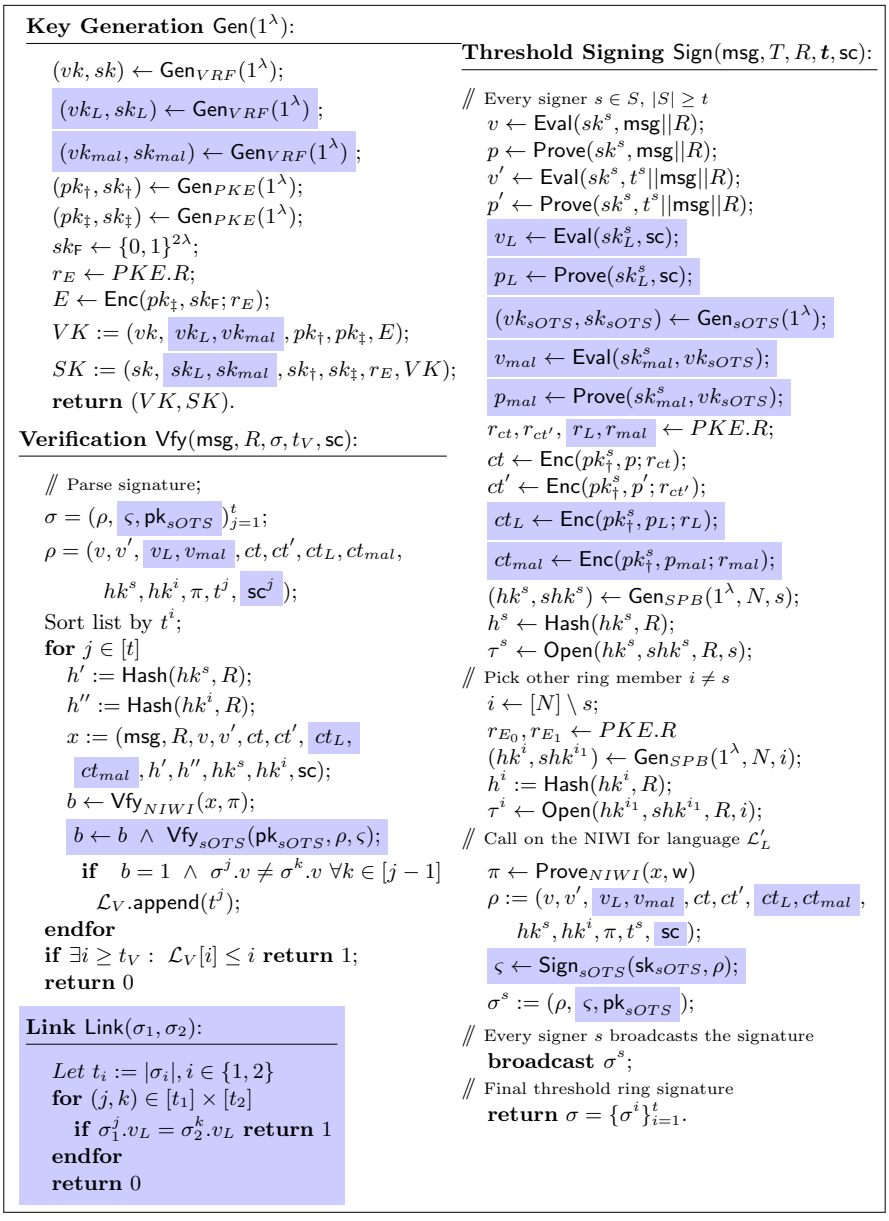

Fig. 7: Our linkable threshold ring signature scheme. Changes to the non-linkable version are highlighted in blue.

Theorem 4. *If* F *is a one-way permutation,* VRF *has residual unpredictability and unique provability,* NIWI *has perfect soundness, SPB is somewhere per-*

fectly binding and PKE is perfectly correct, SPB is index hiding then LTRS *is unforgeable.*

Theorem 5. *If* F *is a one-way permutation, 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* F *is a one-way permutation, 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.*

Acknowledgments. 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). We thank anonymous reviewers from EC'20 and CRYPTO'20 for their comments.

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer
2. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup - from standard assumptions. In: EUROCRYPT 2019, Part III
3. Backes, M., Hanzlik, L., Kluczniak, K., Schneider, J.: Signatures with flexible public key: Introducing equivalence classes for public keys. In: ASIACRYPT 2018, Part II
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: ASIACRYPT 2001
5. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: TCC 2006
6. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. Journal of Cryptology **22**(1), 114–138
7. Bettaieb, S., Schrek, J.: Improved lattice-based threshold ring signature scheme. In: Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013
8. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., Petit, C.: Short accountable ring signatures based on DDH. In: ESORICS 2015, Part I
9. Brakerski, Z., Kalai, Y.T.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086
10. Bresson, E., Stern, J., Szydlo, M.: Threshold ring signatures and applications to ad-hoc groups. In: CRYPTO 2002

11. Camenisch, J., Krenn, S., Lehmann, A., Mikkelsen, G.L., Neven, G., Pedersen, M.Ø.: Formal treatment of privacy-enhancing credential systems. In: SAC 2015
12. Cayrel, P.L., Lindner, R., Rückert, M., Silva, R.: A lattice-based threshold ring signature scheme. In: LATINCRYPT 2010
13. Chandran, N., Groth, J., Sahai, A.: Ring signatures of sub-linear size without random oracles. In: ICALP 2007
14. Chang, Y.F., Chang, C.C., Lin, P.Y.: A concealed t-out-of-n signer ambiguous signature scheme with variety of keys. Informatica **18**(4), 535–546
15. Chen, Y., Lei, C., Chiu, Y., Huang, C.: Confessible threshold ring signatures. In: ICSNC 2006
16. Dallot, L., Vergnaud, D.: Provably secure code-based threshold ring signatures. In: 12th IMA International Conference on Cryptography and Coding
17. Das, D., Au, M.H., Zhang, Z.: Ring signatures based on middle-product learning with errors problems. In: AFRICACRYPT 19
18. Derler, D., Ramacher, S., Slamanig, D.: Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In: Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018
19. Derler, D., Slamanig, D.: Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. Des. Codes Cryptogr. **87**(6), 1373–1413
20. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: PKC 2005
21. Dwork, C., Naor, M.: Zaps and their applications. In: 41st FOCS
22. Esgin, M.F., Steinfeld, R., Liu, J.K., Liu, D.: Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In: CRYPTO 2019, Part I
23. Esgin, M.F., Steinfeld, R., Sakzad, A., Liu, J.K., Liu, D.: Short lattice-based one-out-of-many proofs and applications to ring signatures. In: ACNS 19
24. Feige, U., Shamir, A.: Witness Indistinguishable and Witness Hiding Protocols. In: STOC
25. González, A.: Shorter ring signatures from standard assumptions. In: PKC 2019, Part I
26. Goodell, B., Noether, S.: Thring signatures and their applications to spender-ambiguous digital currencies. Cryptology ePrint Archive, Report 2018/774
27. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: ASIACRYPT 2006
28. Groth, J.: On the size of pairing-based non-interactive arguments. In: EUROCRYPT 2016, Part II
29. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: EUROCRYPT 2015, Part II
30. Haque, A., Scafuro, A.: Threshold ring signatures:new definitions and post-quantum security. In: PKC 2020
31. Hofheinz, D., Jager, T.: Verifiable random functions from standard assumptions. In: TCC 2016-A, Part I
32. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In: EUROCRYPT 2016, Part II
33. Libert, B., Peters, T., Qian, C.: Logarithmic-size ring signatures with tight security from the DDH assumption. In: ESORICS 2018, Part II
34. Lin, H., Wang, M.: Repudiable ring signature: Stronger security and logarithmic-size. Cryptology ePrint Archive, Report 2019/1269

35. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: ACISP 04
36. Liu, J.K., Wong, D.S.: On the security models of (threshold) ring signature schemes. In: ICISC 04
37. Lv, J., Wang, X.: Verifiable ring signature. In: DMS 2003
38. Lysyanskaya, A.: Unique signatures and verifiable random functions from the dh-ddh separation. In: Annual International Cryptology Conference. Springer
39. Malavolta, G., Schröder, D.: Efficient ring signatures in the standard model. In: ASIACRYPT 2017, Part II
40. Melchor, C.A., Cayrel, P.L., Gaborit, P., Laguillaumie, F.: A new efficient threshold ring signature scheme based on coding theory. IEEE Transactions on Information Theory **57**(7), 4833–4842
41. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: 40th FOCS
42. Munch-Hansen, A., Orlandi, C., Yakoubov, S.: Stronger notions and a more efficient construction of threshold ring signatures. Cryptology ePrint Archive, Report 2020/678
43. Okamoto, T., Tso, R., Yamaguchi, M., Okamoto, E.: A k-out-of-n ring signature with flexible participation for signers. Cryptology ePrint Archive, Report 2018/728
44. Okamoto, T., Pietrzak, K., Waters, B., Wichs, D.: New realizations of somewhere statistically binding hashing and positional accumulators. In: ASIACRYPT 2015, Part I
45. Park, S., Sealfon, A.: It wasn't me! - Repudiability and claimability of ring signatures. In: CRYPTO 2019, Part III
46. Petzoldt, A., Bulygin, S., Buchmann, J.: A multivariate based threshold ring signature scheme. Applicable Algebra in Engineering, Communication and Computing **24**(3-4), 255–275
47. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: ASIACRYPT 2001
48. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS
49. Sahai, A.: Simulation-sound non-interactive zero knowledge. Tech. rep., IBM RESEARCH REPORT RZ 3076
50. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: International Conference on Information Security Practice and Experience. Springer
51. Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. In: INDOCRYPT 2004
52. Yuen, T.H., Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Threshold ring signature without random oracles. In: ASIACCS 11

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 14 (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 15 (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 16 (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 17 (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 18 (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 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 19 (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 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 20 (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*

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

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 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 21 (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*

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

Definition 22 (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*

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

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 23 (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*

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

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 24 (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 $1 \leq \mathsf{ind} \leq n$, and all witnesses τ it holds that:*

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

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

Fig. 12: The index hiding experiment.

Note that this definition in particular also covers the case where the public hashing key has not been honestly generated.

Definition 25 (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*

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

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.

$\textbf{Experiment } \mathsf{IND-CPA}^{\mathcal{A}}(\lambda)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}(pk)$
$b \leftarrow \{0,1\}$
$ct \leftarrow \mathsf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}(\mathsf{st}, ct)$
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.

Definition 26 (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 27 (PKE Key Privacy [4]). *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 28 (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* $\Pr[\mathsf{sUF\text{-}CMA}^{\mathsf{sOTS}}_{\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

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

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

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

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

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 $\mathsf{ind}_0, \mathsf{ind}_1$ 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 [45] as well as flexibility [43].

C.1 Claimability and Repudiability

The concepts of (un)claimability and (un)repudiability were recently introduced by Park and Sealfon (PS for short) [45] (where claimability was already introduced under different names, e.g., in [37, 15]). PS 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 definitions 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 [21], and, (3) an unclaimable scheme based on the scheme of Brakerski and Kalai [9]. 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 themselves by claiming they produced a signature. While the original ring signature is anonymous, here a signer needs to be able to show that they were one of the signing set by revealing some information that only he could have produced to create the signature, 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 signatures 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 [43] introduce the concept of flexibility for threshold 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 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.