Anonymous Zether: Technical Report

Benjamin E. Diamond
J.P. Morgan

Abstract

We describe a protocol for the *anonymous Zether* payment system of Bünz, Agrawal, Zamani, and Boneh [BAZB]. We first discuss "Σ-Bullets"—which adapt *Bulletproofs* to the setting of ElGamal ciphertexts—and address shortcomings in the approach of [BAZB].

We further study the *anonymous* extension proposed in [BAZB, §D]. Extending Groth and Kohlweiss's *one out of many* proofs [GK15], we prove knowledge of a secret permutation (of a certain form) of ElGamal ciphertexts with heterogeneous keys. We finally give an efficient implementation based on the number-theoretic transform.

Introduction

Zether is a cryptographic protocol for confidential payment, described in a manuscript of Bünz, Agrawal, Zamani and Boneh [BAZB]. In contrast to well-known protocols like Monero and Zcash, Zether is "account-based", and features constant long-term space overhead per participant (though see also Quisquis [FMMO]); Zether also lacks a trusted setup.

The manuscript [BAZB] focuses on *basic Zether*, in which account balances and transfer amounts are concealed, but participants' identities are not. Its central cryptographic technique, called "Σ-Bullets", seeks to adapt Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell's *Bulletproofs* [BBB$^+$] (originally designed for Pedersen commitments) to the setting of ElGamal ciphertexts. We argue in Section 2 below that the Σ-Bullets protocol of [BAZB] has security flaws, and describe concrete attacks on it (targeting both soundness and zero-knowledge). We also propose a secure amendment.

In Section 3, we turn to *anonymous* payment, as proposed in [BAZB, §D]. The appendix [BAZB, §D] suggests a relation, but no proof protocol; our work proposes one. We extend *one out of many proofs*—introduced in Groth and Kohlweiss [GK15] and extended by Bootle, Cerulli, Chaidos, Ghadafi, Groth, and Petit [BCC$^+$15]—so as to prove knowledge not just of a secret index but of a secret *permutation* (of a certain form) of ElGamal ciphertexts; we further allow ciphertexts encrypted under heterogeneous keys. Our protocol conducts a variant of basic Zether on these *permuted* (re-encrypted) ciphertexts. Our security proofs show, among other things, that knowledge of discrete logarithm relations on these permuted ciphertexts carries over to the originals (a technique which may be of independent interest).

We observe further that our permutation protocol entails, computationally, a handful of discrete circular convolutions; we make our protocol efficient using ideas related to the number-theoretic transform. We finally describe an implementation of our system, and report its performance characteristics.

1 Review of Basic and Anonymous Zether

We briefly summarize both basic and anonymous Zether; for further details we refer to [BAZB].

Zether's global state consists of a mapping acc from ElGamal public keys to ElGamal ciphertexts; y's table entry contains an encryption of y's balance b (in the exponent). Schematically, we can write:

$$\begin{aligned} \mathsf{acc} &\colon \mathbb{G} \to \mathbb{G}^2, \\ y &\mapsto \mathsf{acc}[y] = \mathsf{Enc}_y(b, r) = \left(g^b y^r, g^r\right), \end{aligned}$$

for some randomness r which y in general does not know. (For details on the synchronization issues surrounding "epochs", we refer to [BAZB].)

1.1 Basic Zether

In "basic" (non-anonymous) Zether, a non-anonymous sender y may transfer funds to a non-anonymous recipient $\overline{y}$. To do this, y should publish the public keys y and $\overline{y}$, as well as a pair of ciphertexts (C, D) and $(\overline{C}, D)$ (the randomness may be shared). These should encrypt, under y and $\overline{y}$'s keys, the quantities g^{b^*} and g^{-b^*}, respectively, for some integer $b^* \in \{0, \dots \mathsf{MAX}\}$ (MAX is a fixed constant of the form $2^n - 1$). To apply the transfer, the administering system (e.g., smart contract) should group-subtract (C, D) and $(\overline{C}, D)$ from y and $\overline{y}$'s account balances (respectively). We denote by (C_{Ln}, C_{Rn}) y's balance *after* the homomorphic deduction is performed.

Finally, the prover should prove knowledge of:

- sk for which $g^{\mathsf{sk}} = y$ (knowledge of secret key),
- r for which:
  - $g^r = D$ (knowledge of randomness),
  - $(y \cdot \overline{y})^r = (C \cdot \overline{C})$ (ciphertexts encrypt opposite balances),
- b^* and b^* in $\{0, \dots, \mathsf{MAX}\}$ for which $C = g^{b^*} \cdot D$ and $C_{Ln} = g^{b'} \cdot C_{Rn}$ (overflow and overdraft protection).

Formally, we have the relation below, which essentially reproduces [BAZB, (2)]:

$$\mathsf{st}_{\mathsf{ConfTransfer}} : \left\{ y, \overline{y}, C_{Ln}, C_{Rn}, C, \overline{C}, D, g; \mathsf{sk}, b^*, b', r : \begin{array}{c} C = g^{b^*} \cdot D^{\mathsf{sk}} \wedge C_{Ln} = g^{b'} \cdot C_{Rn}^{\mathsf{sk}} \wedge (y \cdot \overline{y})^r = C \cdot \overline{C} \wedge \\ g^{\mathsf{sk}} = y \wedge D = g^r \wedge b^* \in \{0, \dots, \mathsf{MAX}\} \wedge b' \in \{0, \dots, \mathsf{MAX}\} \end{array} \right\}.$$

1.2 Anonymous Zether

In anonymous Zether [BAZB, §D], a sender may hide herself and the recipient in a larger "ring" $(y_i)_{i=0}^{N-1}$. To an observer, it should be impossible to discern which among a ring's members sent or received funds. Specifically, a sender should choose a list $(y_i)_{i=0}^{N-1}$, as well as indices idx_0 and idx_1 for which y_{idx_0} and y_{idx_1} belong to the sender and recipient, respectively. The sender should then publish this list, as well as a list of ciphertexts $(C_i, D)_{i=0}^{N-1}$ for which (C_{idx_0}, D) encrypts g^{b^*} under y_{idx_0}, (C_{idx_1}, D) encrypts g^{-b^*} under y_{idx_1}, and (C_i, D) for each $i \notin \{\mathsf{idx}_0, \mathsf{idx}_1\}$ encrypts g^0 under y_i. To apply the transfer, the contract should deduct each (C_i, D) from y_i's balance; we denote the list of *new* balances by $(C_{Ln,i}, C_{Rn,i})_{i=0}^{N-1}$.

Finally, the prover should prove knowledge of:

- $\mathsf{idx}_0, \mathsf{idx}_1 \in \{0, \dots, N-1\}$ (sender's and recipient's secret indices),
- sk for which $g^{\mathsf{sk}} = y_{\mathsf{idx}_0}$ (knowledge of secret key),
- r for which:
  - $g^r = D$ (knowledge of randomness),
  - $(y_{\mathsf{idx}_0} \cdot y_{\mathsf{idx}_1})^r = C_{\mathsf{idx}_0} \cdot C_{\mathsf{idx}_1}$ (sender's and receiver's ciphertexts encrypt opposite balances),
  - for each $i \notin \{\mathsf{idx}_0, \mathsf{idx}_1\}$, $y_i^r = C_i$ (all ciphertexts other than the sender's and recipient's encrypt 0),
- b^* and b^* in $\{0, \dots, \mathsf{MAX}\}$ for which $C_{\mathsf{idx}_0} = g^{b^*} \cdot D$ and $C_{Ln,\mathsf{idx}_0} = g^{b'} \cdot C_{Rn,\mathsf{idx}_0}$ (overflow and overdraft protection).

We group these facts into a formal relation, closely following [BAZB, (8)]. For technical reasons (described below), we actually prove a slight variant of this relation, in which N is required to be *even* and idx_0 and idx_1 are required to have opposite parity. Formally:

$$\mathsf{st}_{\mathsf{AnonTransfer}} : \left\{ \begin{array}{l} \left((y_i, C_i, C_{Ln,i}, C_{Rn,i})_{i=0}^{N-1}, D, u, g, g_{\mathsf{epoch}}; \mathsf{sk}, b^*, b', r, (s_i, t_i)_{i=0}^{N-1} \right) : \\ \prod_{i=0}^{N-1} C_i^{s_i} = g^{b^*} \prod_{i=0}^{N-1} y_i^{r \cdot s_i} \wedge \prod_{i=0}^{N-1} C_i^{s_i + t_i} = \prod_{i=0}^{N-1} y_i^{r \cdot (s_i + t_i)} \wedge D = g^r \wedge \\ \left(C_i^{(1-s_i)(1-t_i)} = y_i^{r \cdot (1-s_i)(1-t_i)} \right)_{i=0}^{N-1} \wedge \prod_{i=0}^{N-1} C_{Ln,i}^{s_i} = g^{b'} \left(\prod_{i=0}^{N-1} C_{Rn,i}^{s_i} \right)^{\mathsf{sk}} \wedge \\ g^{\mathsf{sk}} = \prod_{i=0}^{N-1} y_i^{s_i} \wedge g_{\mathsf{epoch}}^{\mathsf{sk}} = u \wedge b^* \in \{0, \ldots, \mathsf{MAX}\} \wedge b' \in \{0, \ldots, \mathsf{MAX}\} \\ (s_i \in \{0,1\} \wedge t_i \in \{0,1\})_{i=0}^{N-1} \wedge \sum_{i=0}^{N-1} s_i = \sum_{i=0}^{N-1} t_i = 1 \wedge \\ N \equiv 0 \mod 2 \wedge s_{i_0} = t_{i_1} = 1 \implies i_0 \not\equiv i_1 \mod 2 \end{array} \right\}. \tag{1}$$

The requirement that $\mathsf{idx}_0 \not\equiv \mathsf{idx}_1 \mod 2$ decreases privacy, but not by much. Indeed, this requirement decreases the cardinality of the set of possible pairs $(\mathsf{idx}_0, \mathsf{idx}_1) \in \{0, \ldots, N-1\}^2$ from N^2 to $\frac{N^2}{2}$; this latter cardinality of course still grows quadratically in N.

2 Bulletproofs and ElGamal Ciphertexts

We now describe the "Σ-Bullets" protocol of [BAZB], and our improvements. In fact, the protocol as described on page 48 of [BAZB, §G] is not complete as written; to see this, note that $\left(\frac{C^c}{D^{s_{\mathsf{sk}}}}\right)^{z^2} \cdot \left(\frac{C_{Ln}^c}{C_{Rn}^{s_{\mathsf{sk}}}}\right)^{z^3} = \left(D^{z^2} \cdot C_{Rn}^{z^3}\right)^{-k_{\mathsf{sk}}} \cdot g^{c \cdot (b^* \cdot z^2 + b' \cdot z^3)}$, whereas $g^{s_b} = g^{k_b} \cdot g^{c \cdot (b^* \cdot z^2 + b' \cdot z^3)}$. We thus consider the version implemented in the repository `bbuenz / BulletProofLib`. In this version, the the prover sends $A_t = \left(D^{z^2} \cdot C_{Rn}^{z^3}\right)^{k_{\mathsf{sk}}}$; the verifier then checks

$$g^{c \cdot \hat{t}} \cdot h^{c \cdot \tau_x} \stackrel{?}{=} g^{c \cdot \delta(y,z)} \cdot A_t \cdot {c_{\mathsf{commit}}}^{-1} \cdot \left(T_1^x \cdot T_2^{x^2}\right)^c, \tag{2}$$

where $c_{\mathsf{commit}} = \left(D^{z^2} \cdot C_{Rn}^{z^3}\right)^{s_{\mathsf{sk}}} \cdot \left(C^{z^2} \cdot C_{Ln}^{z^3}\right)^{-c}$.

2.1 Attacks on [BAZB]

Attack (Soundness)**.** Informally, nothing prevents the message of (C, D) from having an "h part". Suppose that the prover were to construct (C, D) so as to encrypt the message $g^{b^*} h^\gamma$, for some nonzero γ (and b^* as usual). (The recipient's ciphertext, $(\overline{C}, D)$, would encrypt $g^{-b^*} h^{-\gamma}$.) By adding the constant term $z^2 \cdot \gamma$ to τ_x (as implemented in `BulletProofLib`, τ_x has no constant term) the prover can preserve the validity of equation 2; on the other hand, the message $g^{b^*} h^\gamma$ would completely invalidate the recipient's ciphertext (and the sender's).

Attack (Zero knowledge)**.** Informally, the term $A_t = \left(D^{z^2} \cdot C_{Rn}^{z^3}\right)^{k_{\mathsf{sk}}}$ allows the verifier to decrypt the combined ciphertext $(C, D)^{z^2} \cdot (C_{Ln}, C_{Rn})^{z^3}$. Indeed, after honest behavior by the prover, the term $A_t \cdot {c_{\mathsf{commit}}}^{-1}$ equals $g^{c \cdot (b^* \cdot z^2 + b' \cdot z^3)}$; from this quantity, the verifier may brute-force the values b^* and b' using only 2^{64} work (and much less, if the verifier can "guess" these values sooner).

2.2 A refined "Σ-Bullets"

We propose a modified version of "Σ-Bullets" which addresses both of these issues. In fact, this is a *generic* approach for applying Bulletproofs to ElGamal-encrypted integers (in the exponent). For notational ease, we specialize to the case of basic Zether.

We informally describe our amended version; a detailed protocol (which also includes anonymity) can be found in Section 5 below. Alongside the initial commitments A and S, the prover should publish additional ciphertexts (C', D') and (C'_{Ln}, C'_{Rn}) which encrypt h^{γ^*} and $h^{\gamma'}$ respectively (for randomly chosen scalars γ^* and γ'). Upon receiving x, the prover should add to τ_x the constant term $z^2 \cdot \gamma^* + z^3 \cdot \gamma'$. During the sigma protocols, in addition to proving knowledge of sk for which $g^{\mathsf{sk}} = y$, the prover should *also* prove knowledge of b^*, b', γ^*, and γ' for which:

$$g^{b^*} = D^{-\mathsf{sk}} \cdot C, \quad g^{b'} = C_{Rn}^{-\mathsf{sk}} \cdot C_{Ln}, \quad h^{\gamma^*} = D'^{-\mathsf{sk}} \cdot C', \quad h^{\gamma'} = C'^{-\mathsf{sk}}_{Rn} \cdot C'_{Ln}.$$

Finally, the prover instead defines $A_t = \left((D \cdot D')^{z^2} \cdot (C_{Rn} \cdot C'_{Rn})^{z^3}\right)^{k_{\mathsf{sk}}}$, while in the check 2 the verifier instead uses $c_{\mathsf{commit}} = \left((D \cdot D')^{z^2} \cdot (C_{Rn} \cdot C'_{Rn})^{z^3}\right)^{s_{\mathsf{sk}}} \cdot \left((C \cdot C')^{z^2} \cdot (C_{Ln} \cdot C'_{Ln})^{z^3}\right)^{-c}$.

The additional sigma-proofs mitigate the soundness attack: they ensure that (C, D) and (C_{Ln}, C_{Rn}) only have "g parts" (and that (C', D') and (C'_{Ln}, C'_{Rn}) only have "h parts"). The ciphertexts (C', D') and (C'_{Ln}, C'_{Rn}) themselves serve to blind the messages of (C, D) and (C_{Ln}, C_{Rn}). Indeed, the verifier may only decrypt a linear combination of all four ciphertexts; we note that $A_t \cdot {c_{\mathsf{commit}}}^{-1} = \left(g^{b^*} h^{\gamma^*}\right)^{c \cdot z^2} \cdot \left(g^{b'} h^{\gamma'}\right)^{c \cdot z^3}$. This decryption reveals nothing about b^* and b', and can be "fed into" the standard Bulletproofs protocol.

3 Anonymous Payments

3.1 One-out-of-many proofs and vector rotations

We now discuss our approach to anonymity. We begin with *one out of many proofs*, introduced by Groth and Kohlweiss [GK15] and extended by Bootle, Cerulli, Chaidos, Ghadafi, Groth, and Petit [BCC+15]. These techniques serve to prove knowledge—given a fixed list of commitments—of a secret element among this list and an opening for this element to the message 0.

In fact, these techniques admit more flexibility than is explicitly described in [GK15] and [BCC+15]. An *intermediate* step of these protocols entails the construction of a "re-encryption" of the secret list element (same message but new randomness); in the final step, the prover simply reveals this re-encryption's randomness (the verifier may check explicitly whether it opens to 0). Instead of revealing the re-encryption's randomness, however, the prover may use it in further protocols. This idea will be key in our work.

The first challenge is that [GK15] and [BCC+15] require homomorphic commitment schemes, like Pedersen commitments or ElGamal encryptions under *the same* key. Using minor adjustments, we extend these protocols so as to support lists of ElGamal encryptions belonging to *arbitrary* public keys.

The most significant challenge of [GK15] and [BCC+15] is that only *one* re-encryption is delivered, whereas we want something more like a permutation. Informally, we want to first run [BCC+15] twice, so as to deliver to the verifier re-encryptions of (C_{idx_0}, D) and (C_{idx_1}, D) (while concealing these elements' indices in the original list); using fairly simple extensions, we may also deliver a re-encryption of $(C_{Ln,\mathsf{idx}_0}, C_{Rn,\mathsf{idx}_0})$ and "re-encryptions" (i.e., exponentiations) of y_{idx_0} and y_{idx_1}, where the secrets idx_0 and idx_1 are provably chosen consistently throughout. Having in hand these re-encrypted values, the prover and verifier may conduct basic Zether on them. (That the protocol remains sound even when conducted on *re-encryptions* is non-trivial, but true, as we show below.)

All this says nothing, however, about the remaining equalities $y_i^r = C_i$ for $i \notin \{\mathsf{idx}_0, \mathsf{idx}_1\}$, which present a serious difficulty. A naïve approach would essentially conduct [BCC+15] N times, adopting certain additional modifications to ensure that the secret index is chosen distinctly each time. These modifications—at least as we are able to see—are incompatible with the logarithmic optimizations of

[GK15], however, so that this approach would demand $\mathcal{O}(N^2)$ communication ($\mathcal{O}(N \log N)$ communication would *perhaps* be possible, though non-trivial).

We reduce the communication to $\mathcal{O}(N)$ using the following trick, which necessitates that we further deconstruct the protocols [GK15] and [BCC+15]. A key step in these protocols is the definition of a vector $(p_i(X))_{i=0}^{N-1}$ of polynomials, and the delivery to the verifier of their *evaluations* $(p_i(w))_{i=0}^{N-1}$ at a challenge w. Because $p_i(X)$ is of "high degree" just when i equals the secret index idx_0, the verifier can use $(p_i(w))_{i=0}^{N-1}$ (plus some pre-challenge "correction terms") to pick out a re-encryption of the desired list element. Our observation is essentially that this vector may be reused and "rotated" in order to systematically pick out the *other* terms. Thus, the protocol need only be conducted once—or twice, rather—to handle (C_{idx_0}, D) and (C_{idx_1}, D); once this is done, the *same* vectors $(p_i(w))_{i=0}^{N-1}$ and $(q_i(w))_{i=0}^{N-1}$ (let's say) may be rotated by the verifier to obtain the remaining terms (in some still-unknown order). In fact, in each step we rotate both $(p_i(w))_{i=0}^{N-1}$ and $(q_i(w))_{i=0}^{N-1}$ by *two positions*; in precisely the case that N is even and idx_0 and idx_1 have opposite parity, this approach then yields each element of the original list exactly once. That the parities of the secret indices idx_0 and idx_1 are indeed opposite may be proven using variants of ideas which are already present in [GK15] and [BCC+15].

We thus construct a secret permutation $\kappa\colon \{0, \ldots, N-1\} \to \{0, \ldots, N-1\}$ of a special form. We begin with secret elements idx_0 and idx_1 of opposite parities; for arbitrary i, we then have that:

$$i \mapsto \kappa(i) = \begin{cases} (\mathsf{idx}_0 + 2 \cdot k) \mod N & \text{if } i = 2 \cdot k \\ (\mathsf{idx}_1 + 2 \cdot k) \mod N & \text{if } i = 2 \cdot k + 1. \end{cases}$$

We observe that such permutations κ are exactly those residing in the subgroup of $\mathbf{S}_N$ described by the generators $\langle (0, 1, 2, \ldots, N-1), (0, 2, 4, \ldots, N-2) \rangle$. In fact, these generators are "tight", in the sense that the natural map $\langle (0, 1, 2, \ldots, N-1) \rangle \times \langle (0, 2, 4, \ldots, N-2) \rangle \to \mathbf{S}_N$ is an injection.

A further way to understand these permutations is through permutation *matrices*. Effectively, we send only the top two rows of an unknown matrix; by performing two-step rotations, we obtain the remaining $N-2$ rows. The ultimate matrix is a permutation matrix if and only if the top two rows attain the value 1 at indices of opposite parity. This is described in the figures below:

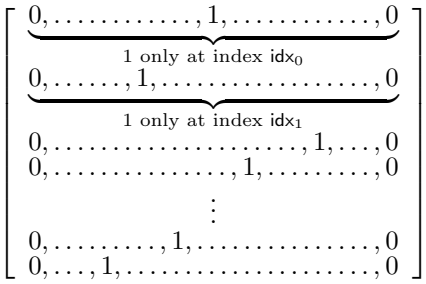

Figure 1: "Prover's view".

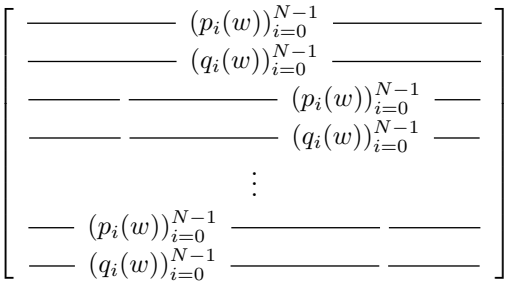

Figure 2: "Verifier's view".

The intuition behind our technique is that the prover doesn't *need* the freedom of choosing an arbitrary permutation. Indeed, as far as the prover and verifier are concerned, two permutations are "essentially the same" as long as they agree at the points 0 and 1. The prover and verifier may thus agree in advance on a scheme which, given prescribed (and secret!) values only at 0 and 1, constructs from these a full permutation.

We remark finally upon our choice of parameters in [BCC+15, Fig. 4, Fig. 5] (where the notation $N = n^m$ is used). [BCC+15] typically takes $n = \mathcal{O}(1), m = \mathcal{O}(\log N)$, though [BCC+15, p. 13] mentions the choice $m = \mathcal{O}(1)$. Indeed, this latter choice requires transmitting only "a constant number of group elements"; [BCC+15] cites its utility in settings where group elements are large compared to field elements.

Strikingly, this setting mirrors ours, in that we reuse our field elements, whereas we have to transmit

group elements for each among our $\mathcal{O}(N)$ re-encryptions. We accordingly adopt the choice $n = N, m = 2$, yielding $\mathcal{O}(N)$-sized proofs.

3.2 Circular convolutions and the number-theoretic transform

The main *computational* bottleneck of the above scheme concerns the matrix of Fig. 2 above. Indeed, in practice the verifier must "multiply" the vectors $(C_i)_{i=0}^{N-1}$ and $(y_i)_{i=0}^{N-1}$ of curve points by this matrix of scalars (we view the elliptic curve point group $\mathbb{G}$ as a module over $\mathbb{F}_q$). The prover too has to perform a similar matrix multiplication, in the process of deriving the correction terms. In fact, the above scheme implemented *naïvely* requires $\mathcal{O}(N^2)$ computation for both the prover and the verifier.

Our second key observation is that the matrix of Fig. 2 is a "circulant" matrix, or rather the interleaving of the even-indexed rows of two such matrices. The multiplication of $(C_i)_{i=0}^{N-1}$ (say) by this matrix, then, is exactly an interleaving of (the even indices of) two discrete circular convolutions—namely of $(C_i)_{i=0}^{N-1}$ by the vectors $(p_i(w))_{i=0}^{N-1}$ and $(q_i(w))_{i=0}^{N-1}$, respectively. This matrix multiplication can thus be evaluated in $\mathcal{O}(N \log N)$ time using the number-theoretic transform (or rather two of them), provided that N is a power of 2 and $\mathbb{F}_q$'s 2-adic roots of unity number at least N. Fortuitously, the curve used in Ethereum precompiles has a prime order q for which the 2-power-torsion of $(\mathbb{F}_q)^*$ has order 2^{28}—more than sufficient for our purposes. Indeed, this curve was selected with FFTs in mind.

These ideas originated with Pollard [Pol71], and are surveyed in [Nus82, §8]. In both settings, only the convolution of two *field-element* vectors is discussed. Our version—in which a vector of *module* elements (i.e., curve points) is convolved with a vector of field elements—appears to be absent from the literature, despite our having made a cursory search; in any case it naturally complements well-known techniques.

The authors of [BAZB] claim that $\mathcal{O}(\log N)$ communication and $\mathcal{O}(N)$ computation are possible. We are not able to achieve these asymptotics.

4 Security Proofs

In this section, we *sketch* proofs of soundness and zero-knowledge. These shouldn't yet be considered fully rigorous.

4.1 Soundness

Following the soundness proofs [BCC+15, §B.1] and [GK15, Thm. 2], we argue, after running a successful prover against two values of w, that each response $(f_{j,i})_{j,i=0}^{1,N-1}$ takes the form $f_{j,i} = q_{j,i} \cdot w + p_{j,i}$, for bits $q_{j,i} \in \{0,1\}$ for which $\sum_{i=0}^{N-1} q_{j,i} = 1$ (for each $j \in \{0,1\}$). We henceforth denote, for each $j \in \{0,1\}$, by idx_j that element $i \in \{0,\ldots,N-1\}$ for which $q_{j,i} = 1$.

We argue in addition that:

Claim 1. $\mathsf{idx}_0 \not\equiv \mathsf{idx}_1 \mod 2$.

Proof. Again following [BCC+15, §B.1], we extract elements $(x_i, y_i)_{i\in\{0,1\}}$ which open X and Y, and for which, by the verification equation 98, $y_i \cdot w + x_i = (\sum_{k\equiv i \mod 2} f_{0,k}) \cdot (\sum_{k\equiv i \mod 2} f_{1,k})$ (for each $i \in \{0,1\}$). This right-hand side is a polynomial in w whose degree-2 term is exactly $(\sum_{k\equiv i \mod 2} q_{0,k}) \cdot (\sum_{k\equiv i \mod 2} q_{1,k})$. As in the proof of [BCC+15, §B.1], we assert from the equality of these polynomials at three values of w that this quadratic coefficient is 0 (for each i); we conclude that $\mathsf{idx}_0 \not\equiv \mathsf{idx}_1 \mod 2$. □

Defining $(s_i)_{i=0}^{N-1}$ and $(t_i)_{i=0}^{N-1}$ using q_0 and q_1, we conclude that $\sum_{i=0}^{N-1} s_i = \sum_{i=0}^{N-1} t_i = 1$ and that $s_{i_0} = t_{i_1} = 1 \implies i_0 \not\equiv i_1 \mod 2$.

Claim 2. *The prover possesses* sk *for which* $g^{\mathsf{sk}} = y_{\mathsf{idx}_0}$*, as well as elements* $b^*, b' \in \{0,\ldots,\mathsf{MAX}\}$ *for which* $g^{b^*} \cdot D^{\mathsf{sk}} = C_{\mathsf{idx}_0}$ *and* $g^{b'} \cdot (C_{Rn,\mathsf{idx}_0})^{\mathsf{sk}} = C_{Ln,\mathsf{idx}_0}$.

Proof. We first fix w, y, z, and x. For each such choice, the extractor, by running the standard Schnorr extractor on two choices of c, may extract a value $\widehat{\mathsf{sk}}$ (not necessarily equal to sk) for which $\overline{g}^{\widehat{\mathsf{sk}}} = \overline{y_{0,0}}$. Similarly, we consider equation 113, reproduced below:

$$g^{w\cdot c\cdot \hat{t}} \cdot h^{w\cdot c\cdot \tau_x} \stackrel{?}{=} g^{w\cdot c\cdot \delta(y,z)} \cdot A_t \cdot {c_{\mathsf{commit}}}^{-1} \cdot \left(T_1^x \cdot T_2^{x^2}\right)^{w\cdot c},$$

where $c_{\mathsf{commit}} = \left((\overline{D}\cdot D')^{z^2} \cdot (\overline{C_{Rn}}\cdot C'_{Rn})^{z^3}\right)^{s_{\mathsf{sk}}} \cdot \left((\overline{C_{0,0}}\cdot C')^{z^2} \cdot (\overline{C_{Ln}}\cdot C'_{Ln})^{z^3}\right)^{-c}$. By subtracting two copies of this equation, the Schnorr extractor simultaneously obtains an equality:

$$g^{w\cdot \hat{t}} \cdot h^{w\cdot \tau_x} \stackrel{?}{=} g^{w\cdot \delta(y,z)} \cdot {V_1}^{z^2} \cdot {V_2}^{z^3} \cdot \left(T_1^x \cdot T_2^{x^2}\right)^{w},$$

where, for abbreviation, we write $V_1 = (\overline{D}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{0,0}}) \cdot (D'^{-\widehat{\mathsf{sk}}} \cdot C')$ and $V_2 = (\overline{C_{Rn}}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{Ln}}) \cdot (C'^{-\widehat{\mathsf{sk}}}_{Rn} \cdot C'_{Ln})$.

We now adapt the Bulletproofs extractor. Exactly as in [BBB$^+$, §C], by running the inner product extractor against two values of x, the extractor may obtain openings α, ρ, $\mathbf{a}_L$, $\mathbf{a}_R$, $\mathbf{s}_L$, and $\mathbf{s}_R$ of A and S; for each *particular* choice of x, the extractor obtains expressions for $\mathbf{l}$ and $\mathbf{r}$ of the form of 56 and 57.

Meanwhile, by obtaining values $\hat{t}$ and τ_x as above for three distinct values of x, the extractor may obtain openings t_1, t_2, τ_1, and τ_2 of T_1 and T_2, as well as openings b, γ for which $\left(g^b h^\gamma\right)^w = {V_1}^{z^2} \cdot {V_2}^{z^3}$. (We remark that V_1 and V_2 do not depend on z, y, x, or c.) As in [BBB$^+$, §C], by obtaining such openings b, γ for two distinct values of z, the extractor may calculate individual openings (b^*, γ^*) and (b', γ') for which $\left(g^{b^*} h^{\gamma^*}\right)^w = V_1$ and $\left(g^{b'} h^{\gamma'}\right)^w = V_2$. We thus establish, for 2 distinct challenges z and $2\cdot n$ distinct challenges y, the equality (at 3 values x) of the polynomials $t(X) = w \cdot \left(\delta(y,z) + z^2 \cdot b^* + z^3 \cdot b' + t_1 \cdot X + t_2 \cdot X^2\right)$ and $p(X) = w \cdot \langle l(X), r(X)\rangle$. Assuming now that w is nonzero, conclude as in [BBB$^+$, §C] b^* and b' reside in $\{0, \ldots, \mathsf{MAX}\}$.

Finally, the Schnorr extractor also obtains quantities v^*, v', ν^*, and ν' for which:

$$g^{v^*} = \overline{D}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{0,0}}, \quad g^{v'} = \overline{C_{Rn}}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{Ln}}, \quad h^{\nu^*} = D'^{-\widehat{\mathsf{sk}}} \cdot C', \quad h^{\nu'} = C'^{-\widehat{\mathsf{sk}}}_{Rn} \cdot C'_{Ln}.$$

By comparing the openings (b^*, γ^*) and (b', γ') of V_1 and V_2 with the representations above, we derive from the uniqueness of Pedersen representations the *individual* equalities:

$$g^{b^*\cdot w} = \overline{D}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{0,0}},$$
$$g^{b'\cdot w} = \overline{C_{Rn}}^{-\widehat{\mathsf{sk}}} \cdot \overline{C_{Ln}}.$$

We thus obtain, for *each* choice w, an element $\widehat{\mathsf{sk}}$, and equalities as above. We finally run the one-out-of-many extractor. As in the proofs [GK15, Thm. 3] and [BCC$^+$15, §B.2], after running the prover for three choices of w, the extractor may construct, for *each particular* w, representations:

$$(\overline{y_{0,0}}, \overline{g}) = \left(y^w_{\mathsf{idx}_0} \cdot \widehat{y_{0,0}}, g^w \cdot \widetilde{g}^{-1}\right),$$
$$\left(\overline{C_{0,0}}, \overline{D}\right) = \left(C^w_{\mathsf{idx}_0} \cdot \widehat{C_{0,0}}, D^w \cdot \widetilde{D}^{-1}\right),$$
$$\left(\overline{C_{Ln}}, \overline{C_{Rn}}\right) = \left(C^w_{Ln,\mathsf{idx}_0} \cdot \widehat{C_{Ln}}, C^w_{Rn,\mathsf{idx}_0} \cdot \widehat{C_{Rn}}\right),$$

where $\widehat{y_{0,0}}$, $\widehat{C_{0,0}}$, $\widehat{C_{Ln}}$, and $\widehat{C_{Rn}}$ are easily computed by the extractor and depend only on data sent by the prover before receiving w.

Choosing an arbitrary such w and obtaining a corresponding $\widehat{\mathsf{sk}}$ as above, we deduce the refined equalities:

$$y^w_{\mathsf{idx}_0} \cdot \widehat{y_{0,0}} = \left(g^w \cdot \widetilde{g}^{-1}\right)^{\widehat{\mathsf{sk}}},$$
$$\left(g^{-b^*} \cdot C_{\mathsf{idx}_0}\right)^w \cdot \widehat{C_{0,0}} = \left(D^w \cdot \widetilde{D}^{-1}\right)^{\widehat{\mathsf{sk}}},$$
$$\left(g^{-b'} \cdot C_{Ln,\mathsf{idx}_0}\right)^w \cdot \widehat{C_{Ln}} = \left(C^w_{Rn,\mathsf{idx}_0} \cdot \widehat{C_{Rn}}\right)^{\widehat{\mathsf{sk}}}.$$

Taking discrete logarithms with respect to g, the three identities above define (possibly degenerate) hyperbolic equations, with *unknown coefficients*, over $\mathbb{F}_q^2$, which the point $(w, \widehat{\mathsf{sk}})$ simultaneously satisfies:

$$1 \cdot w \cdot \widehat{\mathsf{sk}} - \log(y_{\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widetilde{g}^{-1}) \cdot \widehat{\mathsf{sk}} - \log(\widehat{y_{0,0}}) = 0, \tag{3}$$

$$\log(D) \cdot w \cdot \widehat{\mathsf{sk}} - \log(g^{-b^*} \cdot C_{\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widetilde{D}^{-1}) \cdot \widehat{\mathsf{sk}} - \log(\widehat{C_{0,0}}) = 0, \tag{4}$$

$$\log(C_{Rn,\mathsf{id}\mathsf{x}_0}) \cdot w \cdot \widehat{\mathsf{sk}} - \log(g^{-b'} \cdot C_{Ln,\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widehat{C_{Rn}}) \cdot \widehat{\mathsf{sk}} - \log(\widehat{C_{Ln}}) = 0. \tag{5}$$

Indeed, selecting three distinct values of w, the extractor may generate three such satisfying points, say $(w^{(i)}, \widehat{\mathsf{sk}}^{(i)})$ for $i \in \{1,2,3\}$. We argue that any three such points serve to uniquely determine the value of the coefficient $\log(y_{\mathsf{id}\mathsf{x}_0})$ of equation 3; we argue further that $\log(D) \cdot \log(y_{\mathsf{id}\mathsf{x}_0}) = \log(g^{-b^*} \cdot C_{\mathsf{id}\mathsf{x}_0})$ and $\log(C_{Rn,\mathsf{id}\mathsf{x}_0}) \cdot \log(y_{\mathsf{id}\mathsf{x}_0}) = \log(g^{-b'} \cdot C_{Ln,\mathsf{id}\mathsf{x}_0})$. These facts suffice to establish the statement of the Claim.

Consider first the case that the three points $(w^{(i)}, \widehat{\mathsf{sk}}^{(i)}), i \in \{1,2,3\}$ are colinear. (In this case, the hyperbolas degenerate into two lines.) Choosing scalars l_0, l_1 for which $\widehat{\mathsf{sk}}^{(i)} = l_1 \cdot w^{(i)} + l_0$ for each $i \in \{1,2,3\}$ (recall that the $w^{(i)}$ are distinct), we specialize the equations above to the line $\widehat{\mathsf{sk}} = l_1 \cdot w + l_0$:

$$w \cdot (l_1 \cdot w + l_0) - \log(y_{\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widetilde{g}^{-1}) \cdot (l_1 \cdot w + l_0) - \log(\widehat{y_{0,0}}) = 0, \tag{6}$$

$$\log(D) \cdot w \cdot (l_1 \cdot w + l_0) - \log(g^{-b^*} \cdot C_{\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widetilde{D}^{-1}) \cdot (l_1 \cdot w + l_0) - \log(\widehat{C_{0,0}}) = 0, \tag{7}$$

$$\log(C_{Rn,\mathsf{id}\mathsf{x}_0}) \cdot w \cdot (l_1 \cdot w + l_0) - \log(g^{-b'} \cdot C_{Ln,\mathsf{id}\mathsf{x}_0}) \cdot w + \log(\widehat{C_{Rn}}) \cdot (l_1 \cdot w + l_0) - \log(\widehat{C_{Ln}}) = 0. \tag{8}$$

Each specialization, a quadratic polynomial in w which attains the value 0 at the three distinct values $w^{(i)}$, is necessarily identically zero, with vanishing coefficients. We first study equation 6. Its quadratic coefficient establishes that $l_1 = 0$; from its linear coefficient, we conclude finally that $l_0 = \log(y_{\mathsf{id}\mathsf{x}_0})$. The extractor may thus set $\mathsf{sk} = l_0$. The linear coefficient of equation 7, in light of the equality $l_0 = \log(y_{\mathsf{id}\mathsf{x}_0})$, demonstrates that $\log(D) \cdot \log(y_{\mathsf{id}\mathsf{x}_0}) = \log(g^{-b^*} \cdot C_{\mathsf{id}\mathsf{x}_0})$. Equation 8 is treated similarly.

Suppose now that the points $(w^{(i)}, \widehat{\mathsf{sk}}^{(i)}), i \in \{1,2,3\}$ are not colinear. We express the coefficients $\log(y_{\mathsf{id}\mathsf{x}_0})$, $\log(\widetilde{g}^{-1})$ and $\log(\widehat{y_{0,0}})$ of equation 3 above through the matrix equation:

$$\begin{bmatrix} w^{(1)} \cdot \widehat{\mathsf{sk}}^{(1)} \\ w^{(2)} \cdot \widehat{\mathsf{sk}}^{(2)} \\ w^{(3)} \cdot \widehat{\mathsf{sk}}^{(3)} \end{bmatrix} = \begin{bmatrix} w^{(1)} & -\widehat{\mathsf{sk}}^{(1)} & 1 \\ w^{(2)} & -\widehat{\mathsf{sk}}^{(2)} & 1 \\ w^{(3)} & -\widehat{\mathsf{sk}}^{(3)} & 1 \end{bmatrix} \cdot \begin{bmatrix} \log(y_{\mathsf{id}\mathsf{x}_0}) \\ \log(\widetilde{g}^{-1}) \\ \log(\widehat{y_{0,0}}) \end{bmatrix}.$$

By hypothesis on the points $(w^{(i)}, \widehat{\mathsf{sk}}^{(i)})$, the 3-by-3 matrix above has non-vanishing determinant; by inverting it, the extractor may uniquely determine the value $\mathsf{sk} = \log(y_{\mathsf{id}\mathsf{x}_0})$ (alongside those of the other coefficients).

In fact, a slight variant of this matrix equation—in which its left-hand side is multiplied by the additional unknown scalar quantity $\log(D)$—exactly describes the coefficients of equation 4. We immediately conclude that $\log(D) \cdot \log(y_{\mathsf{id}\mathsf{x}_0}) = \log(g^{-b^*} \cdot C_{\mathsf{id}\mathsf{x}_0})$. The conclusion $\log(C_{Rn,\mathsf{id}\mathsf{x}_0}) \cdot \log(y_{\mathsf{id}\mathsf{x}_0}) = \log(g^{-b'} \cdot C_{Ln,\mathsf{id}\mathsf{x}_0})$ follows similarly. □

Claim 3. *The prover possesses r for which $g^r = D$. Moreover, $(y_{\mathsf{id}\mathsf{x}_0} \cdot y_{\mathsf{id}\mathsf{x}_1})^r = C_{\mathsf{id}\mathsf{x}_0} \cdot C_{\mathsf{id}\mathsf{x}_1}$, and, for each $i \notin \{\mathsf{id}\mathsf{x}_0, \mathsf{id}\mathsf{x}_1\}, y_i^r = C_i$.*

Proof. As in the proof of Claim 2, by fixing w and running the Schnorr extractor on two choices of c, the extractor may obtain some value $\widehat{r}$ for which, simultaneously,

$$\overline{D} = \overline{g}^{\widehat{r}},$$

$$\overline{C_{0,0}} \cdot \overline{C_{1,0}} = (\overline{y_{0,0}} \cdot \overline{y_{1,0}})^{\widehat{r}},$$

$$\left\{ \overline{C_{j,i}} = \overline{y_{j,i}}^{\widehat{r}} \right\}_{j=0,i=1}^{1,\frac{N}{2}-1}.$$

On the other hand, after running the prover on three choices of w, the extractor may generate, for any *particular* w, expressions:

$$(\overline{D}, \overline{g}) = \left(D^w \cdot \widetilde{D}^{-1}, g^w \cdot \widetilde{g}^{-1}\right),$$
$$\left\{(\overline{C_{j,i}}, \overline{y_{j,i}}) = \left(C^w_{\mathsf{idx}_j+2\cdot i} \cdot \widehat{C_{j,i}}, y^w_{\mathsf{idx}_j+2\cdot i} \cdot \widehat{y_{j,i}}\right)\right\}_{j,i=0}^{1,\frac{N}{2}-1},$$

for elements $\widehat{C_{j,i}}$ and $\widehat{y_{j,i}}$ which are easily computed by the extractor and depend only on data sent by the prover before receiving w.

Combining these equations, we derive the refined equalities:

$$D^w \cdot \widetilde{D}^{-1} = \left(g^w \cdot \widetilde{g}^{-1}\right)^{\widehat{r}},$$
$$(C_{\mathsf{idx}_0} \cdot C_{\mathsf{idx}_1})^w \cdot \left(\widehat{C_{0,0}} \cdot \widehat{C_{1,0}}\right) = ((y_{\mathsf{idx}_0} \cdot y_{\mathsf{idx}_1})^w \cdot (\widehat{y_{0,0}} \cdot \widehat{y_{1,0}}))^{\widehat{r}},$$
$$\left\{C^w_{\mathsf{idx}_j+2\cdot i} \cdot \widehat{C_{j,i}} = \left(y^w_{\mathsf{idx}_j+2\cdot i} \cdot \widehat{y_{j,i}}\right)^{\widehat{r}}\right\}_{j=0,i=1}^{1,\frac{N}{2}-1}.$$

Taking discrete logs with respect to g, these equalities induce a number of hyperbolic equations over $\mathbb{F}_q^2$, which the pair $(w, \widehat{r})$ simultaneously satisfies:

$$1 \cdot w \cdot \widehat{r} - \log(D) \cdot w + \log(\widetilde{g}^{-1}) \cdot \widehat{r} - \log(\widetilde{D}^{-1}) = 0, \tag{9}$$
$$\log(y_{\mathsf{idx}_0} \cdot y_{\mathsf{idx}_1}) \cdot w \cdot \widehat{r} - \log(C_{\mathsf{idx}_0} \cdot C_{\mathsf{idx}_1}) \cdot w + \log(\widehat{y_{0,0}} \cdot \widehat{y_{1,0}}) \cdot \widehat{r} - \log(\widehat{C_{0,0}} \cdot \widehat{C_{1,0}}) = 0, \tag{10}$$
$$\left\{\log(y_{\mathsf{idx}_j+2\cdot i}) \cdot w \cdot \widehat{r} - \log(C_{\mathsf{idx}_j+2\cdot i}) \cdot w + \log(\widehat{y_{j,i}}) \cdot \widehat{r} - \log(\widehat{C_{j,i}}) = 0\right\}_{j=0,i=1}^{1,\frac{N}{2}-1}. \tag{11}$$

As in the proof of Claim 2, we argue that the extractor, after generating three such simultaneous solutions $\left(w^{(i)}, \widehat{r}^{(i)}\right), i \in \{1,2,3\}$ (for distinct $w^{(i)}$), may uniquely determine the coefficient $\log(D)$ of equation 9; we argue furthermore that necessarily $\log(y_{\mathsf{idx}_0} \cdot y_{\mathsf{idx}_1}) \cdot \log(D) = \log(C_{\mathsf{idx}_0} \cdot C_{\mathsf{idx}_1})$ as well as that $\log(y_{\mathsf{idx}_j+2\cdot i}) \cdot \log(D) = \log(C_{\mathsf{idx}_j+2\cdot i})$ for each $j \in \{0,1\}$ and $i \in \{1, \ldots, \frac{N}{2}-1\}$. Finally, using Claim 1 we argue that the set $\left\{(C_{\mathsf{idx}_j+2\cdot i}, y_{\mathsf{idx}_j+2\cdot i})\right\}_{j=0,i=1}^{1,\frac{N}{2}-1}$ is in bijection with $\{(C_i, y_i)\}_{i \notin \{\mathsf{idx}_0, \mathsf{idx}_1\}}$. These facts together imply the statement of the Claim.

Indeed, we proceed exactly as in the proof of Claim 2. If the points $(w^{(i)}, \widehat{r}^{(i)})$ are colinear, the extractor may assign to r that unique value l_0 for which the linear equation $\widehat{r}^{(i)} = l_1 \cdot w^{(i)} + l_0$ holds (for each $i \in \{1,2,3\}$). Otherwise, the extractor may determine $r = \log(D)$ through the matrix expression:

$$\begin{bmatrix} w^{(1)} \cdot \widehat{r}^{(1)} \\ w^{(2)} \cdot \widehat{r}^{(2)} \\ w^{(3)} \cdot \widehat{r}^{(3)} \end{bmatrix} = \begin{bmatrix} w^{(1)} & -\widehat{r}^{(1)} & 1 \\ w^{(2)} & -\widehat{r}^{(2)} & 1 \\ w^{(3)} & -\widehat{r}^{(3)} & 1 \end{bmatrix} \cdot \begin{bmatrix} \log(D) \\ \log(\widetilde{g}^{-1}) \\ \log(\widetilde{D}^{-1}) \end{bmatrix},$$

where the matrix on the right-hand side is necessarily invertible. In either case, the equalities $\log(y_{\mathsf{idx}_0} \cdot y_{\mathsf{idx}_1}) \cdot \log(D) = \log(C_{\mathsf{idx}_0} \cdot C_{\mathsf{idx}_1})$, and $\log(y_{\mathsf{idx}_j+2\cdot i}) \cdot \log(D) = \log(C_{\mathsf{idx}_j+2\cdot i})$ straightforwardly follow. □

Remark 1. For a prover executing the protocol *as written*, the hyperbolas 3, 4 and 5 will all degenerate, each with a horizontal line at the height sk, and vertical lines at $\sigma_{0,0}$, $\sigma_{0,0}$, and π/r', respectively (where r' is the randomness of $(C_{Ln,\mathsf{idx}_0}, C_{Rn,\mathsf{idx}_0})$, generally not known to the prover). The hyperbolas 9, 10, and 11 will also degenerate, each with a horizontal line at r, and with vertical lines at σ_0, $\frac{\sigma_{0,0}\cdot\mathsf{sk}+\sigma_{1,0}\cdot\log(y_{\mathsf{idx}_1})}{\mathsf{sk}+\log(y_{\mathsf{idx}_1})}$, and $\sigma_{j,i}$, respectively ($\log(y_{\mathsf{idx}_1})$ is of course generally unknown to the prover). If any equation's leading coefficient vanishes, its zero locus will lack a vertical line.

Yet a successful prover may act so as to use *non-degenerate* hyperbolas (though apparently at the expense of zero knowledge), and indeed the "non-colinear" treatments of Claims 2 and 3 are necessary. For instance, suppose that a prover were to choose each $\sigma_{j,i} = \sigma$ identically, and to use a second, distinct

element ($\widehat{\sigma}$, say) in the construction of $\widetilde{D}$ and $\widetilde{C_{j,i}}$. In this case, 9, 10, and 11 would become non-degenerate, each with a "horizontal asymptote" at r and a "vertical asymptote" at σ (they'd take the form $(\widehat{r} - r)(w - \sigma) = r \cdot (\sigma - \widehat{\sigma})$). Meanwhile, 4's vertical line would move to $\widehat{\sigma}$.

Using other modifications, a prover could make 3, 4 and 5 non-degenerate. We leave the study of these contingencies to the reader. In all cases, the extractor will still succeed.

Remark 2. A prover operating as written could still handle the negligibly probable choice $w = \sigma_0$—simply by setting $\widehat{\mathsf{sk}}$ and $\widehat{r}$ at the height of the horizontal line, as usual (i.e., at the "cross"). In fact, under the special case $\sigma_{j,i} = \sigma$ considered above, the prover could even choose $\widehat{r}$ *arbitrarily* (on the "common vertical line"). If this occurred, the extractor would enter the "non-colinear" second case of Claim 3, despite the hyperbolas' degeneration; it would still succeed.

If in addition the prover were operating so that 9, 10, and 11 (say) were non-degenerate, it would become unable to respond to the challenge $w = \sigma_0$, and would necessarily fail verification or abort.

4.2 Zero Knowledge

We describe a simulator which outputs accepting transcripts; we argue in steps that these transcripts are indistinguishable from actual ones.

Claim 4. *The elements $(f_{j,i})_{j,i=0}^{1,\frac{N}{2}-1}$, P, Q, U, V, X, Y, z_P, z_U, z_X, and w can be simulated perfectly indistinguishably.*

Proof. As in the one-out-of-many simulator [BCC+15, §B.1], our simulator randomly chooses $(f_{j,i})_{j=0,i=1}^{1,\frac{N}{2}-1}$, Q, U, Y, z_P, z_U, z_X, and w; it then sets $f_{j,0} = w - \sum_{i=1}^{N-1} f_{j,i}$ for $j \in \{0, 1\}$, as well as:

$$P = Q^{-w} \cdot \mathrm{Com}(f_{0,0}, \ldots, f_{1,N-1}; z_P),$$
$$V = U^{-w} \cdot \mathrm{Com}((f_{j,i}(w - f_{j,i}))_{j,i=0}^{1,N-1}; z_U),$$
$$X = Y^{-w} \cdot \mathrm{Com}(((\textstyle\sum_{k \equiv i \mod 2} f_{0,k}) \cdot (\sum_{k \equiv i \mod 2} f_{1,k}))_{i \in \{0,1\}}; z_X).$$

We argue just as in [BCC+15, §B.1] that this simulation is perfect. □

Claim 5. *The elements $\hat{t}$, τ_x, μ, A, S, T_1, T_2, $\mathbf{l}$, $\mathbf{r}$, z, y, and x can be simulated perfectly indistinguishably.*

Proof. We adapt the Bulletproofs simulator [BBB+, §C]. Our simulator randomly selects $\hat{t}$, τ_x, μ, S, T_1, T_2, $\mathbf{l}$, $\mathbf{r}$, z, y, and x, and sets:

$$A = h^{\mu} \cdot \mathbf{g}^{\mathbf{l}} \cdot \mathbf{h'}^{\mathbf{r}} \cdot \left(S^{x} \cdot \mathbf{g}^{-z} \cdot \mathbf{h'}^{z \cdot \mathbf{y}^{2 \cdot n}} \cdot \mathbf{h'}^{z^2 \cdot (\mathbf{2}^n \| \mathbf{0}^n) + z^3 \cdot (\mathbf{0}^n \| \mathbf{2}^n)} \right)^{-1}.$$

The simulator also runs the inner product argument with the verifier, using the simulated witnesses $\mathbf{l}$ and $\mathbf{r}$. As in [BBB+, §C], we argue that the elements generated in this way are distributed identically to those in actual transcripts. □

Claim 6. *The elements (C', D') and (C'_{Ln}, C'_{Rn}), $A_{C'}$, and $A_{C'_{Ln}}$, and finally s_{ν^*}, $s_{\nu'}$, and c can be simulated computationally indistinguishably.*

Proof. The simulator may generate (C', D') and (C'_{Ln}, C'_{Rn}) randomly; by the Diffie–Hellman assumption, their analogues in real proofs (ElGamal ciphertexts) are indistinguishable from random elements of $\mathbb{G}^2$. The simulator may also generate s_{ν^*}, $s_{\nu'}$, and c randomly. Finally, as in the Schnorr simulator, these quantities uniquely determine the remaining values $A_{C'}$, and $A_{C'_{Ln}}$. □

Claim 7. *The elements s_{sk}, s_r, s_{v^*}, $s_{v'}$, and c, as well as $\widetilde{C_{Ln}}$, $\widetilde{C_{Rn}}$, $\left(\widetilde{C_{j,i}}, \widetilde{y_{j,i}}\right)_{j,i=0}^{1,\frac{N}{2}-1}$, $\widetilde{D}$, and $\widetilde{g}$, and finally A_y, A_D, A_u, A_B, $\left(A_{C_{j,i}}\right)_{j=0,i=1}^{1,\frac{N}{2}-1}$, A_t, $A_{\overline{C_{0,0}}}$, $A_{\overline{C_{Ln}}}$, can be simulated computationally indistinguishably.*

Proof. The simulator randomly generates $\overline{C_{Ln}}$, $\overline{C_{Rn}}$, $\left(\overline{C_{j,i}}, \overline{y_{j,i}}\right)_{j,i=0}^{1,\frac{N}{2}-1}$, $\overline{D}$, and $\overline{g}$. These values, together with those of w and $(f_{j,i})_{j,i=0}^{1,\frac{N}{2}-1}$, uniquely determine $\widetilde{C_{Ln}}$, $\widetilde{C_{Rn}}$, $\left(\widetilde{C_{j,i}}, \widetilde{y_{j,i}}\right)_{j,i=0}^{1,\frac{N}{2}-1}$, $\widetilde{D}$, and $\widetilde{g}$. Finally, the simulator randomly generates s_{sk}, s_r, s_{v^*}, $s_{v'}$, and c (we assume that c is generated consistently with respect to the above). These further values, together with those generated in the above claims, and finally the verification equations, uniquely determine the remaining quantities A_y, A_D, A_u, A_B, $\left(A_{C_{j,i}}\right)_{j=0,i=1}^{1,\frac{N}{2}-1}$, A_t, $A_{\overline{C_{0,0}}}$, and $A_{\overline{C_{Ln}}}$.

We argue that transcripts generated in this way are indistinguishable from actual transcripts. In real proofs, we have the expression:

$$\left(\overline{C_{Ln}}, \overline{C_{Rn}}\right) = \left(g^{w \cdot b'} \cdot y_{\mathsf{idx}_0}^{w \cdot r' - \pi}, g^{w \cdot r' - \pi}\right),$$

for some quantity r' (the randomness of $(C_{Ln,\mathsf{idx}_0}, C_{Rn,\mathsf{idx}_0})$) generally unknown to the prover. As the secret scalar π varies, $\left(\overline{C_{Ln}}, \overline{C_{Rn}}\right)$ varies uniformly throughout the ElGamal encryptions of $g^{w \cdot b'}$ under y_{idx_0}; by the decisional Diffie–Hellman assumption, this ciphertext can't be efficiently distinguished from a random element of $\mathbb{G}^2$.

We now fix values $j \in \{0, 1\}$ and $i \in \{0, \ldots, \frac{N}{2} - 1\}$. For each such choice, we have in real proofs the expression:

$$\left(\overline{C_{j,i}}, \overline{y_{j,i}}\right) = \left(g^{w \cdot \widehat{b}} \cdot \left(y_{\mathsf{idx}_j + 2 \cdot i}^{w - \sigma_{j,i}}\right)^r, y_{\mathsf{idx}_j + 2 \cdot i}^{w - \sigma_{j,i}}\right),$$

where $\widehat{b}$ is either b^*, $-b^*$, or 0 (depending on j and i). This pair is an ElGamal "encryption" of $g^{w \cdot \widehat{b}}$ under the "public key" $g^r = D$; moreover, its randomness depends bijectively and uniformly on the secret quantity $\sigma_{j,i}$. We conclude that no adversary can distinguish it from a uniform element of $\mathbb{G}^2$.

In fact, across values of j and i, the ciphertexts $\left(\overline{C_{j,i}}, \overline{y_{j,i}}\right)$ under D feature randomnesses which are secret, uniform, and *independent* (each depending on $\sigma_{j,i}$); we conclude that no adversary can distinguish them collectively from a random element of $(\mathbb{G}^2)^N$.

Alongside $\left(\overline{C_{0,0}}, \overline{y_{0,0}}\right)$, real proofs feature a final pair:

$$\left(\overline{D}, \overline{g}\right) = \left((g^{w - \sigma_{0,0}})^r, g^{w - \sigma_{0,0}}\right).$$

This too gives a uniformly random ElGamal encryption (of the identity) under the public key D; we argue as above that it is indistinguishable from a random element of $\mathbb{G}^2$. It remains only to argue that the joint distribution of $\left(\overline{C_{0,0}}, \overline{y_{0,0}}\right)$ and $\left(\overline{D}, \overline{g}\right)$ is indistinguishable from the uniform distribution on $\mathbb{G}^2 \times \mathbb{G}^2$. Yet in light of the expressions

$$\left(\overline{y_{0,0}}, \overline{g}\right) = \left(y_{\mathsf{idx}_0}^{w - \sigma_{0,0}}, g^{w - \sigma_{0,0}}\right)$$

and

$$\left(\overline{C_{0,0}}, \overline{D}\right) = \left(g^{w \cdot b^*} y_{\mathsf{idx}_0}^{r \cdot (w - \sigma_{0,0})}, g^{r \cdot (w - \sigma_{0,0})}\right),$$

any adversary distinguishing these latter distributions would also distinguish from random the Diffie–Hellman quadruple $\left(\overline{g}, \overline{y_{0,0}}, \overline{D}, g^{-b^* \cdot w} \cdot \overline{C_{0,0}}\right)$. We conclude that these terms too are indistinguishable from random elements. □

5 Protocol Specification

We now specify in detail a zero-knowledge proof protocol for the statement $\mathsf{st}_{\mathsf{AnonTransfer}}$ 1 above. We denote by n that integer for which $\mathsf{MAX} = 2^n - 1$, and by idx_0 and idx_1 those indices for which $s_{\mathsf{idx}_0} = 1$ and $t_{\mathsf{idx}_1} = 1$, respectively. We define and make use of the following functions:

- $\mathsf{Shift}(\mathbf{v}, i)$ circularly shifts the vector $\mathbf{v}$ of field elements (i.e., $\mathbb{F}_q$) by the integer i.
- $\mathsf{MultiExp}(\mathbf{V}, \mathbf{v})$ multi-exponentiates the vector $\mathbf{V}$ of curve points by the vector $\mathbf{v}$ of field elements.

We mark in blue font those steps which do not appear in [BAZB], [BBB+], or [BCC+15].

Protocol Anonymous Zether

$\mathcal{P}$ **computes...**

$\alpha, \rho \leftarrow_\$ \mathbb{Z}_q$ ▷ Begin Bulletproof [BBB+, §4]

$\mathbf{a}_L \in \{0,1\}^{2 \cdot n}$ s.t. $\langle \mathbf{a}_{L[:n]}, \mathbf{2}^n \rangle = b^*, \langle \mathbf{a}_{L[n:]}, \mathbf{2}^n \rangle = b'$

$\mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^{2 \cdot n}$

$A = h^\alpha \mathbf{g}^{\mathbf{a}_L} \mathbf{h}^{\mathbf{a}_R}$

$\mathbf{s}_L, \mathbf{s}_R \leftarrow_\$ \mathbb{Z}_q^{2 \cdot n}$

$S = h^\rho \mathbf{g}^{\mathbf{s}_L} \mathbf{h}^{\mathbf{s}_R}$

$r_P, r_Q, r_U, r_V, r_X, r_Y, \pi, (\sigma_{j,i})_{j,i=0}^{1,\frac{N}{2}-1} \leftarrow_\$ \mathbb{Z}_q$ ▷ Begin one-out-of-many proof [BCC+15]

for all $j \in \{0,1\}$ **do**

sample $p_{j,1}, \ldots, p_{j,N-1} \leftarrow_\$ \mathbb{Z}_q$, set $p_{j,0} = -\sum_{i=1}^{N-1} p_{j,i}$

end for

set $(q_{0,0}, \ldots, q_{1,N-1}) = (s_0, \ldots, s_{N-1}, t_0, \ldots, t_{N-1})$

$P = \mathrm{Com}(p_{0,0}, \ldots, p_{1,N-1}; r_P)$

$Q = \mathrm{Com}(q_{0,0}, \ldots, q_{1,N-1}; r_Q)$

$U = \mathrm{Com}((p_{j,i}(1 - 2q_{j,i}))_{j,i=0}^{1,N-1}; r_U)$

$V = \mathrm{Com}(-p_{0,0}^2, \ldots, -p_{1,N-1}^2; r_V)$

$X = \mathrm{Com}\left(((\sum_{k \equiv i \mod 2} p_{0,k}) \cdot (\sum_{k \equiv i \mod 2} p_{1,k}))_{i \in \{0,1\}}, r_X\right)$

$Y = \mathrm{Com}\left((\sum_{k \equiv i \mod 2} p_{\mathsf{idx}_i \not\equiv i \mod 2, k})_{i \in \{0,1\}}, r_Y\right)$ ▷ $\mathsf{idx}_i \not\equiv i \mod 2$ is interpreted as a bit.

$\widetilde{C_{Ln}} = \mathsf{MultiExp}\left((C_{Ln,i})_{i=0}^{N-1}, p_0\right) \cdot (y_{\mathsf{idx}_0})^\pi$

$\widetilde{C_{Rn}} = \mathsf{MultiExp}\left((C_{Rn,i})_{i=0}^{N-1}, p_0\right) \cdot g^\pi$

for all $j \in \{0,1\}, i \in \{0, \ldots, \frac{N}{2} - 1\}$ **do**

$\widetilde{C_{j,i}} = \mathsf{MultiExp}\left((C_k)_{k=0}^{N-1}, \mathsf{Shift}(p_j, 2 \cdot i)\right) \cdot \left(y^r_{\mathsf{idx}_j + 2 \cdot i}\right)^{\sigma_{j,i}}$

$\widetilde{y_{j,i}} = \mathsf{MultiExp}\left((y_k)_{k=0}^{N-1}, \mathsf{Shift}(p_j, 2 \cdot i)\right) \cdot \left(y_{\mathsf{idx}_j + 2 \cdot i}\right)^{\sigma_{j,i}}$

end for

$\widetilde{D} = D^{\sigma_{0,0}}$

$\widetilde{g} = g^{\sigma_{0,0}}$

end $\mathcal{P}$

$\mathcal{P} \to \mathcal{V} : A, S, P, Q, U, V, X, Y, \widetilde{C_{Ln}}, \widetilde{C_{Rn}}, \left(\widetilde{C_{j,i}}, \widetilde{y_{j,i}}\right)_{j,i=0}^{1,\frac{N}{2}-1}, \widetilde{D}, \widetilde{g}$

$\mathcal{V} : w \leftarrow_\$ \mathbb{Z}_q$

$\mathcal{V} \to \mathcal{P} : w$

$\mathcal{P}$ **sets...**

for all $j \in \{0,1\}, i \in \{0, \ldots, N-1\}$ **do**

$f_{j,i} = q_{j,i} \cdot w + p_{j,i}$

end for

$z_P = r_Q \cdot w + r_P$

$z_U = r_U \cdot w + r_V$

$z_X = r_Y \cdot w + r_X$

$\gamma^*, \gamma', \zeta^*, \zeta' \leftarrow_\$ \mathbb{Z}_q$ ▷ Begin computation of blinding ciphertexts

$(C', D') = \left(h^{\gamma^* \cdot w} \cdot \overline{y_{0,0}}^{\zeta^*}, \overline{g}^{\zeta^*}\right)$

$(C'_{Ln}, C'_{Rn}) = \left(h^{\gamma' \cdot w} \cdot \overline{y_{0,0}}^{\zeta'}, \overline{g}^{\zeta'}\right)$

end $\mathcal{P}$

$\mathcal{P} \to \mathcal{V} : (f_{j,1}, \ldots, f_{j,N})$ **for** $j \in \{0,1\}, z_P, z_U, z_X, C', D', C'_{Ln}, C'_{Rn}$

$\mathcal{V} : y, z \leftarrow_\$ \mathbb{Z}_q$

$\mathcal{V} \to \mathcal{P} : y, z$

$\mathcal{P}$ **:**

$l(X) = (\mathbf{a}_L - z \cdot \mathbf{1}^n) + \mathbf{s}_L \cdot X$

$r(X) = \mathbf{y}^n \circ (\mathbf{a}_R + z \cdot \mathbf{1}^n + \mathbf{s}_R \cdot X) + z^2 \cdot (\mathbf{2}^n \parallel \mathbf{0}^n) + z^3 \cdot (\mathbf{0}^n \parallel \mathbf{2}^n)$

$t(X) = \langle l(X), r(X) \rangle = t_0 + t_1 \cdot X + t_2 \cdot X^2$ ▷ l and r are elements of $\mathbb{Z}_q^{2 \cdot n}[X]$; $t \in \mathbb{Z}_q[X]$

$\tau_1, \tau_2 \leftarrow_\$ \mathbb{Z}_q$

$T_i = g^{t_i} h^{\tau_i}$ **for** $i \in \{1, 2\}$

end $\mathcal{P}$

$\mathcal{P} \rightarrow \mathcal{V} : T_1, T_2$

$\mathcal{V} : x \leftarrow_\$ \mathbb{Z}_q$

$\mathcal{V} \rightarrow \mathcal{P} : x$

$\mathcal{P}$ **sets...**

$\mathbf{l} = l(x) = \mathbf{a}_L - z \cdot \mathbf{1}^{2 \cdot n} + \mathbf{s}_L \cdot x$

$\mathbf{r} = r(x) = \mathbf{y}^{2 \cdot n} \circ (\mathbf{a}_R + z \cdot \mathbf{1}^{2 \cdot n} + \mathbf{s}_R \cdot x) + z^2 \cdot (\mathbf{2}^n \parallel \mathbf{0}^n) + z^3 \cdot (\mathbf{0}^n \parallel \mathbf{2}^n)$

$\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$ ▷ $\mathbf{l}$ and $\mathbf{r}$ are elements of $\mathbb{Z}_q^{2 \cdot n}$; $\hat{t} \in \mathbb{Z}_q$

$\tau_x = \tau_2 \cdot x^2 + \tau_1 \cdot x + z^2 \cdot \gamma^* + z^3 \cdot \gamma'$

$\mu = \alpha + \rho \cdot x$

$\overline{C_{Rn}} = (C_{Rn,\mathsf{idx}_0})^w \cdot g^{-\pi}$ ▷ Prover "anticipates" certain secondary re-encryptions

for $j \in \{0, 1\}, i \in \left\{0, \ldots, \frac{N}{2} - 1\right\}$ **do**

$\overline{y_{j,i}} = \left(y_{\mathsf{idx}_j + 2i}\right)^{w - \sigma_{j,i}}$

end for

$\overline{D} = D^{w - \sigma_{0,0}}$

$\overline{g} = g^{w - \sigma_{0,0}}$

$A_y = \overline{g}^{k_{\mathsf{sk}}}$ ▷ Begin sigma protocol proving

$A_D = \overline{g}^{k_r}$

$A_u = g_{\mathsf{epoch}}^{k_{\mathsf{sk}}}$

$A_B = (\overline{y_{0,0}} \cdot \overline{y_{1,0}})^{k_r}$

$A_t = \left((\overline{D} \cdot D')^{z^2} \cdot (\overline{C_{Rn}} \cdot C'_{Rn})^{z^3} \right)^{k_{\mathsf{sk}}}$

for all $j \in \{0, 1\}, i \in \left\{1, \ldots, \frac{N}{2} - 1\right\}$ **do**

$A_{C_{j,i}} = (\overline{y_{j,i}})^{k_r}$

end for

$A_{\overline{C_{0,0}}} = g^{k_{v^*}} \cdot \overline{D}^{k_{\mathsf{sk}}}$

$A_{\overline{C_{Ln}}} = g^{k_{v'}} \cdot \overline{C_{Rn}}^{k_{\mathsf{sk}}}$

$A_{C'} = h^{k_{\nu^*}} \cdot D'^{k_{\mathsf{sk}}}$

$A_{C'_{Ln}} = h^{k_{\nu'}} \cdot C'^{k_{\mathsf{sk}}}_{Rn}$

end $\mathcal{P}$

$\mathcal{P} \rightarrow \mathcal{V} : \hat{t}, \tau_x, \mu, A_y, A_D, A_u, A_B, \left(A_{C_{j,i}}\right)_{j=0,i=1}^{1, \frac{N}{2} - 1}, A_t, A_{\overline{C_{0,0}}}, A_{\overline{C_{Ln}}}, A_{C'}, A_{C'_{Ln}}$

$\mathcal{V} : c \leftarrow_\$ \mathbb{Z}_q$

$\mathcal{V} \rightarrow \mathcal{P} : c$

$\mathcal{P}$ **sets...**

$s_{\mathsf{sk}} = k_{\mathsf{sk}} + c \cdot \mathsf{sk}$

$s_r = k_r + c \cdot r$

$s_{v^*} = k_{v^*} + c \cdot w \cdot b^*$

$s_{v'} = k_{v'} + c \cdot w \cdot b'$

$s_{\nu^*} = k_{\nu^*} + c \cdot w \cdot \gamma^*$

$s_{\nu'} = k_{\nu'} + c \cdot w \cdot \gamma'$

end $\mathcal{P}$

$\mathcal{P} \rightarrow \mathcal{V} : s_{\mathsf{sk}}, s_r, s_{v^*}, s_{v'}, s_{\nu^*}, s_{\nu'}$

$\mathcal{V}$ **:**

for all $j \in \{0, 1\}$ **do**

set $f_{j,0} = w - \sum_{i=1}^{N-1} f_{j,i}$

end for

96: $Q^w P \stackrel{?}{=} \mathrm{Com}(f_{0,0}, \ldots, f_{1,N-1}; z_P)$

97: $U^w V \stackrel{?}{=} \mathrm{Com}((f_{j,i}(w - f_{j,i}))_{j,i=0}^{1,N-1}; z_U)$

98: $Y^w X \stackrel{?}{=} \mathrm{Com}(((\sum_{k \equiv i \mod 2} f_{0,k}) \cdot (\sum_{k \equiv i \mod 2} f_{1,k}))_{i \in \{0,1\}}; z_X)$ ▷ Opposite parity check

99: $\overline{C_{Ln}} = \mathsf{MultiExp}\left((C_{Ln,i})_{i=0}^{N-1}, f_0\right) \cdot (\widetilde{C_{Ln}})^{-1}$ ▷ Begin computation of secondary re-encryptions

100: $\overline{C_{Rn}} = \mathsf{MultiExp}\left((C_{Rn,i})_{i=0}^{N-1}, f_0\right) \cdot (\widetilde{C_{Rn}})^{-1}$

101: **for all** $j \in \{0,1\}, i \in \{0, \ldots, \frac{N}{2} - 1\}$ **do**

102: $\overline{C_{j,i}} = \mathsf{MultiExp}\left((C_k)_{k=0}^{N-1}, \mathsf{Shift}(f_j, 2 \cdot i)\right) \cdot (\widetilde{C_{j,i}})^{-1}$

103: $\overline{y_{j,i}} = \mathsf{MultiExp}\left((y_k)_{k=0}^{N-1}, \mathsf{Shift}(f_j, 2 \cdot i)\right) \cdot (\widetilde{y_{j,i}})^{-1}$

104: **end for**

105: $\overline{D} = D^w \cdot \widetilde{D}^{-1}$

106: $\overline{g} = g^w \cdot \widetilde{g}^{-1}$

107: $A_y \stackrel{?}{=} \overline{g}^{s_{\mathsf{sk}}} \cdot (\overline{y_{0,0}})^{-c}$ ▷ Begin sigma protocol verification

108: $A_D \stackrel{?}{=} \overline{g}^{s_r} \cdot \overline{D}^{-c}$

109: $A_u \stackrel{?}{=} g_{\mathsf{epoch}}^{s_{\mathsf{sk}}} \cdot u^{-c}$

110: $A_B \stackrel{?}{=} (\overline{y_{0,0}} \cdot \overline{y_{1,0}})^{s_r} \cdot \left(\overline{C_{0,0}} \cdot \overline{C_{1,0}}\right)^{-c}$

111: $\delta(y,z) = (z - z^2) \cdot \langle \mathbf{1}^{2 \cdot n}, \mathbf{y}^{2 \cdot n} \rangle - \left(z^3 \cdot \langle \mathbf{1}^n, \mathbf{2}^n \rangle + z^4 \cdot \langle \mathbf{1}^n, \mathbf{2}^n \rangle\right)$

112: $c_{\mathsf{commit}} = \left((\overline{D} \cdot D')^{z^2} \cdot (\overline{C_{Rn}} \cdot C'_{Rn})^{z^3}\right)^{s_{\mathsf{sk}}} \cdot \left((\overline{C_{0,0}} \cdot C')^{z^2} \cdot (\overline{C_{Ln}} \cdot C'_{Ln})^{z^3}\right)^{-c}$

113: $g^{w \cdot c \cdot \hat{t}} \cdot h^{w \cdot c \cdot \tau_x} \stackrel{?}{=} g^{w \cdot c \cdot \delta(y,z)} \cdot A_t \cdot {c_{\mathsf{commit}}}^{-1} \cdot \left(T_1^x \cdot T_2^{x^2}\right)^{w \cdot c}$

114: **for all** $j \in \{0,1\}, i \in \{1, \ldots, \frac{N}{2} - 1\}$ **do**

115: $A_{C_{j,i}} \stackrel{?}{=} \overline{y_{j,i}}^{s_r} \cdot \overline{C_{j,i}}^{-c}$

116: **end for**

117: $A_{\overline{C_{0,0}}} \stackrel{?}{=} g^{s_{v^*}} \cdot \overline{D}^{s_{\mathsf{sk}}} \cdot \overline{C_{0,0}}^{-c}$

118: $A_{\overline{C_{Ln}}} \stackrel{?}{=} g^{s_{v'}} \cdot \overline{C_{Rn}}^{s_{\mathsf{sk}}} \cdot \overline{C_{Ln}}^{-c}$

119: $A_{C'} \stackrel{?}{=} h^{s_{\nu^*}} \cdot D'^{s_{\mathsf{sk}}} \cdot C'^{-c}$

120: $A_{C'_{Ln}} \stackrel{?}{=} h^{s_{\nu'}} \cdot C'^{s_{\mathsf{sk}}}_{Rn} \cdot C'^{-c}_{Ln}$

121: **end** $\mathcal{V}$

122: $\mathbf{h}' = \left(h_1, h_2^{(y^{-1})}, h_3^{(y^{-2})}, \ldots, h_{2 \cdot n}^{(y^{-2 \cdot n + 1})}\right)$ ▷ Complete inner product argument

123: $Z = A \cdot S^x \cdot \mathbf{g}^{-z} \cdot \mathbf{h}'^{z \cdot \mathbf{y}^{2 \cdot n}} \cdot \mathbf{h}'^{z^2 \cdot (\mathbf{2}^n \| \mathbf{0}^n) + z^3 \cdot (\mathbf{0}^n \| \mathbf{2}^n)}$

124: $\mathcal{P}$ and $\mathcal{V}$ engage in Protocol 1 of [BBB$^+$] on inputs $(\mathbf{g}, \mathbf{h}', Zh^{-\mu}, \hat{t}; \mathbf{l}, \mathbf{r})$

6 Performance

We describe our implementation of Anonymous Zether. Verification takes place in Solidity contracts. Proving takes place in a JavaScript library, which is in turn invoked by our front-end (also written in JavaScript).

We report performance measurements below. We note that *gas used* includes not just verification itself, but also the relevant account maintenance associated with the Zether Smart Contract; our gas measurements *do incorporate EIP-1108*. The *verification time* we report reflects only the time taken by the local EVM in evaluating a read-only call to the verification contract. Proving time is self-explanatory. Each number next to *Transfer* indicates the size of the anonymity set used (including the actual sender and recipient). Our "Burn" transaction is actually a *partial burn*, in contrast to that of [BAZB]; in other words, we allow a user to withdraw only part of her balance (using a single range proof).

	Prov. Time (ms)	Verif. Time (ms)	Prf. Size (bytes)	Gas Used
Burn	907	83	1,312	2,393,134
Transfer (2)	1,950	118	2,720	4,989,138
Transfer (4)	2,011	139	3,104	6,011,011
Transfer (8)	2,240	170	3,872	8,286,426
Transfer (16)	2,954	241	5,408	13,396,675
Transfer (32)	4,575	402	8,480	24,847,667
Transfer (64)	8,352	742	14,624	50,544,849
Transfer (N)	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)$	$\mathcal{O}(N \log N)$

References

[BAZB] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. Unpublished manuscript.

[BBB+] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. Full version.

[BCC+15] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, volume 9326 of *Lecture Notes in Computer Science*, pages 243–265. Springer International Publishing, 2015.

[FMMO] Prastudy Fauzi, Sarah Meiklejohn, Rebekah Mercer, and Claudio Orlandi. Quisquis: A new design for anonymous cryptocurrencies. Unpublished manuscript.

[GK15] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280. Springer Berlin Heidelberg, 2015.

[Nus82] H. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms*. Springer-Verlag, 1982.

[Pol71] J. M. Pollard. The fast Fourier transform in a finite field. *Mathematics of Computation*, 25(114):365–374, 1971.