Triptych: logarithmic-sized linkable ring signatures with applications

Sarang Noether and Brandon Goodell
Monero Research Lab
{sarang,surae}.noether@protonmail.com

January 6, 2020

Abstract

Ring signatures are a common construction used to provide signer ambiguity among a non-interactive set of public keys specified at the time of signing. Unlike early approaches where signature size is linear in the size of the signer anonymity set, current optimal solutions either require centralized trusted setups or produce signatures logarithmic in size. However, few also provide linkability, a property used to determine whether the signer of a message has signed any previous message, possibly with restrictions on the anonymity set choice. Here we introduce Triptych, a family of linkable ring signatures without trusted setup that is based on generalizations of zero-knowledge proofs of knowledge of commitment openings to zero. We demonstrate applications of Triptych in signer-ambiguous transaction protocols by extending the construction to openings of parallel commitments in independent anonymity sets. Signatures are logarithmic in the anonymity set size and, while verification complexity is linear, collections of proofs can be efficiently verified in batches. We show that for anonymity set sizes practical for use in distributed protocols, Triptych offers competitive performance with a straightforward construction.

1 Introduction

First introduced in [21] with respect to RSA groups, ring signatures permit the signing of messages using public key sets not fixed in advance, without the need of a trusted group manager. Earlier constructions lacked such flexibility, requiring either centralized key setup or the establishment of fixed signing sets. Later work [2] established more robust security models for unforgeability and anonymity, capturing realistic threat models where an adversary is permitted to corrupt keys, convince honest signers to include malicious anonymity set members, or obtain signatures in advance.

Since a ring signature has an anonymity set of public keys, one of which is the true signer, the detection of signing by the same key requires an additional property, linkability. A linkable ring signature [16] enables verifiers to determine whether the (unknown) signer of a message has signed other messages. Such a construction was proposed for election applications, where it is necessary to ensure that votes are anonymous, but voters are permitted to vote only once on a particular issue. The construction in [16], with a hash-trapdoor structure similar to that of [22], is of particular interest due to potential flexibility in linking; while its linking is limited to signer-selected but unchanging signer anonymity sets, it permits linking on a per-issue basis. Additional recent work in [1] introduces the property of linkable anonymity, using sets of signatures and establishing restrictions on key corruption. Other related interesting properties like traceability [9, 8] imply stronger capabilities, where an attempt to sign two messages with the same key permits verifiers to identify the signer.

Linkable ring signatures have seen particular applications in signer-ambiguous transaction protocols. In such an application, transactions are authorized by a ring signature whose signing anonymity set consists of previously-generated transaction outputs. A signature demonstrates that the signer controls the private key of one such output without revealing which is the signer, and linkability is used to assure verifiers that output has not been used in another signature (signifying a double-spend attempt).

A practical consideration for the use of linkable ring signatures in transaction protocols is how signature size and verification time scale with the size of the anonymity set. In common applied constructions like

[19, 10], signature size and verification time scale linearly with the size of the signing ambiguity set; since such signatures are typically included in a public distributed data structure like a blockchain, there is a balance between the size of the anonymity set and the requirements for storage and verification. Recent protocol work in this area mitigates the size restriction. For example, in [24], the authors introduce a confidential transaction protocol based on a proving system whose size scales logarithmically with the anonymity set size, and which includes a method for demonstrating amount balance; amount commitment range proving is offloaded to other constructions like [4]. In [15], the authors use a more general proving technique to accomplish a similar goal; however, the protocol offers further size benefits by integrating commitment range proofs into the proof structure directly, taking advantage of the logarithmic proof size.

Other signer-ambiguous transaction protocols not based on linkable ring signatures offer more competitive performance. For example, protocols like [13] produce maximal theoretical signer ambiguity through the use of zero-knowledge Merkle proofs (among others) that offer extremely small proofs with low verification time, but at the cost of a trusted structured setup process that arises from the underlying proving system [11]. Like this work, [14] is a transaction protocol also based on [12]; however, it is intended to operate on commitments similar to those used in [18] but with the inclusion of amounts, and has limitations on addressing and sender tracing.

1.1 Our contribution

We produce two instantiations of a linkable ring signature, which we call Triptych. The constructions are a linkable generalization of Groth's one-of-many commitment-to-zero proving system [12], with optimizations from Bootle [3] applied for improvements to proof size and verification complexity. In one version of Triptych, the prover shows that it knows the opening of a commitment to zero within a commitment list, and also that it has constructed a linking tag using the same opening, enabling linkability. We then modify Groth's ring signature definitions to include linkability and a related property, non-frameability.

In the second version of Triptych, we further generalize to include a parallel list of commitments. Here, the prover proceeds as before to show its knowledge of an opening to a commitment in one list, as well as the construction of the linking tag. However, the proof also shows that the prover knows an opening of a commitment at the same position in the second commitment list. This construction has immediate application; in some signer-ambiguous confidential transaction protocols, transaction inputs are commitments to zero, for which the signer shows it knows the opening. Each commitment comes equipped with another commitment to the amount of the input; by offsetting these commitments homomorphically and carefully choosing commitment randomness, the prover can show that a particular transaction balances.

We show that Triptych produces signatures with competitive performance to other modern linkable ring signatures for limited anonymity set sizes. We note that similar constructions also require linear verification time, meaning that the size of anonymity set used in practice is likely to be limited for performance reasons.

2 Preliminaries

2.1 Public parameters

Let $\mathbb{G}$ be a cyclic group in which the discrete logarithm problem is hard, and let $\mathbb{F}$ be the scalar field of $\mathbb{G}$. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{F}$ be a cryptographic hash function. Let G and H be generators of $\mathbb{G}$ with unknown discrete logarithm relationship. Let $N = n^m$ be a size parameter, where $n > 1$ and $m > 1$. Let $\{G_{j,i}\}_{j,i=0}^{m-1,n-1}$ be a set of generators of $\mathbb{G}$ with unknown discrete logarithm relationship to each other, to G, and to H. Let U be a generator of $\mathbb{G}$. Note that all generators may be produced using public randomness; for example, the use of a suitable hash function with domain separation may be appropriate. All such public parameters are assumed to comprise a global reference string known to all players; in particular, we exclude them from algorithm definitions and Fiat-Shamir transcript hashes for readability.

2.2 Pedersen commitment

Let Com be a homomorphic commitment scheme that is perfectly hiding and computationally binding. In this work, we assume use of the Pedersen commitment scheme: for $x, r \in \mathbb{F}$, define $\mathrm{Com}(x, r) \equiv xG + rH$

to be the commitment of the value x with randomness r. This can be trivially extended to support matrix values; for $\{x_{j,i}\}, r \in \mathbb{F}$, define $\text{Com}(x, r) \equiv rH + \sum_{j,i} x_{j,i} G_{j,i}$. Note in particular that Pedersen matrix commitments are similarly homomorphic.

2.3 Other notation

For integers or field elements i, j, the Kronecker delta function $\delta(i, j)$ evaluates to 1 if $i = j$ and 0 otherwise, where the output is taken to be in the appropriate set.

We sometimes use index subscript notation of the form i_j to indicate the j digit of i, where such a decomposition of i is taken base n with padded length m:

$$\sum_{j=0}^{m} i_j n^j = i$$

This notation will be specified explicitly where confusion may occur.

3 Protocol: linkable one-of-many commitment

We wish to build a linkable ring signature construction, where a signer who knows the opening of a commitment may sign messages using an anonymity set containing other commitments for which the signer does not know openings. Included with the proof of knowledge, the signer also provides a linking tag that is the image of the signing commitment's opening under a verifiable pseudorandom function, using the method of [6] that has previously appeared in [15, 24]. Part of the soundness of the proving system relies on the proper construction of this linking tag. Upon receipt, a verifier can check whether the linking tag has previously appeared in any other valid proof; if it has not, injectivity assures the verifier that no other signature has been produced by the (unknown) signer.

More specifically, we modify the construction of Bootle [3], which itself is a generalization of a construction by Groth [12]. We produce a sigma protocol for the following relation:

$$\mathcal{R}_{\text{link}} = \left\{ \{M_i\}_{i=0}^{N-1} \subset \mathbb{G}, J \in \mathbb{G}; (l \in \mathbb{Z}, r \in \mathbb{F}) : M_l = rG \text{ and } U = rJ \right\}$$

Figures 1 and 2 describe the protocol.

Observe that this protocol can be made non-interactive using the Fiat-Shamir heuristic, where the verifier challenge is produced using a collision-resistant hash function (modeling a random oracle) and the proof transcript [7].

We will show that the sigma protocol is complete, sound, and zero-knowledge, the precise definitions of which are common and found in [12]. Informally, we require the protocol be:

- *Perfectly complete*: Given knowledge of a witness to a statement in the proof relation, an honest prover can always convince an honest verifier of the validity of the witness.
- *Special sound*: Given a statement in the proof relation, if a prover can answer multiple honest verifier challenges correctly, then it is possible to extract a witness for this statement.
- *Special honest-verifier zero knowledge*: Given any statement and verifier challenge, it is possible to simulate a transcript that is accepted by an honest verifier without knowledge of a corresponding witness.

Theorem 1. *The protocol in Figures 1 and 2 is perfectly complete, special honest-verifier zero knowledge, and $(m + 1)$-special sound.*

Proof. The proof follows similarly to that of [3].

We first show perfect completeness. Suppose the verifier receives a proof generated by an honest prover. Equation 1 holds using the identity

$$\sum_{i=0}^{n-1} \sigma_{j,i} = 1$$

$\underline{\mathcal{P}_{\text{link}}(\{M_i\}, J; (l, r)) :}$

- Select random $r_A \in \mathbb{F}$ and $\{a_{j,i}\}_{i=1,j=0}^{n-1,m-1} \subset \mathbb{F}$. Define $A \equiv \text{Com}(a, r_A)$.
- Define $\{\sigma_{j,i}\}_{i,j=0}^{n-1,j-1} \subset \mathbb{F}$ such that $\sigma_{j,i} \equiv \delta(l_j, i)$ (using our decomposition notation), and choose random $r_B \in \mathbb{F}$. Define $B \equiv \text{Com}(\sigma, r_B)$.
- Select random $r_C \in \mathbb{F}$, and define $C \equiv \text{Com}(a(1 - 2\sigma), r_C)$.
- Select random $r_D \in \mathbb{F}$, and define $D \equiv \text{Com}(-a^2, r_D)$.
- Define coefficients $\{p_{k,j}\}_{k,j=0}^{N-1,m-1}$ such that

$$p_k(x) \equiv \prod_{j=0}^{m-1} (\sigma_{j,k} x + a_{j,k}) = \delta(l, k) x^m + \sum_{j=0}^{m-1} p_{k,j} x^j$$

  for all $k \in [0, N)$ (using our decomposition notation).
- Select random $\{\rho_j\}_{j=0}^{m-1} \subset \mathbb{F}$.
- Define $\{X_j\}_{j=0}^{m-1} \subset \mathbb{G}$ such that:

$$X_j \equiv \sum_{k=0}^{N-1} p_{k,j} M_k + \rho_j G$$

- Define $\{Y_j\}_{j=0}^{m-1} \subset \mathbb{G}$ such that:

$$Y_j \equiv U \sum_{k=0}^{N-1} p_{k,j} + \rho_j J$$

$\underline{\mathcal{P} \to \mathcal{V} :}$
$A, B, C, D, \{X_j\}, \{Y_j\}$

$\underline{\mathcal{V} \to \mathcal{P} :}$
$\xi \in \{0, 1\}^*$

$\underline{\mathcal{P}(\xi) :}$

- Define $\{f_{j,i}\}_{i=1,j}^{n-1,m-1}$ such that $f_{j,i} \equiv \sigma_{j,i} \xi + a_{j,i}$.
- Define $z_A \equiv r_A + \xi r_B$ and $z_C \equiv \xi r_C + r_D$.
- Define $z \equiv r\xi^m - \sum_{j=0}^{m-1} \rho_j \xi^j$.

$\underline{\mathcal{P} \to \mathcal{V} :}$
$\{f_{j,i}\}_{j=0,i=1}^{m-1,n-1}, z_A, z_C, z$

Figure 1: Sigma protocol for $\mathcal{R}_{\text{link}}$

$\underline{\mathcal{V}_{\text{link}}(\{M_i\}, J):}$

- For $0 \leq j < m$, let $f_{j,0} \equiv \xi - \sum_{i=1}^{n-1} f_{j,i}$.
- Accept if and only if:

$$A + \xi B = \text{Com}(f, z_A) \quad (1)$$

$$\xi C + D = \text{Com}(f(\xi - f), z_C) \quad (2)$$

$$\sum_{k=0}^{N-1} M_k \left(\prod_{j=0}^{m-1} f_{j,k_j} \right) - \sum_{j=0}^{m-1} \xi^j X_j - zG = 0 \quad (3)$$

$$U \sum_{k=0}^{N-1} \left(\prod_{j=0}^{m-1} f_{j,k_j} \right) - \sum_{j=0}^{m-1} \xi^j Y_j - zJ = 0 \quad (4)$$

Figure 2: Sigma protocol for $\mathcal{R}_{\text{link}}$ (continued)

for all $0 \leq j < m$. Equation 2 follows similarly, using the identity

$$(\sigma_{j,i})^2 = \sigma_{j,i}$$

for all $0 \leq j < m$. To show Equation 3 holds:

$$\begin{aligned}
& \sum_{k=0}^{N-1} M_k \left(\prod_{j=0}^{m-1} f_{j,k_j} \right) - \sum_{j=0}^{m-1} \xi^j X_j - zG \\
= & \sum_{k=0}^{N-1} M_k p_k(\xi) - \sum_{j=0}^{m-1} \xi^j \left(\sum_{k=0}^{N-1} p_{k,j} M_k + \rho_j G \right) - zG \\
= & \sum_{k=0}^{N-1} M_k \left(p_k(\xi) - \sum_{j=0}^{m-1} \xi^j p_{k,j} \right) - \sum_{j=0}^{m-1} \xi^j \rho_j G - zG \\
= & \sum_{k=0}^{N-1} M_k \xi^m \delta(l,k) - \sum_{j=0}^{m-1} \xi^j \rho_j G - \left(r\xi^m - \sum_{j=0}^{m-1} \rho_j \xi^j \right) G \\
= & \xi^m r G - \sum_{j=0}^{m-1} \xi^j \rho_j G - \xi^m r G + \sum_{j=0}^{m-1} \xi^j \rho_j G \\
= & 0
\end{aligned}$$

Equation 4 follows similarly:

$$\begin{aligned}
& U\sum_{k=0}^{N-1}\left(\prod_{j=0}^{m-1} f_{j,k_j}\right) - \sum_{j=0}^{m-1}\xi^j Y_j - zJ \\
= \;& U\sum_{k=0}^{N-1} p_k(\xi) - \sum_{j=0}^{m-1}\xi^j\left(U\sum_{k=0}^{N-1} p_{k,j} + \rho_j J\right) - zJ \\
= \;& U\sum_{k=0}^{N-1}\left(p_k(\xi) - \sum_{j=0}^{m-1}\xi^j p_{k,j}\right) - \sum_{j=0}^{m-1}\xi^j\rho_j J - zJ \\
= \;& U\sum_{k=0}^{N-1}\xi^m\delta(l,k) - \sum_{j=0}^{m-1}\xi^j\rho_j J - \left(r\xi^m - \sum_{j=0}^{m-1}\rho_j\xi^j\right)J \\
= \;& \xi^m U - \sum_{j=0}^{m-1}\xi^j\rho_j G - \xi^m r J + \sum_{j=0}^{m-1}\xi^j\rho_j G \\
= \;& 0
\end{aligned}$$

since $J = r^{-1}U$ in a valid proof. Hence the protocol is perfectly complete.

We next show that the protocol is special honest-verifier zero knowledge. To do so, we construct a simulator that, given a random verifier challenge ξ, can construct a proof transcript with identical distribution to a valid proof.

First, observe that the simulator presented in the proof of Lemma 1 in [3] translates identically to our setting. If the simulator chooses $B \in \mathbb{G}$ uniformly at random, the cited lemma assures us a valid simulation of the proof elements $A, C, D, z_A, z_C, \{f^{(u)}_{j,i\neq 0}\}$; we may compute each $f^{(u)}_{j,0}$ from this. Further, in a valid proof, B is independent and uniformly distributed as well.

The proof elements $\{X_j\}_{j=1}^{m-1}$ and $\{Y_j\}_{j=1}^{m-1}$ are independent and uniformly distributed in a valid proof since the set $\{\rho_j\}$ is random and the discrete logarithm problem in $\mathbb{G}$ is hard, so the simulator may choose these uniformly at random. The verification checks require that X_0 and Y_0 be uniquely determined by the other elements in the corresponding sets in both real proofs and by the simulator.

Finally, z is uniformly distributed in valid proofs given random ξ, so the simulator may choose it uniformly at random. Hence the construction is special honest verifier zero-knowledge.

It remains to show that the protocol is $(m+1)$-special sound, where $m > 1$. To show this, we construct an extractor that, given $m+1$ valid responses to $m+1$ distinct verifier challenges for the same initial statement, produces a valid witness.

Suppose that for a given statement, we have a set of $m+1$ distinct verifier challenges $\{\xi_e\}_{e=0}^{m}$ corresponding to unique valid responses of this form:

$$\left\{\{f^{(e)}_{j,i}\}, \{z_e\}\right\}_{e=0}^{m}$$

From the 3-special soundness in [3] and $m > 1$ we have valid extractions $\{\sigma_{j,i}\}_{j,i=0}^{m-1,n-1}$ and $\{a_{j,i}\}_{j,i=0}^{m-1,n-1}$, and the Pedersen binding property ensures that (with high probability) we have:

$$f^{(e)}_{j,i} = \sigma_{j,i}\xi_e + a_{j,i}$$

for all $e \in [0, m]$. Using the extracted values, compute

$$p_k(\xi) \equiv \prod_{j=0}^{m-1}\left(\sigma_{j,k}\xi + a_{j,k}\right)$$

for all $k \in [0, N)$. Extraction of $\{\sigma_{j,i}\}_{j,i=0}^{m-1,n-1}$ immediately yields the signing index l.

We have seen that p_k is of degree m only when $k = l$. Hence there exist coefficients $\{\overline{X}_j, \overline{Y}_j\}_{j=0}^{m-1}$, computed uniquely from the statement and extracted values, such that Equations 3 and 4 are of the following

form:

$$\begin{aligned} \xi^m M_l + \sum_{j=0}^{m-1} \xi^j \overline{X}_j &= zG \\ \xi^m U + \sum_{j=0}^{m-1} \xi^j \overline{Y}_j &= zJ \end{aligned}$$

Construct a Vandermonde matrix V where the e row is the vector $(1, \xi_e, \ldots, \xi_e^m)$. Since all ξ_e are distinct, the rows of V span $\mathbb{F}^{m+1}$; hence there exist weights $\{\theta_e\}_{e=0}^m$ such that the resulting linear combination of rows produces the vector $(0, \ldots, 0, 1)$. That is, $\sum_{e=0}^m \theta_e \xi_e^j = \delta(j, m)$.

For each of the previous two equations, we can therefore build a linear combination over e. For the first:

$$M_l = \sum_{e=0}^m \theta_e \xi_e^m M_l + \sum_{e=0}^m \theta_e \left(\sum_{j=0}^{m-1} \xi_e^j \overline{X}_j \right) = \left(\sum_{e=0}^m \theta_e z_e \right) G$$

Hence we extract $r \equiv \sum_{e=0}^m \theta_e z_e$. For the second:

$$U = \sum_{e=0}^m \theta_e \xi_e^m U + \sum_{e=0}^m \theta_e \left(\sum_{j=0}^{m-1} \xi_e^j \overline{X}_j \right) = \left(\sum_{e=0}^m \theta_e z_e \right) J$$

This implies that $rJ = U$, as required. Hence the protocol is $(m+1)$-special sound, which completes the proof. □

4 Security: linkable ring signature

Informally, a linkable ring signature is a construction permitting signatures on messages using a signer-selected anonymity set (called a *ring*) of possible signers. A valid signature convinces a verifier that the signer knows (at least) one of the private keys to a ring member. The construction is linkable if it is possible to determine whether two signatures were generated using the same private key, regardless of the ring members used.

We use the security definitions in [12] as a starting point, directly adopting definitions for correctness and unforgeability that also appear in more recent work like [1]. However, we modify the definition of anonymity to account for linking tags, such that the adversary is required to differentiate between at least two possible honest signers that the adversary has not corrupted. To account for the linking properties desired in our construction, we use the clever linkability definition from [1], which uses a set-theoretic approach. We use a straightforward definition for non-frameability, where the adversary produces a target signature on an honest key after receiving signing and corruption oracle access, and must then produce a new signature that links.

More formally, a *linkable ring signature* (LRS) construction is a set of algorithms KeyGen, Sign, Verify, and Link satisfying certain properties. A set of public parameters is assumed to be available to each algorithm.

- KeyGen$(r) \to (x, X)$: Generates a secret key x and corresponding public key X, optionally using randomness r; if not specified, the secret key is sampled uniformly at random.
- Sign$(x, M, R) \to \sigma$: Generates a signature σ on a message $M \in \{0,1\}^*$ with respect to the ring $R = \{X_1, \ldots, X_n\}$, provided that x is a secret key corresponding to some $X_i \in R$ generated by KeyGen.
- Verify$(\sigma, M, R) \to \{0,1\}$: Verifies a signature σ on a message M with respect to the ring R. Outputs 0 is the signature is rejected, and 1 if accepted.

- $\text{Link}(\sigma, \sigma') \to \{0,1\}$: Determines if signatures σ and σ' were signed using the same private key. Outputs 0 if the signatures were signed using different private keys, and 1 if they were signed using the same private key.

We require that an LRS have the properties of correctness, anonymity, unforgeability, linkability, and non-frameability.

Correctness requires that a signature generated honestly will always verify.

Definition 1 (Correctness)**.** Consider this game between a challenger and a probabilistic polynomial-time adversary $\mathcal{A}$:

- The challenger runs $\text{KeyGen} \to (x, X)$ and supplies the keys to $\mathcal{A}$.
- The adversary $\mathcal{A}$ chooses a ring such that $X \in R$ and a message $M \in \{0,1\}^*$, and sends them to the challenger.
- The challenger signs the message with $\text{Sign}(x, M, R) \to \sigma$.

If $\Pr[\text{Verify}(\sigma, M, R) = 1] = 1$, we say that the LRS is *perfectly correct*.

Note that we do not require any ring members (except for X) to have been generated by KeyGen. However, distributed applications may in practice place additional restrictions on public keys used in anonymity sets. This allows for the possibility that $\mathcal{A}$ maliciously chooses ring members.

Unforgeability requires that an adversary who does not control the private key to a ring member cannot generate a valid signature on any message using that ring.

Definition 2 (Unforgeability)**.** Consider this game between a challenger and a probabilistic polynomial-time adversary $\mathcal{A}$:

- The adversary $\mathcal{A}$ is granted access to a public-key oracle GenOracle that (on the i^{th} invocation) runs $\text{KeyGen} \to (x_i, X_i)$ and returns X_i to $\mathcal{A}$.
- The adversary $\mathcal{A}$ is granted access to a corruption oracle $\text{CorruptOracle}(i)$ that returns x_i if it corresponds to a query to GenOracle.
- The adversary $\mathcal{A}$ is granted access to a signing oracle $\text{SignOracle}(X, M, R)$ that runs $\text{Sign}(x, M, R) \to \sigma$ and returns σ to $\mathcal{A}$, provided that X corresponds to a query to GenOracle and $X \in R$.
- Then, $\mathcal{A}$ outputs (σ, M, R) such that SignOracle was not queried with $(-, M, R)$, all keys in R were generated by queries to GenOracle, and no key in R was corrupted by CorruptOracle.

If $\Pr[\text{Verify}(\sigma, M, R) = 1] \approx 0$, we say that the LRS is *unforgeable with respect to insider corruption*.

Anonymity requires that as long as a ring contains at least two members that have not been corrupted, an adversary can do no better than guessing at determining the signer of an honest signature.

Definition 3 (Anonymity)**.** Consider this game between a challenger and a probabilistic polynomial-time adversary $\mathcal{A}$:

- The adversary $\mathcal{A}$ is granted access to the public-key oracle GenOracle and the corruption oracle CorruptOracle.
- The adversary $\mathcal{A}$ chooses a message $M \in \{0,1\}^*$, a ring R, and indices i_0 and i_1, and sends them to the challenger. We require that $X_{i_0}, X_{i_1} \in R$ such that both keys were generated by queries to GenOracle, and neither key was queried to CorruptOracle.
- The challenger selects a uniformly random bit $b \in \{0,1\}$, generates the signature $\text{Sign}(x_{i_b}, M, R) \to \sigma$, and sends it to $\mathcal{A}$.
- The adversary $\mathcal{A}$ chooses a bit $b' \in \{0,1\}$.

If $\Pr[b' = b] \approx 1/2$ and $\mathcal{A}$ did not make any corruption queries after receiving the challenge bit, we say that the LRS is *anonymous*.

We observe that this definition permits the adversary to have corrupted or maliciously generated all but two keys in the ring. Some definitions allow the adversary to corrupt more keys, but we will see that this is inconsistent with our linkability construction, where an adversary in control of a ring member's private key can trivially determine if it was the signer by examining the linking tag associated to a signature.

Linkability requires that an adversary be unable to produce $k+1$ non-linked signatures on a combined anonymity set of k public keys.

Definition 4 (Linkability)**.** Consider the following game between a challenger and a probabilistic polynomial-time adversary $\mathcal{A}$:

- For $i \in [0, k-1]$, the adversary $\mathcal{A}$ produces a public key X_i, message M_i, ring R_i, and signature σ_i.
- The adversary $\mathcal{A}$ produces another message M, ring R, and signature σ.
- All tuples $(X_i, M_i, R_i, \sigma_i)$ and (M, R, σ) are sent to the challenger.
- The challenger checks the following:
  - $|V| = k$, where $V \equiv \bigcup_{i=0}^{k-1} R_i$.
  - Each $X_i \in V$.
  - Each $R_i \subset V$.
  - $\text{Verify}(\sigma_i, M_i, R_i) = 1$ for all i.
  - $\text{Verify}(\sigma, M, R) = 1$.
  - For all $i \neq j$, we have $\text{Link}(\sigma_i, \sigma_j) = \text{Link}(\sigma_i, \sigma) = 0$.
- If all checks pass, $\mathcal{A}$ wins.

If $\mathcal{A}$ wins with only negligible probability for all k, we say the LRS is *linkable*.

Non-frameability requires that an adversary be unable to generate a signature that links with an honest signature.

Definition 5 (Non-frameability)**.** Consider also the following game between a challenger and a probabilistic polynomial-time adversary $\mathcal{A}$:

- The adversary $\mathcal{A}$ is granted access to the public-key oracle GenOracle.
- The adversary $\mathcal{A}$ is granted access to the corruption oracle CorruptOracle.
- The adversary $\mathcal{A}$ is granted access to the signing oracle SignOracle.
- The adversary $\mathcal{A}$ chooses a public key X that was generated by a query to GenOracle, but was not presented as a query to CorruptOracle. It selects a message $M \in \{0,1\}^*$ and ring R such that $X \in R$. It queries $\text{SignOracle}(X, M, R) \to \sigma$.
- The adversary $\mathcal{A}$ then produces a tuple (M', R', σ') and sends (M', R', σ') to the challenger, along with (X, M, R, σ).
- If $\text{Verify}(\sigma', M', R') = 0$ or if σ' was produced using a query to SignOracle, the challenger aborts.

If $\Pr[\text{Link}(\sigma, \sigma') = 1 \approx 0$, we say that the LRS is *non-frameable*.

KeyGen(r) :

- If not specified, select $r \in \mathbb{F}$ uniformly at random.
- Compute $R = rG$.
- Return $(x, X) = (r, R)$.

Sign(x, M, R) :

- Let $R = \{X_0, \ldots, X_{N-1}\}$ such that $X_l = x_l G$.
- Compute $J \equiv x_l^{-1} U$.
- Run $\mathcal{P}_{\text{link}}(R, J; (l, x_l)) \to a$ (up to the verifier challenge).
- Set $\xi \equiv \mathcal{H}(M, R, a)$.
- Run $\mathcal{P}_{\text{link}}(\xi) \to z$ (after the verifier challenge).
- Return $\sigma = (a, z, J)$.

Verify(σ, M, R) :

- Let $R = \{X_0, \ldots, X_{N-1}\}$ such that $X_l - x_l G$.
- Let $\sigma = (a, z, J)$.
- Set $\xi \equiv \mathcal{H}(M, R, a)$.
- Return $\mathcal{V}_{\text{link}}(R, J, a, z)$.

Link(σ, σ') :

- We implicitly assume that σ and σ' have been previously verified.
- Let $\sigma = (a, z, J)$ and $\sigma' = (a', z', J')$.
- If $J = J'$, return 1. Otherwise, return 0.

Figure 3: Linkable ring signature using $\mathcal{R}_{\text{link}}$

5 Application: linkable ring signature

The constructions in [12, 3] describe how to use a similar sigma protocol to construct a simple ring signature scheme. Using our modifications, we can easily extend this to account for linkability and non-frameability. We briefly show how to do so.

Theorem 2. *The protocol in Figure 3 is a linkable ring signature construction.*

Proof. Perfect correctness follows immediately from the perfect completeness of the proving system for $\mathcal{R}_{link}$.

Similarly, anonymity follows since the proving system is special honest-verifier zero knowledge, and therefore witness indistinguishable [5]. Any adversarial advantage in breaking anonymity must therefore arise from distinguishing either input commitments or linking tags in signatures. Since honestly-generated input Pedersen commitments are perfectly hiding, they are indistinguishable from elements of $\mathbb{G}$ selected uniformly at random; we assume by definition that at least two such commitments are present in such a signature. Further, honestly-generated linking tags are generated from a one-way pseudorandom function, and therefore in the random oracle model are independently uniformly distributed from other proof elements and input commitments.

The proof for unforgeability in [12] relies on the (special) soundness of the underlying sigma protocol; it applies directly to our modification for $\mathcal{R}_{link}$, and is not repeated here.

To show linkability, observe first that Link simply compares linking tags, so two signatures link if and only if they share a common linking tag. Suppose an adversary can win the linkability game with non-negligible probability for some $k > 1$. Since all provided signatures verify, soundness implies extraction of a witness x_i from signature σ_i for all i, and of a witness x from σ. Note that all $\{x_i\}$ and x are distinct. If $x_i = x_j$ for $i \neq j$, then the corresponding linking tags J_i and J_j are such that $x_i J_i = x_j J_j = U$; then $J_i = J_j$, which contradicts $\text{Link}(\sigma_i, \sigma_j) = 0$. The same reasoning holds to show x is similarly distinct. Soundness also implies that for all i, there exists $X_i \in R_i$ such that $x_i G = X_i$; similarly, there exists $X \in R$ such that $xG = X$. By assumption we have

$$\{X_0, \ldots, X_{k-1}, X\} \subset \left(\bigcup_{i=0}^{k-1} R_i\right) \cup R \subset V.$$

However, note that $|\{X_0, \ldots, X_{k-1}, X\}| = k + 1$, but that $|V| = k$, a contradiction.

Finally, we show non-frameability and assume an adversary has a non-negligible advantage in breaking this property. Because we have $\text{Verify}(\sigma', M', R') = 1$, soundness implies extraction of a witness $x' \in \mathbb{F}$ such that $x'G \in R'$; we also have a witness x such that $xG \in R$ from the known signature σ. Since $\text{Link}(\sigma, \sigma') = 1$, the corresponding linking tags J and J' are equal by definition; hence by soundness $xJ = x'J' = U$, giving $x = x'$. However, the adversary did not query CorruptOracle with X, meaning it breaks the discrete logarithm problem non-negligibly. □

6 Protocol: parallel linkable one-of-many commitment

In this section, we describe a modification of the sigma protocol for $\mathcal{R}_{\text{link}}$ that permits us to prove knowledge of multiple commitments in separate sets at the same index position, while retaining the linking property in the first commitment set only. We later show how to apply such a construction to a signer-ambiguous transaction protocol that can demonstrate balance preservation.

We wish to produce a sigma protocol for the following relation:

$$\mathcal{R}_{\text{par}} = \left\{ \{M_i\}_{i=0}^{N-1} \subset \mathbb{G}, \{P_i\}_{i=0}^{N-1} \subset \mathbb{G}, J \in \mathbb{G}; (l \in \mathbb{Z}, r \in \mathbb{F}, s \in \mathbb{F}) : M_l = rG \text{ and } P_l = sG \text{ and } U = rJ \right\}$$

This requires only minor modifications to the protocol for $\mathcal{R}_{\text{link}}$, so we document only the modified proof elements constructed and verified in Figure 4. All other proof elements are generated and verified identically.

Theorem 3. *The protocol in Figure 4 is perfectly complete, special honest-verifier zero knowledge, and* $(m + 1)$*-special sound.*

$\underline{\mathcal{P}_{\mathrm{par}}(\{M_i\},\{P_i\},J;(l,r,s)):}$

- Define $K \equiv sJ$.
- Define $\mu \equiv \mathcal{H}(\{M_i\},\{P_i\},J,K)$.
- Define $\{X_j\}_{j=0}^{m-1} \subset \mathbb{G}$ such that:

$$X_j \equiv \sum_{k=0}^{N-1} p_{k,j}(M_k + \mu P_k) + \rho_j G$$

- Define $\{Y_j\}_{j=0}^{m-1} \subset \mathbb{G}$ such that:

$$Y_j \equiv (U + \mu K) \sum_{k=0}^{N-1} p_{k,j} + \rho_j G$$

$\underline{\mathcal{P} \to \mathcal{V}:}$
$K, \{X_j\}, \{Y_j\}$

$\underline{\mathcal{V} \to \mathcal{P}:}$
$\xi \in \{0,1\}^*$

$\underline{\mathcal{P}(\xi):}$

- Define $z \equiv (r + \mu s)\xi^m - \sum_{j=0}^{m-1} \rho_j \xi^j$.

$\underline{\mathcal{P} \to \mathcal{V}:}$
z

$\underline{\mathcal{V}_{\mathrm{par}}(\{M_i\},\{P_i\},J):}$

- Define $\mu \equiv \mathcal{H}(\{M_i\},\{P_i\},J,K)$.
- Accept if and only if:

$$\begin{aligned} \sum_{k=0}^{N-1}(M_k + \mu P_k)\left(\prod_{j=0}^{m-1} f_{j,k_j}\right) - \sum_{j=0}^{m-1} \xi^j X_j - zG &= 0 \\ (U + \mu K)\sum_{k=0}^{N-1}\left(\prod_{j=0}^{m-1} f_{j,k_j}\right) - \sum_{j=0}^{m-1} \xi^j Y_j - zJ &= 0 \end{aligned}$$

Figure 4: Sigma protocol (abbreviated) for $\mathcal{R}_{\mathrm{par}}$

Proof. In the random oracle model, extraction of a witness of the form $r + \mu s$ implies knowledge of both r and s such that $rG = M_l$ and $sG = P_l$, similarly to the key-aggregation arguments in [17]. The same extraction shows that $(r + \mu s)J = U + \mu K$, which implies in particular that $rJ = U$, as required.

The rest of the proof follows with only trivial modifications from the proof for $\mathcal{R}_{\text{link}}$. □

7 Application: signer-ambiguous transaction protocol

The parallel construction described in Figure 4 can be used in a signer-ambiguous transaction protocol.

Suppose a user wishes to generate a transaction consuming W previously-generated outputs and generating T fresh outputs. The user shuffles the consumed outputs within a larger list of N outputs $\{M_k\}_{k=0}^{N-1}$, such that there exist indices $\{l_u\}_{u=0}^{W-1}$ where each $M_{l_u} = r_u$ for some known private key r_u. Further, assume each M_{l_u} comes equipped with an amount commitment of the form $P_{l_u} \equiv \text{Com}(a_u, s_u)$ for amount a_u and mask s_u. (All other M_k also come equipped with a corresponding P_k, but the structure of these points is not relevant here.)

The user generates W auxiliary commitments $P'_u \equiv \text{Com}(a_u, s'_u)$ to the same amounts, but with different masks $\{s'_u\}$ chosen uniformly at random from $\mathbb{F}$. Then, the user generates W spend proofs, each using the following prover inputs for $u \in [0, W)$:

$$\mathcal{P}_{par}(\{M_i\}, \{P_i - P'_u\}, r_u^{-1}U; (l_u, r_u, s_u - s'_u))$$

For $j \in [0, T)$, the user generates a fresh output of the form $Q_j \equiv \text{Com}(b_j, t_j)$ for amount b_j and mask t_j. The masks are chosen such that for $j \in [1, T)$, we have t_j chosen uniformly at random from $\mathbb{F}$. We then choose

$$t_0 \equiv \sum_{u=0}^{W-1} s'_u - \sum_{j=1}^{t-1} t_j$$

and include all $\{P'_u\}$ auxiliary commitments in the transaction.

To verify such a transaction, the verifier first performs verification on each spend proof to ensure it is valid. Then, the verifier ensures that

$$\sum_{u=0}^{W-1} P'_u - \sum_{j=0}^{T-1} Q_j = 0$$

such that the transaction balances. This succeeds since the commitments sum to zero if and only if the difference of input and output amounts is zero, which holds since the Pedersen commitment construction is computationally binding.

8 Efficiency

Triptych proofs scale logarithmically with the size of the input anonymity set; this is the best asymptotic scaling known for ring signatures that do not require a trusted setup process over non-pairing groups. Related protocols based on the inner-product compression method of [4] include Omniring [15] and RingCT 3.0 [15]. However, it is challenging to directly compare these protocols' efficiency. Omniring includes all transaction input signatures, output range proofs, and balance within a single proof structure; however, it is not possible to verify a batch of proofs more efficiently by combining common generators. While an early version of RingCT 3.0 used separate input, range, and balance proofs, the most recent version merges all input and balance proofs together but outsources the range proofs to an efficient construction like [4]; this comes at the cost of requiring that the number of inputs be a power of two or otherwise carefully padded. A more direct comparison is to CLSAG, a linear-sized linkable ring signature construction.

We compare size and verification of parallel Triptych, the earlier version of RingCT 3.0 considering only the signature proof component, and CLSAG. For verification scaling, we account for the use of batching in Triptych and RingCT 3.0, where generators common to multiple proofs are used only once in verification. Further, since verification in both of these constructions reduces to checking whether several multiscalar multiplications are zero, we may apply random weighting such that verifying a batch of multiple proofs

reduces to a single multiscalar multiplication. The use of efficient multiscalar multiplication algorithms like [23, 20] means that an n-multiscalar multiplication evaluates as $O(n/\log n)$. This form of batching does not apply to CLSAG, where verification consists of a sequence of hash function evaluations.

Table 1 compares proof/signature sizes and verification complexity for these constructions. Verification complexity is separated into the number of hash-to-$\mathbb{F}$ operations (denoted $\mathcal{H}$), hash-to-$\mathbb{G}$ operations (denoted $\mathbb{H}$), and i-multiscalar multiplication operations of size $k(i)$. Figure 5 examines size as a function of input anonymity size under the assumption that elements of both $\mathbb{G}$ and $\mathbb{F}$ occupy 32 kB of space. We note that although RingCT 3.0 proofs are smaller for large N, Triptych proofs are smaller in the range $16 < N < 1024$, a reasonable range given practical verification times.

	Size ($\mathbb{G}$)	Size ($\mathbb{F}$)	Verification
CLSAG [10]	1	$N+1$	$(N+2)\mathcal{H} + N\mathbb{H} + 2Nk(3)$
RingCT 3.0 [24]	$2\lg(N)+8$	10	$k(4N+2\lg(N)+16)$
Triptych (this work)	$2\lg(N)+5$	$\lg(N)+3$	$k(2N+4\lg(N)+8)$

Table 1: Proof sizes and verification complexity, for anonymity set size N

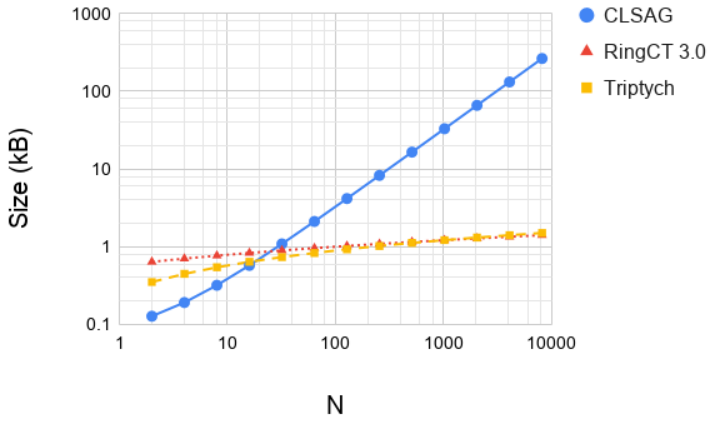

Figure 5: Proof sizes for input anonymity set size N

9 Future work

It is possible to further extend the Triptych proving system to support proving knowledge of openings of multiple commitments within the *same* anonymity set, while permitting the construction of linking tags for each such opening and demonstrating balance directly within a single proof. Such a construction is much more efficient than the present work when built into a transaction protocol, but the security definitions applying to such a construction are still under evaluation.

References

[1] Michael Backes, Nico Döttling, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Ring signatures: Logarithmic-size, no setup—from standard assumptions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 281–311, Cham, 2019. Springer International Publishing.

[2] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 60–79, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[3] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, pages 243–265, Cham, 2015. Springer International Publishing.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334. IEEE, 2018.

[5] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 174–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[6] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005*, pages 416–431, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[7] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[8] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 393–415, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[9] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 181–200, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[10] Brandon Goodell, Sarang Noether, and Arthur Blue. Compact linkable ring signatures and applications. Cryptology ePrint Archive, Report 2019/654, 2019. `https://eprint.iacr.org/2019/654`.

[11] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[12] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 253–280, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[13] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification. *Tech. rep. 2016–1.10. Zerocoin Electric Coin Company, Tech. Rep.*, 2016.

[14] Aram Jivanyan. Lelantus: Towards confidentiality and anonymity of blockchain transactions from standard assumptions. Cryptology ePrint Archive, Report 2019/373, 2019. `https://eprint.iacr.org/2019/373`.

[15] Russell W. F. Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling up private payments without trusted setup - formal foundations and constructions of ring confidential transactions with log-size proofs. Cryptology ePrint Archive, Report 2019/580, 2019. `https://eprint.iacr.org/2019/580`.

[16] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[17] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple Schnorr multi-signatures with applications to Bitcoin. *Designs, Codes and Cryptography*, 87(9):2139–2164, Sep 2019.

[18] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411. IEEE, 2013.

[19] Shen Noether, Adam Mackenzie, and the Monero Research Lab. Ring confidential transactions. *Ledger*, 1(0):1–18, 2016.

[20] Nicholas Pippenger. On the evaluation of powers and monomials. *SIAM Journal on Computing*, 9(2):230–250, 1980.

[21] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[22] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, Jan 1991.

[23] Ernst G Straus. Addition chains of vectors (problem 5125). *American Mathematical Monthly*, 70(806-808):16, 1964.

[24] Tsz Hon Yuen, Shi-feng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, and Dawu Gu. RingCT 3.0 for blockchain confidential transaction: Shorter size and stronger security. Cryptology ePrint Archive, Report 2019/508, 2019. `https://eprint.iacr.org/2019/508`.