FCMP++ review

Cypher Stack[*]

June 27, 2024

This report describes a full-chain membership proof construction and provides a relevant security analysis. As with any such report, it may contain errors and cannot guarantee correctness or security. Further, it cannot guarantee that any particular implementation of the construction is correct, secure, or suitable for intended use cases.

The author asserts no warranty and disclaims liability for its use. The author further expresses no endorsement of any kind. This report has not undergone any further formal or peer review.

Contents

[*] https://cypherstack.com

1 Introduction

Full-chain membership proofs are a proposed construction intended for use in privacy-respecting transaction protocols, specifically the Monero protocol. They can be used to replace membership proofs in protocols like Seraphis[1] or as part of linkable ring signature designs for use in protocols like RingCT (as introduced in [14]).

A recent technical note[2] describes a particular design intended to replace CLSAG linkable ring signatures from [12] in the Monero protocol, which uses a RingCT design. This construction is specifically intended to be compatible with outputs generated from the current and previous versions of the protocol, which can increase practical security and reduce engineering risk. In this report, we examine the technical note to determine the extent to which its design meets intended security properties.

The technical note seeks a construction providing the following informal properties:

1. All outputs consumed in a valid transaction were generated in previous valid transactions.
2. No output consumed in a valid transaction was consumed in any previous valid transaction.
3. The transaction was authorized by an entity possessing all signing keys associated to consumed outputs.
4. The sum of values represented by consumed outputs in a valid transaction is equal to the sum of values represented by generated outputs (including any fees).
5. The value represented by each output generated in a valid transaction is within a given range.

Since properties 4 and 5 can be achieved through suitable use of existing balance and range assertions, the technical note presents a relation intended to capture the other properties. It then breaks this relation into two components: one effectively showing property 1 using a membership proof, and the other effectively showing properties 2 and 3 using an authorization and linkability proof.

1.1 Summary

The construction presented in the technical note appears overall to meet its desired goals. However, the technical note was written with implementation in

[1] `https://github.com/UkoeHB/Seraphis/`

[2] `https://github.com/kayabaNerve/fcmp-ringct`

mind, and therefore was not intended to reflect particular formal presentation, notation, or analysis.

This report is intended to provide such formalization, as well as specify particular component security properties that are necessary for analysis. It therefore makes changes to the presentation and structure of the proving systems in the technical note to allow for such analysis. It also provides a suggested optimization and proves it secure.

2 Notation

The technical note introduces notation that is important for analysis. While much of the notation is explained by definition or context, some is not. We review it here for completeness.

Let $\mathbb{G}$ be an elliptic curve group where the discrete logarithm problem is hard, and let $\mathbb{F}$ be its scalar field.

Fix group generators $G, H, T, U, V \in \mathbb{G}$ with no efficiently-computable nontrivial discrete logarithm relation. This means that if a bounded adversary can find $g, h, t, u, v \in \mathbb{F}$ such that

$$0 = gG + hH + tT + uU + vV$$

holds, then $g = h = t = u = v = 0$. When specifying relations instantiated by proving systems, we elide these generators for brevity; they are assumed to be known public parameters.

Transactions consume existing outputs and generate new outputs. Output keys are of the form $O = xG + yT$, where x and y are output signing keys required to authorize a transaction consuming the output.

Output values are represented by Pedersen commitments of the form $C = c_g G + c_h H$, where c_h represents the value and c_g the commitment mask.

Each output consumed in a transaction comes equipped with a linking tag of the form $L = x\mathsf{H}(O)$, where H is a cryptographic hash function with codomain $\mathbb{G}$. Note that without knowledge of x, it is infeasible to identify an output O with its linking tag L. In the technical note, the deterministic notation $I = \mathsf{H}(O)$ is used.

To allow for balance computation without leaking information about consumed outputs, each consumed output comes equipped with a rerandomized commitment $\widetilde{C} = C + r_c G$ for random r_c. This means $\widetilde{C}$ is a commitment to the same value as C.

3 Security properties

The technical note does not specify security properties that should be achieved by proving systems it requires or defines.

Inherent is that each such proving system be perfectly complete, which asserts that an honest verifier will always accept a proof generated by an honest prover.

Another important property is computational witness-extended emulation (CWEE), a generalization of special soundness. This property requires the existence of an extractor, a bounded algorithm that can rewind an accepting proof transcript for a given statement with new randomness at challenge points. The extractor must be able to compute a valid witness for the corresponding statement using the rewound transcript tree, except possibly with negligible probability. It will be important and useful that proving systems be CWEE, as this allows us to reason about relationships between extracted witness values. We therefore require that proving systems under consideration be CWEE.

Also useful is the special honest-verifier zero knowledge (SHVZK) property. This property requires the existence of a simulator, a bounded algorithm that samples verifier randomness on a given statement; the simulator must be able to use this to produce valid transcripts that are identically distributed to those of honestly-generated proofs, but without knowledge of a corresponding witness. It provides a related (but weaker) property, witness indistinguishability, which asserts that an adversary cannot determine the witness used to produce a valid proof. We require that proving systems under consideration be SHVZK.

Finally, a less common property is simulation extractibility. This property, examined in [9], effectively requires that a bounded adversary in possession of valid (possibly simulated) proofs be required to produce an extracted witness for subsequent proofs that it produces. The property is useful in part because it relates to a notion of non-malleability. It holds for certain classes of proving systems, but does not immediately follow from CWEE in general. We therefore do not specifically require this for proving systems under consideration, but consider it beneficial.

Throughout this report, we require the conversion of interactive protocols to non-interactive equivalents using the strong Fiat-Shamir technique, formalized in [1]. This requires the prover and verifier to maintain a transcript of the statement used in the proving relation instance, as well as all elements transmitted from the prover to the verifier. Verifier challenges are produced through the use of a cryptographic hash function evaluation (with suitable domain separation) on the transcript. We stress the importance of maintaining this transcript and computing challenges securely, as failure to do so can result in exploitable loss of intended security properties, as in the examples of [8].

4 Main relation

The technical note seeks an instantiation of a proving system for the following relation for each consumed output in a transaction:

$$\Big\{S \subset \mathbb{G}^3, \widetilde{C}, L \in \mathbb{G}; O, I, C \in \mathbb{G}, x, y, c_g, c_h, r_c \in \mathbb{F} : \\ (O, I, C) \in S, O = xG + yT, L = xI, C = c_gG + c_hH, \widetilde{C} = C + r_cG\Big\} \quad (1)$$

The technical note indicates that witness elements are integers in $\mathbb{Z}$, which is incorrect notation; we correct this here. We also slightly modify the presentation of $\widetilde{C}$ (but not its effective definition) here for clarity.

The technical note specifies that the proving systems it introduces satisfy relation 1, which is not strictly the case. While its membership proving system (which we discuss later) proves that a given group element $\widetilde{C}$ offsets a value C in an ambiguous way, it does not extract an opening (c_g, c_h) of C as a witness in the relation, violating our CWEE requirement.

However, this is not problematic in practice, as the use of a suitable range proving system associated to transactions provides such an opening that must be unique up to the hardness of the discrete logarithm problem. Consumed outputs in the Monero protocol correspond to generated outputs with associated range proofs that use either a variant of Borromean ring signatures [15] or the Bulletproofs [2] or Bulletproofs+ [5] range proving systems.

Both the Bulletproofs and Bulletproofs+ range proving systems are CWEE and SHVZK. Further, work in [10] (for the algebraic group model) and [11] (for the random oracle model) shows that the Bulletproofs design is simulation extractible. The Borromean range proof construction admits an extractor (referred to as a simulator in [15]) as required for CWEE. While it is specified to be "zero knowledge", the definition in [15] is more akin to witness indistinguishability; the corresponding proof of this property is informal, but likely would apply to SHVZK as well.

We therefore modify relation 1 to remove the opening of C, allowing us to perform a more complete and correct analysis that does not rely on externally-provided range proofs:

$$\Big\{ S \subset \mathbb{G}^3, \widetilde{C}, L \in \mathbb{G}; O, I, C \in \mathbb{G}, x, y, r_c \in \mathbb{F} : \\ (O, I, C) \in S, O = xG + yT, L = xI, \widetilde{C} = C + r_c G \Big\} \quad (2)$$

In practice, this achieves the desired security properties.

Property 1, which asserts that a consumed output exist, is trivially satisfied by membership in S, the set of previously-generated outputs.

Property 2 states that a consumed output not have been previously consumed. This follows since the mapping $x \mapsto I$ between output signing keys and linking tags is deterministic.

Property 3 is that a transaction be authorized by a holder of all consumed output signing keys. This can be achieved by instantiating a proving system for the relation in a non-interactive manner that binds transaction details as part of the strong Fiat-Shamir technique.

Property 4 asserts transaction balance. The definitions of C and $\widetilde{C}$ mean that $\widetilde{C}$ is a Pedersen commitment to the same value c_h as C. In the Monero protocol, all commitment masks used in generated outputs are deterministically generated such that they are uniformly distributed at random. All consumed output rerandomized commitment masks are sampled uniformly at random except one, which is chosen such that the difference between generated output commitments and consumed output rerandomized commitments is zero. From this observation, the balance security requirement is achieved. However, we note that this means the set of commitment masks is not independently sampled, which means that knowledge of the sum of all but one such mask leaks the remaining one.

Property 5 states that generated output values are within a valid range. This is achieved by the use of range proofs that accompany each transaction, as discussed.

5 Membership proving system

The technical note specifies a requirement for a proving system for the following relation:

$$\Big\{ S \subset \mathbb{G}^3, \widetilde{O}, \widetilde{I}, \widetilde{C}, R \in \mathbb{G}; O, I, C \in \mathbb{G}, r_o, r_i, r_j, r_c \in \mathbb{F} : \\ (O, I, C) \in S, \widetilde{O} = O + r_o T, \widetilde{I} = I + r_i U, \widetilde{C} = C + r_c G, R = r_i V + r_j T \Big\} \tag{3}$$

It does not immediately instantiate such a proving system, but notes that it does so later using an approach based on [4].

It is unspecified what security requirements a proving system for relation 3 must satisfy. As discussed earlier, we require that such a proving system be complete, CWEE, and SHVZK.

We note that the select-and-rerandomize construction in [4] comes equipped with a proof of an associated binding property that does effectively achieve CWEE through its use of extractors of constituent protocols. Further, the design in [4] also has an associated zero-knowledge property that effectively satisfies SHVZK.

The analysis in [4] does not directly claim or show that its design satisfies simulation extractibility. However, it internally uses a circuit-based design based on the Bulletproofs arithmetic circuit satisfiability proving system in [2], albeit using modifications[3] required to handle vector commitments. Work in [10, 11, 7] shows that the Bulletproofs design is simulation extractible, and we hypothesize that the vector commitment modifications do not affect this. However, it is not clear that the overall membership proving system instantiation must inherit this property. Further examination into this would be useful, but is out of scope here.

[3] `https://github.com/cypherstack/generalized-bulletproofs`

6 Spend authorization and linkability proving systems

The technical note specifies a requirement for a proving system for the following relation:

$$\Big\{\widetilde{O}, \widetilde{I}, R, L \in \mathbb{G}; x', y', r_i', r_j' : \widetilde{O} = x'G + y'T, R = r_i'V + r_j'T, L = x'\widetilde{I} - (x'r_i')U\Big\} \tag{4}$$

Note that we slightly modify the notation for witnesses; this is for consistency and ease of later analysis.

It is then stated that this relation may be instantiated by composing the Bulletproofs+ weighted inner-product argument from [5] and a Schnorr-based protocol from [3], both of which are then described. We make the nature of this composition more clear later.

As with relations 2 and 3, security requirements for such instantiations are unspecified, but are described in more formal detail in the cited references. For proper composition, we again require that CWEE and SHVZK be satisfied.

While these properties are satisfied for relations specific to the weighted inner-product and Schnorr protocols that are not given, CWEE does not hold for relation 4. This is the case since the witness value r_j is not extractible from the combined proving systems. However, this is not problematic in practice if relation 4 is replaced by relations for the two protocols and composed properly with relation 3 to then satisfy relation 2. We show later how to do this.

6.1 Weighted inner-product proving system

One proving system used to instantiate relation 4 is a weighted inner-product argument from [5]. It is an instantiation of the following relation:

$$\Big\{P \in \mathbb{G}; x', r_i', r_p' \in \mathbb{F}; P = x'G + r_i'V + (x'r_i')U + r_p'T\Big\} \tag{5}$$

The notation used for witness values is slightly modified here for ease of analysis. As shown in [5], the proving system is CWEE and SHVZK.

The technical note reproduces the interactive protocol, but also uses different notation. While this is not a problem (the notation is arbitrary), it incorrectly uses y in the prover's definition of A. This should be replaced with r_i, which then matches (up to the modified notation) the protocol in [5].

6.2 Schnorr protocol

The other proving system used to instantiate relation 4 is a Schnorr-like protocol for a relation that is not given, but described as using techniques from [3]. We give the relation here:

$$\Big\{\widetilde{O}, P, R, L, \widetilde{I} \in \mathbb{G}; x'', y'', z'', r_p'' \in \mathbb{F} : \\ \widetilde{O} = x''G + y''T, P - \widetilde{O} - R = z''U + r_p''T, L = x''\widetilde{I} - z''U\Big\} \quad (6)$$

We again modify witness notation for clearer analysis.

While the technical note provides the interactive protocol, we do so here with our modified notation for clarity:

- The prover samples $r_x, r_y, r_z, r_{r_p} \in \mathbb{F}$ and computes the following:

$$\begin{aligned} R_O &= r_xG + r_yT \\ R_P &= r_zU + r_{r_p}T \\ R_L &= r_x\widetilde{I} - r_zU \end{aligned}$$

- The prover sends R_O, R_P, R_L to the verifier.
- The verifier samples a challenge $e \in \mathbb{F}$ and sends it to the prover.
- The prover computes

$$\begin{aligned} s_x &= r_x + ex'' \\ s_y &= r_y + ey'' \\ s_z &= r_z + ez'' \\ s_{r_p} &= r_{r_p} + er_p'' \end{aligned}$$

  and sends these values to the verifier.

- The verifier accepts the proof if and only if the following hold:

$$\begin{aligned} R_O + e\widetilde{O} &= s_xG + s_yT \\ R_P + e(P - \widetilde{O} - R) &= s_zU + s_{r_p}T \\ R_L + eL &= s_x\widetilde{I} - s_zU \end{aligned}$$

The protocol can be made non-interactive using the (strong) Fiat-Shamir technique.

We now show that the protocol has desired security properties. Specifically, we do so to aid later work, which will use the extractor algorithm we define here to show CWEE for another protocol.

Theorem 1. *The protocol presented for relation 6 is perfectly complete, and satisfies CWEE and SHVZK.*

Proof. Perfect completeness follows immediately by inspection.

To show CWEE, we construct an extractor that produces a valid witness given a tree of accepting transcripts on arbitrary rewinding. The extractor fixes a statement and partial accepting transcript (R_O, R_P, R_L). It rewinds a single time to obtain distinct challenges $e \neq e'$ and corresponding partial accepting transcripts (s_x, s_y, s_z, s_{r_p}) and $(s_x', s_y', s_z', s_{r_p}')$. Using both transcripts, the first verification equation yields

$$(e - e')\widetilde{O} = (s_x - s_x')G + (s_y - s_y')T$$

from which we obtain that

$$\widetilde{O} = \left(\frac{s_x - s_x'}{e - e'}\right) G + \left(\frac{s_y - s_y'}{e - e'}\right) T$$

that is well defined since $e - e' \neq 0$. Similarly, we obtain that

$$P - \widetilde{O} - R = \left(\frac{s_z - s_z'}{e - e'}\right) U + \left(\frac{s_{r_p} - s_{r_p}'}{e - e'}\right) T$$

and

$$L = \left(\frac{s_x - s_x'}{e - e'}\right) \widetilde{I} + \left(\frac{s_z - s_z'}{e - e'}\right) U$$

from the remaining verification equations. This gives the witness extractions

$$\begin{aligned} x'' &= \frac{s_x - s_x'}{e - e'} \\ y'' &= \frac{s_y - s_y'}{e - e'} \\ z'' &= \frac{s_z - s_z'}{e - e'} \\ r_p'' &= \frac{s_{r_p} - s_{r_p}'}{e - e'} \end{aligned}$$

and the protocol is CWEE.

To show SHVZK, we construct a simulator that produces transcripts distributed identically to those of real proofs. Fix a statement and sampled challenge e. The simulator first samples s_x, s_y, s_z, s_{r_p} uniformly at random. It then sets

$$\begin{aligned} R_O &= s_x G + s_y T - e\widetilde{O} \\ R_P &= s_z U + s_{r_p} T - e(P - \widetilde{O} - R) \\ R_L &= s_x \widetilde{I} - s_z U - eL \end{aligned}$$

to satisfy the verification equations. □

6.3 Conjunction

It is possible to combine the weighted inner-product and Schnorr proving systems. This has two benefits: it reduces the communication complexity, and also aids in analysis of a later composition with the proving system used for relation 3.

The combined proving system is an instantiation of the following relation, which combines relations 5 and 6 as a conjunction:

$$\Big\{P, \widetilde{O}, R, L, \widetilde{I} \in \mathbb{G}; x', r_i', r_p', x'', y'', z'', r_p'' \in \mathbb{F} : \\ P = x'G + r_i'V + (x'r_i')U + r_p'T, \\ \widetilde{O} = x''G + y''T, P - \widetilde{O} - R = z''U + r_p''T, L = x''\widetilde{I} - z''U\Big\} \tag{7}$$

While it is well known how to produce an interactive protocol to show such a conjunction, this would not provide any communication complexity benefit. This technique requires that each constituent interactive protocol be run in parallel, using a common challenge derived from the combined transcript.

However, observe that in the weighted inner-product protocol, the prover samples $\alpha \in \mathbb{F}$ and sends $s_\alpha = \alpha + ex'$ to the verifier. In the Schnorr protocol, the prover samples $r_x \in \mathbb{F}$ and sends $s_x = r_x + ex''$ to the verifier. This means that if $x' = x''$ in relation 7, we have $s_\alpha = r_x$ if $\alpha = r_x$.

Protocol 1. The following is an instantiation of relation 7:

- The prover samples $\alpha, \beta, \delta, \mu, r_y, r_z, r_{r_p} \in \mathbb{F}$ uniformly at random.
- The prover computes the following:

$$\begin{array}{rcl} A & = & \alpha G + \beta V + (\alpha r_i' + \beta x')U + \delta T \\ B & = & (\alpha\beta)U + \mu T \\ R_O & = & \alpha G + r_y T \\ R_P & = & r_z U + r_{r_p} T \\ R_L & = & \alpha\widetilde{I} - r_z U \end{array}$$

- The prover sends A, B, R_O, R_P, R_L to the verifier.
- The verifier samples a challenge $e \in \mathbb{F}$ and sends it to the prover.
- The prover computes

$$\begin{array}{rcl} s_\alpha & = & \alpha + ex' \\ s_\beta & = & \beta + er_i' \\ s_\delta & = & \mu + e\delta + e^2 r_p' \\ s_y & = & r_y + ey'' \\ s_z & = & r_z + ez'' \\ s_{r_p} & = & r_{r_p} + er_p'' \end{array}$$

and sends these values to the verifier.

- The verifier accepts the proof if and only if the following hold:

$$\begin{aligned}
e^2P + eA + B &= (s_\alpha e)G + (s_\beta e)V + (s_\alpha s_\beta)U + s_\delta T \\
R_O + e\widetilde{O} &= s_\alpha G + s_y T \\
R_P + e(P - \widetilde{O} - R) &= s_z U + s_{r_p} T \\
R_L + eL &= s_\alpha \widetilde{I} - s_z U
\end{aligned}$$

This construction does not follow the standard approach for sigma protocol conjunction since the prover uses common randomness to sample $\alpha = r_x$ in the two constituent protocols, so we cannot immediately claim that CWEE and SHVZK hold. Further, it is not even the case that the protocol is complete, since this only holds in the case where $x' = x''$. However, both CWEE and SHVZK hold even when $x' \neq x''$. We will show later that this restriction on completeness is sufficient for the intended use case.

Theorem 2. *The protocol presented for relation 7 satisfies CWEE, SHVZK, and simulation extractibility. It is perfectly complete if $x' = x''$.*

Proof. Perfect completeness immediately follows by inspection if $x' = x''$.

To show CWEE, we construct an extractor that produces a valid witness given a tree of accepting transcripts on arbitrary rewinding. The extractor first runs the steps of the weighted inner-product extractor in [5] to produce witness values x', r_i' satisfying the first condition in relation 7. It then runs the steps of the Schnorr extractor to produce witness values x'', y'', z'', r_p'' satisfying the remaining conditions.

To show SHVZK, we construct a simulator that produces transcripts distributed identically to those of real proofs, following the method used in [5]. Fix a statement $P, \widetilde{O}, R, L, \widetilde{I}$ and sampled challenge e. The simulator first samples

$$A, s_\alpha, s_\beta, s_\delta, s_y, s_z, s_{r_p}$$

uniformly at random. It then sets the following:

$$\begin{aligned}
B &= (s_\alpha e)G + (s_\beta e)V + (s_\alpha s_\beta)U + s_\delta T - e^2P - eA \\
R_O &= s_\alpha G + s_y T - e\widetilde{O} \\
R_P &= s_z U + s_{r_p} T - e(P - \widetilde{O} - R) \\
R_L &= s_\alpha \widetilde{I} - s_z U - eL
\end{aligned}$$

The resulting transcript is accepting by definition.

It remains to show that the distributions of simulated and real transcripts are identical; that is, that the transcript values sampled uniformly at random by the simulator are similarly distributed in real proofs. We first note that in real proofs, the values sampled uniformly at random by the prover are $\alpha, \beta, \delta, \mu, r_y, r_z, r_{r_p}$. It therefore suffices to show that the mapping

$$(\alpha, \beta, \delta, \mu, r_y, r_z, r_{r_p}) \mapsto (\log_T A, s_\alpha, s_\beta, s_\delta, s_y, s_z, s_{r_p})$$

defined by the prover is a bijection.

To show that it is surjective, consider an arbitrary codomain tuple

$$(\log_T A, s_\alpha, s_\beta, s_\delta, s_y, s_z, s_{r_p}) \in \mathbb{F}^7$$

and observe that the following values constitute a preimage:

$$\begin{array}{rcl}
\alpha & = & s_\alpha - ex' \\
\beta & = & s_\beta - er'_i \\
\delta & = & \log_T A - \alpha G - \beta V - (\alpha r'_i + \beta x')U \\
\mu & = & s_\delta - e\delta - e^2 r'_p \\
r_y & = & s_y - ey'' \\
r_z & = & s_z - ez'' \\
r_{r_p} & = & s_{r_p} - er''_p
\end{array}$$

To show that it is injective, let $\zeta_G = \log_T G, \zeta_V = \log_T V, \zeta_U = \log_T U$. Then the mapping can be described by the following matrix equation:

$$\begin{pmatrix} \log_T A \\ s_\alpha \\ s_\beta \\ s_\delta \\ s_y \\ s_z \\ s_{r_p} \end{pmatrix} = \begin{pmatrix} \zeta_G & \zeta_V & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \delta \\ \mu \\ r_y \\ r_z \\ r_{r_p} \end{pmatrix} + \begin{pmatrix} (\alpha r'_i + \beta x')\zeta_U \\ ex \\ er'_i \\ e^2 r'_p \\ ey'' \\ ez'' \\ er''_p \end{pmatrix}$$

Suppose that $(\widehat{\alpha}, \widehat{\beta}, \widehat{\delta}, \widehat{\mu}, \widehat{r}_y, \widehat{r}_z, \widehat{r}_{r_p})$ is another preimage to the same image. Then

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \zeta_G & \zeta_V & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha - \widehat{\alpha} \\ \beta - \widehat{\beta} \\ \delta - \widehat{\delta} \\ \mu - \widehat{\mu} \\ r_y - \widehat{r}_y \\ r_z - \widehat{r}_z \\ r_{r_p} - \widehat{r}_{r_p} \end{pmatrix} + \begin{pmatrix} (\alpha r'_i + \beta x' - \widehat{\alpha} r'_i - \widehat{\beta} x')\zeta_U \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

holds. This immediately gives the following equalities:

$$\begin{aligned}\alpha &= \widehat{\alpha} \\ \beta &= \widehat{\beta} \\ r_y &= \widehat{r}_y \\ r_z &= \widehat{r}_z \\ r_{r_p} &= \widehat{r}_{r_p}\end{aligned}$$

Substituting these values, we find $\delta = \widehat{\delta}$ and $\mu = \widehat{\mu}$ as well.

To show simulation extractibility, we use the result of [9]. This gives the property if the protocol has quasi-unique responses, which means that a bounded adversary cannot produce accepting transcripts on the same statement, initial transcript, and challenge that differ in the complete transcript. In order to show quasi-unique responses, we examine the verification equations for the protocol, fixing a statement $(P, \widetilde{O}, R, L, \widetilde{I})$, initial transcript (A, B, R_O, R_P, R_L), and challenge e.

In the first verification equation

$$e^2P + eA + B = (s_\alpha e)G + (s_\beta e)V + (s_\alpha s_\beta)U + s_\delta T$$

observe that if two accepting transcripts differ in the tuple $(s_\alpha, s_\beta, s_\delta)$, then the adversary has produced distinct openings for $e^2P + eA + B$ with respect to generators (G, V, U, T); this is a contradiction since these generators are independent and such an opening breaks computational binding. Similarly, from the second verification equation

$$R_O + e\widetilde{O} = s_\alpha G + s_y T$$

observe that transcripts differing in (s_α, s_y) result in distinct openings for $R_O + e\widetilde{O}$ with respect to (G, T), again a contradiction. Finally, from the third verification equation

$$R_P + e(P - \widetilde{O} - R) = s_z U + s_{r_p} T$$

observe that transcripts differing in (s_z, s_{r_p}) result in distinct openings for $R_P + e(P - \widetilde{O} - R)$ with respect to (U, T), also a contradiction. This means the protocol has quasi-unique responses, and is therefore simulation extractible. □

7 Composed proving system

We now describe how to compose protocols that can be described by relations 3 (for membership) and 7 (for spend authority and linkability) to obtain a proving system meeting the conditions of the main relation 2. Then, we will analyze its security using the analyses of the constituent protocols.

Suppose the prover wishes to generate a proof to be used in a transaction that consumes an output with the following components:

$$\begin{aligned}O &= xG + yT \\ I &= \mathsf{H}(O)\end{aligned}$$

The output also comes equipped with a value commitment C, but its opening is not needed here. Suppose that the prover has also chosen $r_c \in \mathbb{F}$ such that $\widetilde{C} = C + r_c G$. Let $S \subset \mathbb{G}^3$ be the set of valid outputs, such that $(O, I, C) \in S$ and $L = xI$ is the linking tag for the output.

Protocol 2. The interactive protocol is as follows:

- The prover samples $r_o, r_i, r_j \in \mathbb{F}$ uniformly at random, and computes the following:

$$\begin{aligned} \widetilde{O} &= O + r_o T \\ \widetilde{I} &= I + r_i U \\ R &= r_i V + r_j T \end{aligned}$$

- The prover runs the prover for the membership relation 3 on the following input:

$$\left\{ \widetilde{O}, \widetilde{I}, \widetilde{C}, R; r_o, r_i, r_j, r_c, (O, I, C) \right\}$$

- The prover samples $r_p' \in \mathbb{F}$ uniformly at random, and computes the following:

$$P = xG + r_i V + (xr_i)U + r_p' T$$

- The prover runs the prover for the conjunction relation 7 on the following input:

$$\left\{ P, \widetilde{O}, R, L, \widetilde{I}; x, r_i, r_p', x, y + r_o, xr_i, r_p' - y - r_o - r_j \right\}$$

- The verifier runs the verifiers for each of the relations, and accepts the proof if and only if each succeeds.

We claim that this protocol meets the requirements of the main relation 2.

Theorem 3. *The protocol presented for relation 2 is perfectly complete, and satisfies CWEE and SHVZK.*

Proof. Perfect completeness immediately follows by inspection due to the common witness x used for relation 7.

To show SHVZK, we construct a simulator that produces transcripts distributed identically to those of real proofs. Given a statement and fixed challenges, the simulator first samples $\widetilde{O}, \widetilde{I}, \widetilde{C}, R, P \in \mathbb{G}$ uniformly at random. It then runs the simulators for the proving systems instantiating relations 3 and 7 using the corresponding statements given in the protocol.

The resulting transcript is accepting by definition. To show such transcripts are distributed identically to those of real proofs, it suffices to observe that because the values r_o, r_i, r_c, r_j, r_p' are sampled uniformly at random, the transcript elements $\widetilde{O}, \widetilde{I}, \widetilde{C}, R, P$ are independently distributed uniformly at random as well.

To show CWEE, we construct an extractor that produces a valid witness given a tree of accepting transcripts on arbitrary rewinding. Given a statement, the extractor first runs the extractor for the proving system instantiating relation 3 to obtain witness values $r_o, r_i, r_j, r_c, (O, I, C)$ satisfying the relation. It then runs the extractor for the proving system instantiating relation 7 to obtain witness values $x', r'_i, r'_p, x'', y'', z'', r''_p$ satisfying the relation.

In particular, we have

$$\begin{aligned} P - \widetilde{O} - R &= z''U + r''_pT \\ &= (x'G + r'_iV + (x'r'_i)U + r'_pT) - (x''G + y''T) - (r_iV + r_jT) \end{aligned}$$

If the discrete logarithm problem is hard in $\mathbb{G}$ and all group generators are independently sampled uniformly at random, we can compare coefficients of each generator separately (or else the extractor has produced a nontrivial discrete logarithm relation). In particular, we obtain the following:

$$\begin{aligned} z'' &= x'r'_i \\ x' &= x'' \\ r'_i &= r_i \end{aligned}$$

Then

$$\begin{aligned} L &= x''\widetilde{I} - z''U \\ &= x'(I + r_iU) - (x'r_i)U \\ &= x'I \end{aligned}$$

and since

$$\begin{aligned} O + r_oT &= x''G + y''T \\ &= x'G + y''T \end{aligned}$$

it follows that $O = x'G + (y'' - r_o)T$.

This provides the required witness $x = x', y = y'' - r_o, (O, I, C)$ corresponding to the statement and completes the proof. □

Note that we do not claim that the protocol is simulation extractible. As discussed earlier, it may not be the case that the protocol used to instantiate the membership relation is simulation extractible. Even if so, it may not be the case that the protocol's use of multiple rounds satisfies the result of [11], which places additional requirements on quasi-unique responses; this is out of scope.

8 Analysis

8.1 Security properties

Previously, we introduced informal security properties that a transaction protocol built from this construction must possess. Using this analysis, we can show

how these properties might hold if the construction is suitably integrated into such a protocol.

First, it must be the case that all outputs consumed in a transaction be generated in previous transactions. This follows directly from the extraction of a witness tuple (O, I, C) that exists in the set S of valid transaction outputs.

Next, no output consumed in a valid transaction may have been consumed in any previous transaction. To show this, we observe that extraction provides x and y such that $O = xG + yT$ and $L = xI$. Suppose another transaction consumes O and presents the linking tag L' as part of its statement. Since the witness values x and y bind computationally to O, this second transaction must contain a proof extracting these values. This means $L' = xI = L$; if the verifier rejects transactions providing non-unique linking tags, the second transaction will be rejected and the desired property holds.

It must also hold that the transaction was authorized by an entity possessing all signing keys for consumed outputs. This follows from correct use of the strong Fiat-Shamir technique properly applied to the composed proving system since valid proofs extract the signing keys x and y associated to each consumed output $O = xG + yT$ extracted in the proof. We stress that, as noted in [1], proper binding of statement data into challenge hashes using the strong technique is crucial to avoid generic attacks.

However, we note that this non-interactive transformation must bind transaction data in a non-malleable way, which requires in part that the prover and verifier actually query the cryptographic hash function used for challenge computation to include such data as the so-called "message" component of the hash input. This would not occur, for example, in the (malicious) edge case where all statement and transcript elements multiplicatively applied to the challenge e are zero; in this case, the adversary could trivially bind any message of its choice since neither it nor the verifier need query the hash function on the message. Such a case may be easily avoided by requiring that at least one such value be nonzero. Fortunately, the Monero protocol already requires that $L \neq 0$ as an added verifier check, which is sufficient. This assertion should be retained.

8.2 Denial of service via slanderability

We note a property not directly referenced and therefore out of scope: that an entity possessing the signing key associated to an output be able to produce a proof consuming it that a verifier will accept. This is not trivial, in part because it relies on the structure of linking tags. Such a property is not captured by the informal security requirements of the technical note; however, we mention it here since the relationship between linking tags and outputs defined here is different from the existing Monero protocol.

Specifically, in the existing Monero protocol, output keys are of the form $O = xG$ and linking tags are $L = xI = x\mathsf{H}(O)$. In the updated protocol, output keys are of the form $O = xG + yT$, with the same linking tag definition. This means that existing analysis of Monero protocol security does not necessarily translate to the updated design.

To demonstrate the idea, suppose an honest transaction extracts an opening $O = xG + yT$ to a consumed output and reveals the linking tag $L = xI = x\mathsf{H}(O)$. An adversary may attempt to generate a valid transaction extracting an opening $O' = x'G + y'T$ to a different consumed output that reveals the linking tag $L' = xI' = x'\mathsf{H}(O')$. The adversary succeeds if $L = L'$; that is, if $x\mathsf{H}(O) = x'\mathsf{H}(O')$. In this case, if the adversary's transaction is accepted by the network before the honest transaction, the honest transaction will be rejected, and the adversary has executed a denial of service.

One approach to analyzing this attack is the concept of non-slanderability, which is addressed for the Monero protocol in [6] and for the related Omniring transaction protocol in [13]. Intuitively, it should be infeasible for the adversary to produce targeted collisions against honest linking tags.

In [13], an instantiation of a specific transaction protocol design has the non-slanderability property if an adversary can produce a valid transaction whose linking tag matches that of an honest transaction using a particular security game. This relates to unforgeability through a property showing that linking tags bind computationally to outputs. Non-slanderability is then proven for the Omniring transaction protocol by reducing the success of such an adversary to a break of a one-way property of the linking tag construction; this property is shown to hold for the linking tag design currently used in the Monero protocol as well. We note that the one-way linking tag property is proven in a manner intended to capture Monero-style output construction using derived one-time addresses, whereas the technical note defines output signing keys more generally.

In [6], a comprehensive security model for the Monero transaction protocol is introduced under the algebraic group model. It defines non-slanderability in a similar manner, but using different definitions and security games. The property is proven for the Monero protocol and its linking tag structure using a general lemma applied to specific kinds of oracle queries. While this lemma does not directly rely on the algebraic group model, its application to a non-slanderability game does.

It is difficult to definitely claim that the methods and conclusions in [13] and [6] apply here. As noted, both methods of proof assume different security games as part of a specific transaction security model; the technical note does not construct its proving relations as part of either security model. Additionally, the individual results about the Monero linking tag structure (where $y = 0$) do not directly apply to the modified linking tag construction. We expect, however, that the well-structured nature of the constituent proving systems would be amenable to the design of [6], though reliance on the algebraic group model may be undesirable.

8.3 Post-quantum forward secrecy

The technical note defines forward secrecy in a post-quantum context, assuming efficient computation of arbitrary discrete logarithms in $\mathbb{G}$. Specifically, it describes an algorithm (via a separately-provided script) that, given a statement for relation 2 and corresponding valid composed spend authorization and linkability proof, can reconstruct the prover's view using witness data consistent with any given output. Using this logic, a post-quantum adversary cannot definitively identify which outputs were consumed in a valid transaction.

The construction of this algorithm is correct and meets its goal. However, it does not address the presence of other transaction data that must be checked by the network in order to conclude that a transaction is valid, aside from noting that verification properties involving the value commitment C are likely to follow trivially from its construction as a perfectly-hiding Pedersen commitment and the "sum-to-zero" method used in the Monero protocol to assert balance in transactions via commitment rerandomization. In particular, each transaction must come equipped with a valid membership proof instantiating relation 3. It may be the case that the proof does not admit prover reconstruction using an arbitrary specified output as part of its witness. The forward secrecy property therefore depends in part on the choice of instantiation of the relation for this proof.

References

[1] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 626–643, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[2] 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, 2018.

[3] Jan Camenisch, Aggelos Kiayias, and Moti Yung. On the portability of generalized Schnorr proofs. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 425–442, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[4] Matteo Campanelli, Mathias Hall-Andersen, and Simon Holmgaard Kamp. Curve trees: Practical and transparent zero-knowledge accumulators. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4391–4408, Anaheim, CA, August 2023. USENIX Association.

[5] Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. *IEEE Access*, 10:42081–42096, 2022.

[6] Cas Cremers, Julian Loss, and Benedikt Wagner. A holistic security analysis of Monero transactions. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 129–159, Cham, 2024. Springer Nature Switzerland.

[7] Quang Dao and Paul Grubbs. Spartan and Bulletproofs are simulation-extractable (for free!). In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 531–562, Cham, 2023. Springer Nature Switzerland.

[8] Quang Dao, Jim Miller, Opal Wright, and Paul Grubbs. Weak Fiat-Shamir attacks on modern proof systems. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 199–216, 2023.

[9] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012*, pages 60–79, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] Chaya Ganesh, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Fiat–Shamir Bulletproofs are non-malleable (in the algebraic group model). In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 397–426, Cham, 2022. Springer International Publishing.

[11] Chaya Ganesh, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Fiat-Shamir Bulletproofs are non-malleable (in the random oracle model). Cryptology ePrint Archive, Paper 2023/147, 2023. `https://eprint.iacr.org/2023/147`.

[12] Brandon Goodell, Sarang Noether, and Arthur Blue. Concise linkable ring signatures and forgery against adversarial keys. Cryptology ePrint Archive, Paper 2019/654, 2019. `https://eprint.iacr.org/2019/654`.

[13] Russell W. F. Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling private payments without trusted setup. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 31–48, New York, NY, USA, 2019. Association for Computing Machinery.

[14] Shen Noether, Adam Mackenzie, et al. Ring confidential transactions. *Ledger*, 1:1–18, 2016.

[15] Andrew Poelstra, Adam Back, Mark Friedenbach, Gregory Maxwell, and Pieter Wuille. Confidential assets. In Aviv Zohar, Ittay Eyal, Vanessa Teague, Jeremy Clark, Andrea Bracciali, Federico Pintore, and Massimiliano Sala, editors, *Financial Cryptography and Data Security*, pages 43–63, Berlin, Heidelberg, 2019. Springer Berlin Heidelberg.