One Scalar to Fold Them All, and in the Commitment Bind Them:
Reduced Logarithmic ElGamal Shuffle Proofs

Freeman Slaughter

University of South Florida, Tampa FL, USA
fslaughter@usf.edu

Abstract. Xue, Lu, and Au design a logarithmic-size verifiable shuffle for rerandomized ElGamal ciphertexts by combining a KZG permutation argument with a tagged inner-product argument, with cost $(2\log N+11)\mathbb{G}_1+8\mathbb{F}$. We reduce the communication and computation of their construction by observing that the rerandomization vector $\boldsymbol{\rho} \in \mathbb{F}^N$, despite being utilized as a full witness vector, only enters the consistency equations through the scalar $\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle$. We bind this scalar into an existing KZG commitment and replace XLA's two-vector consistency argument with a one-sided folding proof, while preserving the original relation, polynomial degrees, and powers-of-τ setup. We show that the the naive scalarization is unsound. Applied correctly, the committed scalar indeed permits extraction of the full rerandomization vector.
We present two variants: Variant A has proof size $(2\log_2 N+8)\mathbb{G}_1+5\mathbb{F}$, while Variant B adds one group element but removes one interactive round. Under XLA's grouped base-scalar-pair accounting, the prover's exponentiation count decreases from approximately $18N$ to $13N$ (or $14N$, for Variant B), and the verifier's drops from approximately $7N$ to $6N$. Each proof is a constant 240 B smaller than XLA's; in a 4-server mix-net election protocol with $N = 2^{10}$ ballots, our Variant A mixing proof costs 5.88 KiB in total, about 14% smaller than XLA.

1 Introduction

Mix-nets, which were introduced by Chaum [10], anonymize a batch of encrypted messages by routing them through servers in sequence, with each of them permuting and rerandomizing the ciphertexts. So long as at least one server is honest, the correspondence between the input and output ciphertexts remains hidden. This construction is highly applicable to the field of electronic voting wherein encrypted ballots are mixed before threshold decryption, with everything being posted to a public bulletin board, and has found use on the blockchain with coin-mixing services like MixEth [29]. This framework has a potential vulnerability where a malicious server – instead of simply mixing the

This research was supported by the National Science Foundation (Grant Number DMS-2342254).

ballots – could duplicate or replace ciphertexts undetected. The standard patch to this is a *verifiable shuffle*: a zero-knowledge proof that the output ciphertexts are nothing more than a rerandomized permutation of the input ciphertexts. Seeing as each server involved in mixing posts their own proof, which remain on the bulletin board or blockchain permanently, it is critical to realize small proof sizes, since these accumulate.

The first generation of linear-communication shuffle protocols was of two types: Furukawa and Sako [16] had the prover commit to a permutation matrix, then prove it was well formed and consistent with the shuffled ciphertexts. This line was simplified by Terelius and Wikström [30] and underlies Wikström's Verificatum [31], a mix-net software that sees use in government elections. On the other hand, Neff [27] observed that a polynomial is invariant under the action of permutation of its roots, a notion that was later refined by Groth [18]. Groth and Ishai [20] first achieved sublinear communication by arranging the N ciphertexts into an $m \times n$ matrix at cost $O(m^2 + n)$, then Bayer and Groth [4] reduced the cost to $O(\sqrt{N})$ using a quasilinear prover – this was the state of the art for about a decade. Bayer-Groth also popularized the aggregating technique leveraged in this paper: sampling a vector with entries x^i formed by powers of a challenge, then using these positional weights to reduce vector statements into inner products.

Logarithmic proofs required an different methodology. Bootle et al. [6] presented powerful recursive inner-product arguments with $O(\log N)$ communication in the discrete-log setting, which was later refined by Bulletproofs [8], and subsequent improvements [11,13]. Hoffmann, Klooß, and Rupp [22] (HKR) combined this machinery with the Bayer-Groth framework to obtain the first logarithmic-size shuffle for rerandomized ElGamal ciphertexts at cost $(6\log_2(N)+8)\mathbb{G}_1+4\mathbb{F}$. Similar relations have enjoyed their own logarithmic arguments – CurdleProofs [14] shuffles public keys and commitments for secret leader election for instance – but general-purpose SNARKs such as Groth16 [19] or PLONK [17] appear to be a poor fit for this relation: expressing a secret permutation requires a routing network to use $\Omega(N \log N)$ gates, and the ElGamal group operations involve non-native field arithmetic. By working with the group structure directly, one can avoid this costly overhead.

Xue, Lu, and Au [32] (XLA) currently hold the smallest proof for the ElGamal shuffle relation. They replace the Bayer-Groth permutation layer with a constant-size KZG grand-product argument [25,17], then aggregate the ciphertext consistency check into an inner-product argument, in which the same commitment is shared between the two halves. The XLA proof is $(2\log_2(N) + 11)\mathbb{G}_1 + 8\mathbb{F}$ – which accounts for 2.64 KiB at $N = 2^{20}$, one third of HKR's logarithmic sizes – with prover cost about $18N$ exponentiations and verifier cost about $7N$ plus three pairings, over a universal updatable powers-of-τ SRS [7]. One benefit compared to HKR is the structured setup: the ceremony is performed once, then may be reused across applications.

This paper compresses XLA further by excising an unneeded witness. Their consistency argument considers the full rerandomization vector $\boldsymbol{\rho} \in \mathbb{F}^N$, yet each

verification equation only interacts with it through the scalar $\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle$. One should not think of this scalarization as rote: a prover who is allowed to select ρ^* after the ciphertext-merging challenge may present false statements that pass verification, so ρ^* must be committed to early. We bind it in a blinder slot of the commitment that is already shared between both halves, which causes the second witness vector, its commitment, its masking, and one whole side of the folding argument to vanish. Our contributions are as follows:

1. We show that the naive scalarization is unsound: for this protocol, there exists an adversarial choice $\rho^*(w)$ which permits non-shuffles to satisfy the consistency check with probability $1 - 1/q$ (Section 3). This necessitates binding ρ^* before w, to prevent this weak Fiat-Shamir type vulnerability [24,5,12,26].
2. We present two compressed consistency arguments for the relation R_{vs}, which is unchanged from XLA. Variant A has proof sizes of $(2\log_2 N + 8)\mathbb{G}_1 + 5\mathbb{F}$, a constant uniform 240 B below XLA, while Variant B costs one additional group element to merge two challenge rounds together. Under XLA's accounting, grouped prover cost goes from $\approx 18N$ to $\approx 13N$ ($\approx 14N$ for Variant B) and verifier cost from $\approx 7N$ to $\approx 6N$, with all committed degrees and the SRS remaining unaltered (Sections 4 and 6).
3. We prove knowledge soundness for algebraically generated statements in the AGM and ROM: the positional challenge x lets the extractor reconstruct the full componentwise vector $\boldsymbol{\rho}$ from the committed scalar, with a rank argument showing that the remaining blinders preserve zero knowledge (Section 5).
4. We reconsider XLA's mix-net election accounting using our protocol. The smaller proofs cause compounding savings per server and per round, and a four-server mix shuffling $N = 2^{10}$ ballots achieves proof sizes of 5.88 KiB for Variant A, a 14% reduction compared to XLA (Section 7).

1.1 Technical Overview

The construction of Xue, Lu, and Au has two main parts:

1. A KZG-based permutation argument, in the grand-product style of PLONK [17], which proves that a committed vector $\mathbf{a}$ satisfies $a_i = x^{\pi(i)}$ for a permutation π and verifier-supplied challenge x, and
2. A consistency argument demonstrating that the same vector $\mathbf{a}$ links the input and output ElGamal ciphertexts.

Both portions share the commitment C_a, which is interpreted as a polynomial commitment in the first part and as a vector commitment in the second.

The consistency equations contain more linear structure than their witness suggests. Once $a_i = x^{\pi(i)}$ is fixed, the rerandomization vector appears in the arguments only through

$$\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle = \sum_{i=1}^{N} x^{\pi(i)} \rho_i.$$

This is the same scalar compression used in the Bayer-Groth shuffle [4]. Since C_a already carries blinding coefficients that multiply the vanishing polynomial $Z_H(X)$, we bind ρ^* into one of them, setting

$$f_a(X) = \sum_i a_i L_i(X) + \big(r_{a,0} + \rho^* X\big) Z_H(X).$$

That is, the second random blinder of XLA is replaced by the witness-dependent scalar ρ^*. This fixes ρ^* before the aggregation challenge w without adding a commitment, a round, or an additional degree, thus every committed polynomial retains its degree from XLA, so the same SRS can be used in our modified protocol. Note that adding an extra $X^2 Z_H(X)$ coefficient would not work, as it would raise $\deg(f_a)$ to $N+2$ and hence the permutation quotient q to degree $N+4$, which is one larger than the degree-$(N+3)$ SRS. See Remark 1. Because one of the two blinder slots of f_a is now dedicated to ρ^*, hiding must be argued differently. A rank computation in Lemma 9 shows that the three remaining random blinders of f_p and f_a still jointly hide the two commitments and the opened linear combination.

The order of the rounds here is important. Section 3 exhibits a false statement for which a prover can solve the merged consistency equation and falsely pass the verifier's checks, by adversarially selecting ρ^* as a rational function of w. Binding ρ^* to C_a, which is sent before w, excludes this attack. After this change, the consistency protocol has only a single secret vector. The second witness basis and the quadratic masking terms vanish, leaving a one-sided folding argument.

The main point requiring care is knowledge soundness. The consistency extractor returns $\mathbf{a}$ and ρ^*, but the shuffle relation R_{vs} requires a componentwise witness $\boldsymbol{\rho}$. Section 5.5 shows how to recover it in the AGM. The proof keeps externally supplied ciphertext components as independent formal coordinates and utilizes the same algebraic statement-generation model used by XLA. The random challenge x forces each first-component ratio to be a g-exponent, then the second aggregate equation proves that the same scalar works with Y.

Overall:

- Variant A keeps w and u in separate verifier rounds and uses a nested fork for extraction.
- Variant B samples (w, u) together and extracts from four accepting transcripts whose vectors $(1, w, u, wu)$ are linearly independent.
- The ρ^* coefficient remains hidden because the three remaining polynomial blinders preserve the inherited outer trapdoor simulator, and the consistency responses are rerandomized by fresh masks.
- Fiat-Shamir is conditional on a multi-round theorem [2,3].

2 Background and Notation

2.1 Groups, Vectors, and Commitments

Let $\mathbb{G}_1$ be a cyclic group of prime order q, written multiplicatively, and let $\mathbb{F} = \mathbb{F}_q$. For a group vector $\mathbf{A} = (A_1, \ldots, A_N)$ and scalar vector $\mathbf{c} = (c_1, \ldots, c_N)$,

write

$$\mathbf{A}^{\mathbf{c}} = \prod_{i=1}^{N} A_i^{c_i}.$$

Products and exponentiation of group vectors are componentwise. In particular,

$$\mathbf{A} \circ \mathbf{B}^w = (A_i B_i^w)_{i=1}^N.$$

Let $H = \{\omega^i \mid i \in [N]\} \subseteq \mathbb{F}$ be a multiplicative subgroup of size $N = 2^m$, let $L_i(X)$ be its Lagrange basis, and put $Z_H(X) = X^N - 1$. The protocol reuses the powers-of-τ SRS [25,7], where the only portion we use is

$$\mathsf{ck} = \Big(\{g^{\tau^i}\}_{i=0}^{N+3},\ \mathbf{g} = (g_1, \ldots, g_N),\ u_1, u_2, u_3,\ h, h^\tau, h^{\tau^2}\Big),$$

where

$$g_i = g^{L_i(\tau)}, \quad u_j = g^{\tau^{j-1} Z_H(\tau)}.$$

The $\mathbb{G}_2$ elements h, h^τ, h^{τ^2} enter as the standard batched KZG check of the permutation argument, and u_3 is retained as a blinder for the grand-product commitment C_z. For any coefficients,

$$\prod_i g_i^{c_i} u_1^{d_0} u_2^{d_1} u_3^{d_2} = g^{f(\tau)}, \quad f(X) = \sum_i c_i L_i(X) + (d_0 + d_1 X + d_2 X^2) Z_H(X).$$

The scalarized protocol uses only a subset of the XLA parameters (the generator family $\mathbf{k}$, the base u_ρ, and the extra hash-to-curve challenge for consistency are never used) and maintains the same setup.

2.2 ElGamal Shuffle Relation

The public key is an arbitrary $Y \in \mathbb{G}_1$. For an honestly generated key, one may write $Y = g^\kappa$, but the protocol assumes the extractor does not know κ. Define

$$\mathsf{Enc}_Y(M, \rho) = (g^\rho, Y^\rho M).$$

For input ciphertexts $C_j = (E_j, F_j)$ and outputs $\widetilde{C}_i = (\widetilde{E}_i, \widetilde{F}_i)$, the shuffle relation is

$$R_{\mathrm{vs}} = \Big\{(\mathsf{ck}, Y, \mathbf{C}, \widetilde{\mathbf{C}}; \pi, \boldsymbol{\rho}) \;\Big|\; \pi \in \Sigma_N, \quad \boldsymbol{\rho} \in \mathbb{F}^N, \quad \widetilde{C}_i = C_{\pi(i)} \mathsf{Enc}_Y(1, \rho_i) \quad \forall i \in [N]\Big\}.$$

After the outer verifier samples the positional challenge x from its challenge set $\mathcal{X} \subseteq \mathbb{F}^*$ (by default $\mathcal{X} = \mathbb{F}^*$), the honest prover sets

$$a_i = x^{\pi(i)}, \quad V_E = \prod_{j=1}^{N} E_j^{x^j}, \quad V_F = \prod_{j=1}^{N} F_j^{x^j}.$$

The shuffle equations imply

$$V_E = \widetilde{\mathbf{E}}^{\mathbf{a}} g^{-\rho^*}, \quad V_F = \widetilde{\mathbf{F}}^{\mathbf{a}} Y^{-\rho^*}, \quad \rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle . \tag{1}$$

For any $w \in \mathbb{F}$, define

$$\mathbf{h}_w = \widetilde{\mathbf{E}} \circ \widetilde{\mathbf{F}}^w, \quad u_w = gY^w, \quad V_w = V_E V_F^w. \tag{2}$$

Then the two equations in (1) merge into

$$V_w = \mathbf{h}_w^{\mathbf{a}} u_w^{-\rho^*}. \tag{3}$$

3 Why the Naive Scalarization Is Broken

The natural, but incorrect, modification would be to drop the vector rerandomizer entirely and let the prover choose a scalar ρ^* only after seeing w. Equation (3) then checks one merged relation, but it does not force both ElGamal components to utilize the *same* rerandomizer.

Proposition 1. *Suppose the prover may choose the scalar used in* (3) *after receiving uniform* $w \xleftarrow{\$} \mathbb{F}$. *Then an efficient algebraic prover can make the merged check accept a statement outside* R_{vs} *with probability* $1 - 1/q$.

Proof. Pick $\kappa_0 \xleftarrow{\$} \mathbb{F}^*$ and set $Y = g^{\kappa_0}$. Let every input ciphertext be $(1, 1)$, and publish outputs

$$\widetilde{C}_i = (g^{\rho_i}, g^{\rho'_i}),$$

where $\rho'_i \neq \kappa_0 \rho_i$ for at least one index. This is trivially not a shuffle; the first component of a valid rerandomization would force the rerandomizer to be ρ_i, while the second would then have to correspondingly be $Y^{\rho_i} = g^{\kappa_0 \rho_i}$.

The adversary can nonetheless pass the full verification. It performs the permutation argument honestly for the identity permutation, so $a_i = x^i$. Put

$$P = \langle \mathbf{a}, \boldsymbol{\rho} \rangle , \quad P' = \langle \mathbf{a}, \boldsymbol{\rho}' \rangle .$$

Since $V_E = V_F = 1$,

$$\mathbf{h}_w^{\mathbf{a}} = g^{P + wP'}, \quad u_w = gY^w = g^{1 + w\kappa_0}.$$

Thus the merged equation $1 = \mathbf{h}_w^{\mathbf{a}} u_w^{-\rho^*}$ is satisfied by the choice

$$\rho^*(w) = \frac{P + wP'}{1 + w\kappa_0}.$$

The denominator vanishes only at the single invalid $w = -\kappa_0^{-1}$, which occurs with probability $1/q$. For every other challenge, the adversary can compute this scalar and continue the protocol honestly, so it enjoys a success probability of $1 - 1/q$.

4 The Scalarized Protocol

4.1 Outer Protocol Changes

The permutation argument is retained. Anything not explicitly addressed is unaltered from the XLA protocol.

Modified outer steps for Π'_{vs}.

1. The prover sends the permutation commitment
$$C_p = \prod_i g_i^{p_i} u_1^{r_{p,0}} u_2^{r_{p,1}}.$$
The vector-rerandomization commitment C_ρ of XLA is omitted.
2. After receiving x, the prover sets $a_i = x^{p_i}$ and computes $\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle$. It samples $r_{a,0} \xleftarrow{\$} \mathbb{F}$, defines
$$f_a(X) = \sum_i a_i L_i(X) + \big(r_{a,0} + \rho^* X\big) Z_H(X),$$
and sends
$$C_a = \prod_i g_i^{a_i} u_1^{r_{a,0}} u_2^{\rho^*}.$$
The degree of f_a is at most $N+1$, as in XLA.
3. After the unchanged permutation checks, run Π'_{cons} with public input
$$(\mathsf{ck}, Y, V_E, V_F, C_a, \widetilde{\mathbf{E}}, \widetilde{\mathbf{F}})$$
and witness $(\mathbf{a}, \rho^*, r_{a,0})$.

Fig. 1. The modified outer steps. All checks are inherited from XLA.

Because Z_H vanishes on H, the new coefficient does not change the evaluations used by the permutation argument; that is, $f_a(\omega^i) = a_i$ still. Note that the commitment C_a is communicated before w. For an accepting prover, Lemma 4 establishes its intended opening and Lemma 1 makes ρ^* unique across the later forks, which is the precommitment required by Proposition 1. The effect of spending one blinder fo f_a on ρ^* is addressed by the rank argument of Lemma 9.

Remark 1. The blinder slot is the only available register; attempting to append a new coefficient $\rho^* X^2 Z_H(X)$ instead would raise $\deg(f_a)$ to $N+2$, hence $\deg d = \deg(f_p + y f_a)$ to $N+2$. This makes the grand-product quotient
$$q = \frac{(d-\alpha)z - (I-\alpha)z(\omega X) + \beta(z-1)L_1}{Z_H}$$
increase to degree $N+4$, which is one larger than the degree-$(N+3)$ SRS, so C_q could no longer be committed to. Reusing the $XZ_H(X)$ slot keeps every committed degree identical to XLA.

4.2 Consistency Relation and Variant A

The reduced consistency relation is

$$R'_{\text{cons}} = \left\{ (\mathsf{ck}, Y, V_E, V_F, C_a, \widetilde{\mathbf{E}}, \widetilde{\mathbf{F}}; \mathbf{a}, \rho^*, r_{a,0}) \;\middle|\; V_E = \widetilde{\mathbf{E}}^{\mathbf{a}} g^{-\rho^*},\ V_F = \widetilde{\mathbf{F}}^{\mathbf{a}} Y^{-\rho^*},\ C_a = \mathbf{g}^{\mathbf{a}} u_1^{r_{a,0}} u_2^{\rho^*} \right\}.$$

Protocol Π'_{cons}, Variant A. The prover knows $(\mathbf{a}, \rho^*, r_{a,0})$.

1. $P \to V$: sample $\mathbf{s}_a \xleftarrow{\$} \mathbb{F}^N$ and $s_\rho, r_0 \xleftarrow{\$} \mathbb{F}$, and send
$$M_a = \mathbf{g}^{\mathbf{s}_a} u_1^{r_0} u_2^{s_\rho}.$$
2. $V \to P$: sample $w \xleftarrow{\$} \mathbb{F}$. Both parties form $\mathbf{h}_w, u_w, V_w$ as in (2).
3. $P \to V$: send
$$M_{ct} = \mathbf{h}_w^{\mathbf{s}_a} u_w^{-s_\rho}.$$
4. $V \to P$: sample $u \xleftarrow{\$} \mathbb{F}^*$.
5. $P \to V$: send
$$\mu_0 = r_{a,0} + u r_0, \quad \hat{\rho} = \rho^* + u s_\rho.$$
6. Both parties compute
$$P_a = C_a M_a^u u_1^{-\mu_0} u_2^{-\hat{\rho}},$$
$$P_{ct} = V_w M_{ct}^u u_w^{\hat{\rho}}.$$
For an honest prover, with $\hat{\mathbf{a}} = \mathbf{a} + u\mathbf{s}_a$,
$$P_a = \mathbf{g}^{\hat{\mathbf{a}}}, \quad P_{ct} = \mathbf{h}_w^{\hat{\mathbf{a}}}.$$
7. $V \to P$: sample a fresh seed $\sigma \xleftarrow{\$} \{0,1\}^\lambda$. Append it to the transcript and derive $\mathbf{b} = \mathsf{Gen}(\mathsf{tr}, N)$.
8. $P \to V$: send $Q = \mathbf{b}^{\hat{\mathbf{a}}}$.
9. $V \to P$: sample $r'_1, r'_2 \xleftarrow{\$} \mathbb{F}^*$ and set
$$P_{\text{IPA}} = Q P_a^{r'_1} P_{ct}^{r'_2}, \quad \mathbf{G}_{\text{IPA}} = \mathbf{b} \circ \mathbf{g}^{r'_1} \circ \mathbf{h}_w^{r'_2}.$$
10. Run the one-sided argument Π'_{IPA} proving $P_{\text{IPA}} = \mathbf{G}_{\text{IPA}}^{\hat{\mathbf{a}}}$.

Fig. 2. The scalarized consistency protocol, Variant A.

The final relation $P_{\text{IPA}} = \mathbf{G}_{\text{IPA}}^{\hat{\mathbf{a}}}$ is linear in the single secret vector, so the protocol has no bilinear cross terms, second witness basis, or quadratic masking (T_1, T_2 and the extra hash-to-curve base s) like XLA. Structurally, the entire vector $\boldsymbol{\rho}$ branch is replaced by the single scalar response $\hat{\rho}$.

4.3 The One-Sided Asymmetric Folding Argument

The final relation has the form

$$R'_{\mathrm{IPA}} = \{(P, \mathbf{G}; \mathbf{a}) \mid P = \mathbf{G}^{\mathbf{a}}\}.$$

The following protocol is the asymmetric folding protocol of XLA in the style of [6,8,22], specialized to a single vector, while the compression technique is that of [1].

Protocol Π'_{IPA} for $N = 2^m$.

1. Initialize $\mathbf{G}_0 = \mathbf{G}$, $\mathbf{a}_0 = \mathbf{a}$, and $P_0 = P$.
2. For $j = 1, \ldots, m$, split $\mathbf{a}_{j-1} = (\mathbf{a}_L \parallel \mathbf{a}_R)$ and $\mathbf{G}_{j-1} = (\mathbf{G}_L \parallel \mathbf{G}_R)$ into equal halves.
   (a) $P \to V$: send $L_j = \mathbf{G}_R^{\mathbf{a}_L}$ and $R_j = \mathbf{G}_L^{\mathbf{a}_R}$.
   (b) $V \to P$: sample $\xi_j \xleftarrow{\$} \mathbb{F}^*$.
   (c) Fold
   $$\mathbf{G}_j = \mathbf{G}_L \circ \mathbf{G}_R^{\xi_j}, \quad \mathbf{a}_j = \xi_j \mathbf{a}_L + \mathbf{a}_R,$$
   and
   $$P_j = L_j^{\xi_j^2} P_{j-1}^{\xi_j} R_j.$$
3. $P \to V$: send the final scalar $\hat{a} = \mathbf{a}_m$.
4. Write the binary expansion of $i-1$ from most significant bit to least significant bit as $(\mathrm{bit}_1(i-1), \ldots, \mathrm{bit}_m(i-1))$, and set
   $$s_i = \prod_{j=1}^{m} \xi_j^{\mathrm{bit}_j(i-1)}, \quad \eta_j = \prod_{k=j}^{m} \xi_k, \quad \eta_{m+1} = 1.$$
   Accept iff
   $$\mathbf{G}^{\mathbf{s}\hat{a}} = P^{\eta_1} \prod_{j=1}^{m} \left(L_j^{\xi_j^2 \eta_{j+1}} R_j^{\eta_{j+1}} \right).$$

Fig. 3. One-sided asymmetric folding; cost is $2\log_2(N)\mathbb{G}_1 + 1\mathbb{F}$.

We now prove the security of the one-sided folding argument.

Theorem 1 (One-sided folding security). *Condition on the representation-soundness event of Section 5.1, and suppose the public basis $\mathbf{G}$ is computationally admissible: no efficient algebraic prover produces a nonzero $\mathbf{c}$ with $\mathbf{G}^{\mathbf{c}} = 1$. Then Π'_{IPA} is perfectly complete and has a witness-extended emulator for R'_{IPA}. Its generalized special-sound access structure is a complete ternary challenge tree of depth $m = \log_2(N)$, and the emulator extracts the canonical algebraic opening of P over $\mathbf{G}$ in expected time polynomial in N, the security parameter, and the inverse prover success probability.*

Proof. Completeness: The folding preserves $P_j = \mathbf{G}_j^{\mathbf{a}_j}$. Indeed, one round gives

$$\mathbf{G}_j^{\mathbf{a}_j} = (\mathbf{G}_L \circ \mathbf{G}_R^{\xi_j})^{\xi_j \mathbf{a}_L + \mathbf{a}_R} = L_j^{\xi_j^2} P_{j-1}^{\xi_j} R_j.$$

After m rounds, both the folded basis and the folded commitment expand to the verifier's final equation.

Extraction from a ternary tree: Fix one round's messages (L, R) and consider three accepting transcripts with distinct challenges ξ. By induction, each gives an opening of

$$P(\xi) = L^{\xi^2} P^{\xi} R$$

over the folded basis $\mathbf{G}_L \circ \mathbf{G}_R^{\xi}$. In any formal coordinate outside those spanned by the current basis, the left-hand side is a polynomial $\ell\xi^2 + p\xi + r$ of degree at most two. Vanishing at three distinct challenges forces all three coefficients to vanish. Hence L, P, and R lie in the span of the current basis, and solving the Vandermonde system in $(\xi^2, \xi, 1)$ recovers their openings. Two different recovered openings of the same element would give a nontrivial relation on $\mathbf{G}$, contradicting admissibility.

Applying this argument from the bottom of the recursion to the top gives a complete ternary extraction tree with $3^m = N^{\log_2 3}$ leaves. The usual recursive argument for compressed public-coin proofs [1,32] provides the required distinct children at each node whenever the prover's continuation probability exceeds the corresponding generalized-special-soundness knowledge error. Since the depth and number of leaves are polynomial in N, the total expected running time is polynomial as well.

4.4 Variant B: Merging the (w, u) Challenges

Variant A sends M_{ct} after w, so w and u occupy separate rounds. Variant B instead sends, along with M_a,

$$M_E = \widetilde{\mathbf{E}}^{\mathbf{s}_a} g^{-s_\rho}, \quad M_F = \widetilde{\mathbf{F}}^{\mathbf{s}_a} Y^{-s_\rho}.$$

The verifier later computes $M_{ct} = M_E M_F^w$. For honest masks, this equals $\mathbf{h}_w^{\mathbf{s}_a} u_w^{-s_\rho}$, and the remaining steps are unchanged. All prover messages preceding the responses are now independent of both challenges, so the verifier may send the pair (w, u) in one go. The consistency phase saves one round of communication and one public-coin challenge block, while communication increases by one group element because (M_E, M_F) replaces M_{ct}.

After the challenges are merged, the extractor cannot rewind u while holding w fixed. Instead it uses four accepting transcripts such that $(1, w_j, u^{(j)}, w_j u^{(j)})$ are linearly independent. Lemma 6 proves this in the security section.

Challenge order: The omitted outer messages and challenges are those of XLA. The displayed Figure 4 flowchart outlines the challenge and message order for these variants.

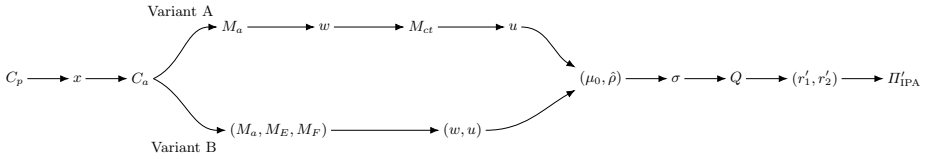

Fig. 4. Challenge and message order for the two protocol variants.

5 Security Analysis

The security proof is easiest to understand as a sequence of five extraction steps:

1. The group element C_a is fixed before the consistency challenges are sampled; the u-fork later establishes its unique intended opening $(\mathbf{a}, r_{a,0}, \rho^*)$.
2. The fresh basis $\mathbf{b}$ fixes the response vector used by the one-sided IPA.
3. The challenges (r'_1, r'_2) separate the three commitments Q, P_a, and P_{ct}, forcing them to open to that same vector.
4. The challenge u recovers the opening of C_a, and the challenge w (or in Variant B, the joint challenge pairs) separates the two ElGamal component equations.
5. The earlier positional challenge x turns those two aggregate equations into a componentwise rerandomization vector $\boldsymbol{\rho}$.

Note that the last step uses the algebraic representation of the statement, so we first need to state precisely what the extractor may read.

5.1 Algebraic Representations

In the AGM [15], every adversarial group element carries a representation over the elements seen so far. We expand SRS-derived elements such as g_i and u_j into the powers-of-τ block, but keep statement elements and fresh random-oracle outputs as independent formal coordinates. Write $\mathrm{rep}(A) \in \mathcal{V}$ for this representation, let $\mathbf{e}_g = \mathrm{rep}(g)$, and fix a splitting $\mathcal{V} = \langle \mathbf{e}_g \rangle \oplus \mathcal{V}_\perp$ with projection $\mathrm{pr}_\perp$. This can be thought of as accounting in the extractor's coordinate space. The fresh $\mathbf{b}$ block fixes the IPA response in Lemmas 2 and 3, while $\mathrm{pr}_\perp$ eliminates ρ^* in Lemma 8. Only SRS-derived elements are known to lie in the powers-of-τ span.

Definition 1 (Representation soundness). *Let* Rep *be the event that every accepting group equality used by the verifier or extractor also holds in the formal coordinates after the explicitly known SRS identities are expanded. Let* $\epsilon_{\mathsf{Rep}} = 1 - \Pr[\mathsf{Rep}]$. *A violation of* Rep *yields either a nonzero polynomial of degree at most* $N + 3$ *with known root* τ*, contradicting the* $(N + 3)$*-DLOG assumptions of XLA (Lemma 3 of [32], following [17]), or a nontrivial relation*

among independent group points, contradicting the discrete-logarithm assumption. Hence $\epsilon_{\mathsf{Rep}} = \mathsf{negl}(\lambda)$ *under the assumptions of XLA. Relations involving fresh random-oracle events are accounted for separately under* $\epsilon_{\mathrm{fresh}}$.

5.2 The Shared Commitment

Lemma 1 (Uniqueness of the intended opening of C_a). *Let*

$$f_a(X) = \sum_{i=1}^{N} a_i L_i(X) + \big(r_{a,0} + \rho^* X\big) Z_H(X).$$

Then

$$C_a = g^{f_a(\tau)} = \mathbf{g}^{\mathbf{a}} u_1^{r_{a,0}} u_2^{\rho^*}, \quad f_a(\omega^i) = a_i.$$

Conditioned on Rep *and KZG representation binding, if C_a admits an opening over $(\mathbf{g}, u_1, u_2)$, the tuple $(\mathbf{a}, r_{a,0}, \rho^*)$ is unique for an efficient algebraic prover.*

Proof. The evaluation claim follows from $Z_H(\omega^i) = 0$. Moreover,

$$\{L_i(X) \mid i \in [N]\} \cup \{Z_H(X), X Z_H(X)\}$$

is a basis of the polynomials of degree at most $N+1$. Evaluation on H fixes all of the Lagrange coefficients, and the remaining Z_H terms fix its two coefficients. Two different openings over $(\mathbf{g}, u_1, u_2)$ would therefore give a nonzero polynomial vanishing at τ, contradicting KZG representation binding. Existence of such an opening for an accepting prover is established by the u-fork in Lemma 4, and the uniqueness statement makes that opening shared with later forks.

Remark 2. The group element C_a is fixed before w. Once an accepting extraction fork establishes its intended opening, Lemma 1 binds three objects to that commitment: the permutation evaluations $\mathbf{a}$, the vector used by the consistency argument, and the aggregate rerandomizer ρ^*. This is the precommitment required by Proposition 1.

5.3 The Fresh Basis

The seed σ is sampled only after the statement-dependent elements and the values P_a, P_{ct} are fixed. Hence the derived basis $\mathbf{b} = (b_1, \ldots, b_N)$ occupies a fresh formal-coordinate block that no earlier element can use.

Lemma 2 (Admissibility of the combined basis). *Conditioned on random-oracle freshness and* Rep*, the basis*

$$\mathbf{G}_{\mathrm{IPA}} = \mathbf{b} \circ \mathbf{g}^{r_1'} \circ \mathbf{h}_w^{r_2'}$$

is computationally admissible for every $r_1', r_2' \in \mathbb{F}^*$.

Proof. Suppose $\mathbf{G}_{\mathrm{IPA}}^{\mathbf{c}} = 1$. Project the formal equality onto the fresh $\mathbf{b}$ block. The ith basis element contains b_i with coefficient one, while $\mathbf{g}$ and $\mathbf{h}_w$ have no coordinates in this block. The projection is therefore no more than $\mathbf{c}$, so $\mathbf{c} = 0$.

The fresh basis first anchors one response vector, while the random linear combination forces each of the three component commitments to open to that same vector, so we require a lemma to ensure linking.

Lemma 3. *Fix a transcript through Q, and fix (w, u). Suppose the combined IPA accepts on a pair $(r'_1, r'_2) \in (\mathbb{F}^*)^2$ and the folding challenges. Except for the IPA-emulation, freshness, and representation failures, and except on at most a $1/(q-1)$ fraction of the (r'_1, r'_2) pairs, the emulator extracts a vector $\boldsymbol{\alpha} \in \mathbb{F}^N$ such that*

$$Q = \mathbf{b}^{\boldsymbol{\alpha}}, \quad P_a = \mathbf{g}^{\boldsymbol{\alpha}}, \quad P_{ct} = \mathbf{h}_w^{\boldsymbol{\alpha}}. \tag{4}$$

Proof. Theorem 1 extracts $\boldsymbol{\alpha}$ from

$$Q P_a^{r'_1} P_{ct}^{r'_2} = \left(\mathbf{b} \circ \mathbf{g}^{r'_1} \circ \mathbf{h}_w^{r'_2}\right)^{\boldsymbol{\alpha}}.$$

Projecting onto the fresh $\mathbf{b}$ block (P_a, P_{ct}, and $\mathbf{h}_w$ are fixed before σ, so they have zero coordinates there) shows that $\boldsymbol{\alpha}$ equals the $\mathbf{b}$-coordinates of Q, hence is fixed before (r'_1, r'_2). Define the residuals

$$A = Q\mathbf{b}^{-\boldsymbol{\alpha}}, \quad B = P_a \mathbf{g}^{-\boldsymbol{\alpha}}, \quad C = P_{ct}\mathbf{h}_w^{-\boldsymbol{\alpha}}.$$

They clearly satisfy $AB^{r'_1}C^{r'_2} = 1$. If one residual is nontrivial, some coordinate gives a nonzero affine polynomial in (r'_1, r'_2), which vanishes on at most a $1/(q-1)$ fraction of $(\mathbb{F}^*)^2$. Outside this undesirable event, all three residuals are the identity, proving (4).

5.4 Consistency Witness Extraction

Variant A: The challenge order supports a nested extraction: first two u values at a fixed w, and then two distinct w values.

Lemma 4 (Two u branches at fixed w). *Fix a transcript through M_{ct}, including w, and obtain two accepting continuations with distinct challenges $u^{(0)}, u^{(1)} \in \mathbb{F}^*$. After applying Lemma 3 to both branches, the extractor recovers an opening $(\mathbf{a}, \rho^*, r_{a,0})$ of C_a and obtains*

$$V_w = \mathbf{h}_w^{\mathbf{a}} u_w^{-\rho^*}. \tag{5}$$

Proof. For branch $j \in \{0, 1\}$, separation gives

$$C_a M_a^{u^{(j)}} = \mathbf{g}^{\boldsymbol{\alpha}^{(j)}} u_1^{\mu_0^{(j)}} u_2^{\hat{\rho}^{(j)}}, \quad V_w M_{ct}^{u^{(j)}} u_w^{\hat{\rho}^{(j)}} = \mathbf{h}_w^{\boldsymbol{\alpha}^{(j)}}.$$

Let $\Delta u = u^{(1)} - u^{(0)}$. Divided differences recover the masks,

$$\mathbf{s}_a = \frac{\boldsymbol{\alpha}^{(1)} - \boldsymbol{\alpha}^{(0)}}{\Delta u}, \quad s_\rho = \frac{\hat{\rho}^{(1)} - \hat{\rho}^{(0)}}{\Delta u}, \quad r_0 = \frac{\mu_0^{(1)} - \mu_0^{(0)}}{\Delta u}.$$

Dividing the two first equations gives the opening $M_a = \mathbf{g}^{\mathbf{s}_a} u_1^{r_0} u_2^{s_\rho}$, then substituting back gives the opening of C_a. Dividing the two second equations gives $M_{ct} = \mathbf{h}_w^{\mathbf{s}_a} u_w^{-s_\rho}$, and substituting back gives (5). Lemma 1 makes the extracted opening of C_a the same in every later fork.

Lemma 5 (Two w branches split the ElGamal equations). *Suppose Lemma 4 is applied at two distinct values $w^{(0)}, w^{(1)}$. Then*

$$V_E = \widetilde{\mathbf{E}}^{\mathbf{a}} g^{-\rho^*}, \quad V_F = \widetilde{\mathbf{F}}^{\mathbf{a}} Y^{-\rho^*}.$$

Proof. For either value of w, rewrite (5) as $AB^w = 1$, where

$$A = V_E \widetilde{\mathbf{E}}^{-\mathbf{a}} g^{\rho^*}, \quad B = V_F \widetilde{\mathbf{F}}^{-\mathbf{a}} Y^{\rho^*}.$$

The opening of C_a is canonical, so the same A and B occur in both branches. Dividing the two equations gives $B^{w^{(1)} - w^{(0)}} = 1$, hence $B = 1$ by distinctness of the challenges. Substituting back gives $A = 1$.

Remark 3 (Variant A fork). At a fixed w, scalarization reduces the local degree in u, so two accepting u transcripts suffice. The complete Variant A emulator still needs two values of w, two values of u below each selected w, and the recursive IPA extraction tree.

Variant B: Here w and u are sampled in the same verifier move, with $(w, u) \in \mathbb{F} \times \mathbb{F}^*$, so a nested u-fork with fixed w is no longer an option. The extractor instead uses four accepting joint challenge pairs.

Lemma 6 (Four-point extraction for Variant B). *Fix (M_a, M_E, M_F) and obtain four accepting transcripts with challenge pairs $(w_j, u^{(j)}) \in \mathbb{F} \times \mathbb{F}^*$, where $j = 0, 1, 2, 3$. Define*

$$\Phi = \begin{pmatrix} 1 & w_0 & u^{(0)} & w_0 u^{(0)} \\ 1 & w_1 & u^{(1)} & w_1 u^{(1)} \\ 1 & w_2 & u^{(2)} & w_2 u^{(2)} \\ 1 & w_3 & u^{(3)} & w_3 u^{(3)} \end{pmatrix}.$$

If $\det \Phi \neq 0$, then after applying Lemma 3 to each of the four branches, the extractor obtains the canonical intended opening of C_a and proves

$$V_E = \widetilde{\mathbf{E}}^{\mathbf{a}} g^{-\rho^*}, \quad V_F = \widetilde{\mathbf{F}}^{\mathbf{a}} Y^{-\rho^*}.$$

Proof. Invertibility of Φ implies that at least two u values differ. Those two branches recover the openings of C_a and M_a precisely as in Lemma 4, since the P_a equation does not involve w. Lemma 1 then forces every response branch to have the form

$$\boldsymbol{\alpha}^{(j)} = \mathbf{a} + u^{(j)} \mathbf{s}_a, \quad \hat{\rho}^{(j)} = \rho^* + u^{(j)} s_\rho, \quad \mu_0^{(j)} = r_{a,0} + u^{(j)} r_0.$$

Substitution into the ciphertext equation $P_{ct}^{(j)} = \mathbf{h}_{w_j}^{\boldsymbol{\alpha}^{(j)}}$ gives

$$AB^{w_j} C^{u^{(j)}} D^{w_j u^{(j)}} = 1 \tag{6}$$

for the branch-independent equations

$$A = V_E \widetilde{\mathbf{E}}^{-\mathbf{a}} g^{\rho^*}, \quad B = V_F \widetilde{\mathbf{F}}^{-\mathbf{a}} Y^{\rho^*}, \quad C = M_E \widetilde{\mathbf{E}}^{-\mathbf{s}_a} g^{s_\rho}, \quad D = M_F \widetilde{\mathbf{F}}^{-\mathbf{s}_a} Y^{s_\rho}.$$

Since Φ is nonsingular, one can take a linear combination of the rows to obtain the four standard basis vectors. Using these vectors in the four group equations gives $A = B = C = D = 1$. The identities $A = B = 1$ are the two required ElGamal equations.

Lemma 7 (Rank of accepted joint challenges). *Let*

$$\mathcal{D} = \mathbb{F} \times \mathbb{F}^*, \quad \phi(w, u) = (1, w, u, wu) \in \mathbb{F}^4.$$

If $S \subseteq \mathcal{D}$ and $\operatorname{span}\{\phi(w, u) \mid (w, u) \in S\}$ is a proper subspace of $\mathbb{F}^4$, then $|S| \leq 2(q - 1)$.

Fix a transcript before the joint challenge, let $p(w, u)$ be the acceptance probability of the remaining public-coin portion, and let

$$\varepsilon = \frac{1}{q(q-1)} \sum_{(w,u)\in\mathcal{D}} p(w, u).$$

If $\varepsilon > 8/q$, the standard rewinding procedure obtains four accepting continuations whose ϕ vectors have rank four in expected time polynomial in the security parameter, N, and $1/\varepsilon$. Thus failure of Lemma 6 contributes at most $8/q$ to the Variant B knowledge error, apart from the emulation errors already accounted for separately.

Proof. If the ϕ vectors of S lie in a proper subspace, there is a nonzero $(a, b, c, d) \in \mathbb{F}^4$ such that

$$f(w, u) = a + bw + cu + dwu$$

vanishes on S. For a fixed w, this is affine in u. For all but at most one value of w, it has at most one root in $\mathbb{F}^*$. At a possible exceptional value of w, it has at most $q - 1$ roots. Hence a nonzero f has at most $2(q - 1)$ roots in $\mathcal{D}$.

Let $H = \{(w, u) \mid p(w, u) \geq \varepsilon/2\}$. Averaging gives

$$\frac{|H|}{|\mathcal{D}|} \geq \frac{\varepsilon}{2 - \varepsilon} \geq \frac{\varepsilon}{2}.$$

At any intermediate rank below four, the challenge pairs whose ϕ vectors lie in the current span form a set of size at most $2(q - 1)$, hence density at most $2/q$. If $\varepsilon > 8/q$, the pairs that increase the rank have density greater than $\varepsilon/4$, and each has acceptance probability at least $\varepsilon/2$. Standard truncation therefore finds and emulates a rank-increasing accepting pair in expected polynomial time. Repeating this at most four times gives a rank-four quadruple. If $\varepsilon \leq 8/q$, the success probability is accounted for directly in the knowledge error.

Theorem 2 (Security of the consistency protocol). *Conditioned on* Rep*, KZG representation binding, and random-oracle freshness, Π'_{cons} is perfectly complete, perfectly special honest-verifier zero knowledge, and has a witness-extended emulator for R'_{cons}. Variant A uses two values of w, two values of u for each selected w, and, in each u branch, separation over (r'_1, r'_2) followed by the recursive IPA tree. Variant B uses Lemma 7 to obtain the structure of Lemma 6, followed by the same emulators.*

Proof. Completeness: For honest responses $\hat{\mathbf{a}} = \mathbf{a} + u\mathbf{s}_a$ and $\hat{\rho} = \rho^* + us_\rho$, direct substitution gives $P_a = \mathbf{g}^{\hat{\mathbf{a}}}$ and $P_{ct} = \mathbf{h}_w^{\hat{\mathbf{a}}}$. Hence the final IPA statement is valid.

Special honest-verifier zero knowledge: Fix the verifier's challenges, with $u \neq 0$. The simulator samples the responses $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{F}^N$ and $\hat{\rho}, \mu_0 \xleftarrow{\$} \mathbb{F}$ uniformly, then solves the verification equations in reverse. For both variants it sets

$$M_a = \left(\mathbf{g}^{\boldsymbol{\alpha}} C_a^{-1} u_1^{\mu_0} u_2^{\hat{\rho}}\right)^{1/u}.$$

For Variant A it additionally sets

$$M_{ct} = \left(\mathbf{h}_w^{\boldsymbol{\alpha}} V_w^{-1} u_w^{-\hat{\rho}}\right)^{1/u},$$

while for Variant B it instead sets

$$M_E = \left(\widetilde{\mathbf{E}}^{\boldsymbol{\alpha}} V_E^{-1} g^{-\hat{\rho}}\right)^{1/u}, \quad M_F = \left(\widetilde{\mathbf{F}}^{\boldsymbol{\alpha}} V_F^{-1} Y^{-\hat{\rho}}\right)^{1/u}.$$

For a true statement, these are the honest values determined by the sampled responses. The simulator then sets $Q = \mathbf{b}^{\boldsymbol{\alpha}}$ and runs the honest one-sided IPA on $\boldsymbol{\alpha}$. Since the honest masks are uniform and multiplication by $u \neq 0$ is a bijection on $\mathbb{F}$, the sampled responses are indistinguishable from an honest interaction.

Extraction: For Variant A, Lemmas 4 and 5 give a witness for R'_{cons}. For Variant B, Lemma 7 obtains a nondegenerate accepting quadruple, and Lemma 6 gives the same witness. The separation and recursive-IPA emulators are then applied for each selected branch, with their failure probabilities included in the corresponding witness-extended emulation term.

5.5 Recovering the Componentwise Rerandomizers

The consistency extractor has recovered only the scalar ρ^*, while the shuffle relation asks for the full vector $\boldsymbol{\rho}$. The positional challenge x supplies the missing information: a false componentwise relation can satisfy the aggregate identity for at most N choices of x in each ElGamal component.

Lemma 8 (Extraction of $\boldsymbol{\rho}$ from x). *Assume the input and output ciphertexts, the public key, and the permutation commitment are fixed before $x \xleftarrow{\$} \mathcal{X}$. Suppose the outer extractor gives a permutation π satisfying $a_i = x^{\pi(i)}$, and the consistency extractor gives*

$$V_E = \widetilde{\mathbf{E}}^{\mathbf{a}} g^{-\rho^*}, \quad V_F = \widetilde{\mathbf{F}}^{\mathbf{a}} Y^{-\rho^*}. \tag{7}$$

Conditioned on Rep*, except with probability at most* $2N/|\mathcal{X}|$ *over* x*, the AGM extractor computes* $\boldsymbol{\rho} \in \mathbb{F}^N$ *such that*

$$\widetilde{C}_i = C_{\pi(i)}\mathsf{Enc}_Y(1, \rho_i) \quad \textit{for every } i \in [N].$$

Proof. First components: Reindex the first equation in (7) using $a_i = x^{\pi(i)}$ and define

$$R_j = E_j \widetilde{E}^{-1}_{\pi^{-1}(j)}.$$

Then

$$\prod_{j=1}^{N} R_j^{x^j} = g^{-\rho^*}. \tag{8}$$

Write $\mathbf{r}_j = \mathrm{rep}(R_j)$. Every R_j is fixed before x, so the formal-coordinate version of (8) is

$$\sum_{j=1}^{N} x^j \mathbf{r}_j + \rho^* \mathbf{e}_g = 0. \tag{9}$$

Projecting onto $\mathcal{V}_\perp$ removes the last term. Although ρ^* itself depends on x, this projection eliminates it regardless. Unless the vector polynomial

$$\sum_{j=1}^{N} X^j \,\mathrm{pr}_\perp(\mathbf{r}_j)$$

is identically zero, some scalar coordinate is a nonzero polynomial of degree at most N, and therefore vanishes for at most N values of x. Outside this bad set, every coefficient is zero, so each ratio satisfies

$$R_j = g^{c_j}$$

for a scalar c_j that the extractor reads from the representation (the $\mathbf{e}_g$-coordinate of $\mathbf{r}_j$, fixed before x). The $\mathbf{e}_g$-coordinate of the accepted equality then gives

$$\rho^* = -\sum_{j=1}^{N} c_j x^j. \tag{10}$$

Second components: Define

$$S_j = F_j \widetilde{F}^{-1}_{\pi^{-1}(j)}, \quad \mathbf{s}_j = \mathrm{rep}(S_j), \quad \mathbf{y} = \mathrm{rep}(Y).$$

Using the second equation in (7) and (10), we obtain

$$\sum_{j=1}^{N} x^j (\mathbf{s}_j - c_j \mathbf{y}) = 0,$$

where the coefficients are again fixed before x. A second degree-N root argument therefore gives $S_j = Y^{c_j}$ for every j, except on another bad set of at most N challenges.

Finally set,

$$\rho_i = -c_{\pi(i)}. \tag{11}$$

The identities $R_{\pi(i)} = g^{c_{\pi(i)}}$ and $S_{\pi(i)} = Y^{c_{\pi(i)}}$ rearrange to

$$\widetilde{E}_i = E_{\pi(i)} g^{\rho_i}, \quad \widetilde{F}_i = F_{\pi(i)} Y^{\rho_i},$$

which is exactly the componentwise shuffle relation. The two root bounds sum to $2N/|\mathcal{X}|$.

5.6 Outer Zero Knowledge

The coefficient ρ^* of $XZ_H(X)$ in f_a is witness-dependent, so f_a retains only one uniformly at random blinder. It is not sufficient to argue that a single random coefficient hides one evaluation: the simulator must jointly account for the two commitments and the subsequent opened linear combination. To resolve this, we need a small rank calculation.

Lemma 9 (Joint hiding of the outer commitments). *Let*

$$f_p(X) = A_p(X) + (r_{p,0} + r_{p,1}X)Z_H(X),$$
$$f_a(X) = A_a(X) + r_{a,0}Z_H(X),$$

where A_a contains all fixed witness-dependent terms, including $\rho^* XZ_H(X)$. For $v \notin H$, let $y \in \mathbb{F}$, and set $d(v) = f_p(v) + y f_a(v)$. Conditioned on $\tau \notin H$, $v \notin H$, and $v \neq \tau$, the triple

$$\big(f_p(\tau), f_a(\tau), d(v)\big)$$

is uniform over $\mathbb{F}^3$ as $(r_{p,0}, r_{p,1}, r_{a,0}) \xleftarrow{\$} \mathbb{F}^3$, independently of (A_p, A_a). Consequently, the joint distribution of $(C_p, C_a, d(v))$ supplied to the inherited XLA outer trapdoor simulator is unchanged.

Proof. The witness-dependent terms contribute only a fixed additive offset. After factoring the nonzero values $Z_H(\tau)$ and $Z_H(v)$ out of the corresponding rows, the three random blinders map to the three observed values through

$$M = \begin{pmatrix} 1 & \tau & 0 \\ 0 & 0 & 1 \\ 1 & v & y \end{pmatrix}, \quad \det M = \tau - v.$$

This matrix is invertible whenever $v \neq \tau$, even when $y = 0$. Hence a uniform vector in $\mathbb{F}^3$ maps to a uniform vector in $\mathbb{F}^3$, and adding the fixed offset only translates the distribution. Conditioned on this, the commitments C_z, C_q, C_W, and the remaining opening data are generated by the unchanged XLA trapdoor simulator. The consistency responses $\mu_0 = r_{a,0} + u r_0$ and $\hat{\rho} = \rho^* + u s_\rho$ are one-time-padded by the fresh masks r_0, s_ρ and are independent of the triple.

5.7 Main Theorem

Let

- ϵ_{out} be the knowledge error of the unchanged outer permutation extractor,
- $\epsilon_{\text{emul},A}$ and $\epsilon_{\text{emul},B}$ represent the challenge-collision and recursive-IPA emulation failures of the consistency arguments, excluding the explicit algebraic bad events below,
- ϵ_{fresh} be the random-oracle freshness error,
- ϵ_{Rep} be the representation-soundness failure from Definition 1.

Each variant uses four consistency branches. Applying Lemma 3 to those branches contributes at most $4/(q-1)$, by a union bound. Define $\delta_A = 0$ and $\delta_B = \frac{8}{q}$, where the Variant B term comes from Lemma 7.

Theorem 3 (Security of the scalarized shuffle). *Assume that:*

1. *the unchanged outer permutation argument is perfectly complete, knowledge sound, and statistically trapdoor zero knowledge under its stated KZG assumptions,*
2. *statements and prover messages are generated algebraically before their corresponding verifier challenges,*
3. *the representation-soundness, $(N+3)$-DLOG, discrete-log, KZG binding, and random-oracle freshness assumptions used above hold,*
4. *$q^{-1} = \mathsf{negl}(\lambda)$, $N = \text{poly}(\lambda)$, and $N/|\mathcal{X}| = \mathsf{negl}(\lambda)$.*

Then Π'_{vs}, with either variant, is perfectly complete and is an argument of knowledge for the original shuffle relation R_{vs} in the AGM and ROM. For $\star \in \{A, B\}$, its knowledge error is at most

$$\epsilon_{\text{out}} + \epsilon_{\text{emul},\star} + \epsilon_{\text{fresh}} + \epsilon_{\mathsf{Rep}} + \frac{4}{q-1} + \delta_\star + \frac{2N}{|\mathcal{X}|}. \tag{12}$$

The consistency phase is perfectly special honest-verifier zero knowledge. Under the inherited XLA outer trapdoor simulator and the additional outer bad event $v = \tau$, the entire protocol also realizes the outer argument's statistical trapdoor zero-knowledge.

Proof. Completeness: For an honest shuffle, $a_i = x^{\pi(i)}$ and $\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle$. Reindexing the componentwise ciphertext equations gives the two aggregate identities (1). The unchanged permutation checks and Theorem 2 then give acceptance.

Knowledge soundness: The outer extractor first obtains a permutation π and the committed vector $\mathbf{a}$ with $a_i = x^{\pi(i)}$. The selected consistency extractor then obtains an intended opening $(\mathbf{a}', \rho^*, r_{a,0})$ of the same commitment C_a and the two aggregate ciphertext equations. Lemma 1 forces $\mathbf{a}' = \mathbf{a}$. Lemma 8 reconstructs $\boldsymbol{\rho}$ and proves the componentwise ElGamal relations, so the extractor outputs $(\pi, \boldsymbol{\rho}) \in R_{\text{vs}}$. Four applications of Lemma 3 give the term $4/(q-1)$, Lemma 7

gives δ_B, and the two degree-N root events in Lemma 8 give $2N/|\mathcal{X}|$. The remaining terms are the inherited outer, emulation, freshness, and representation failures.

Zero knowledge: The consistency phase is perfectly special honest-verifier simulatable by Theorem 2. Lemma 9 shows that the joint distribution of $(C_p, C_a, d(v))$ supplied to the outer trapdoor simulator is unchanged after one blinder is replaced by ρ^*. Conditioned on that, the remainder of the outer simulation is the XLA simulator. Composing the two simulators gives the claimed inherited zero-knowledge.

5.8 Fiat-Shamir Paradigm

Because the protocol has several public-coin challenge layers, its Fiat-Shamir analysis requires a theorem for multi-round protocols [2,3], and should not use the classical three-move forking lemma.

Corollary 1. *Suppose the transcript encoding, challenge samples and sets, and generalized special-sound structure of Π'_{vs} satisfy the hypotheses of Theorem 3, and suppose every group element is validated before it is hashed. Then hashing each public-coin verifier move yields a noninteractive argument of knowledge in the ROM, with the interactive extraction error of Theorem 3 and the reduction loss as claimed.*

Proof. Apply the multi-round Fiat-Shamir theorem using Theorem 3 as the interactive witness-extended emulator. Care must be taken to avoid weak Fiat-Shamir [5,12,26]: each challenge must be bound to the protocol identifier, the selected variant, the complete statement, and the entire prior transcript. The theorem's structural hypotheses must still be checked against this paradigm. Theorem 1 supplies the ternary structure of the folding rounds, while Lemma 7 provides the accepted-challenge structure of the combined (w, u) round.

Remark 4. Variant B removes one challenge and one back-and-forth round of communication. This simplifies the transcript, but does not imply a tighter knowledge reduction.

6 Efficiency

6.1 Proof Size

Tallying the prover messages in Figures 1-3, we see that the outer argument contributes five commitments C_p, C_a, C_z, C_q, C_W and the two evaluations $d_v, z_{\omega v}$. Variant A's consistency portion contributes M_a, M_{ct}, Q, the responses $\mu_0, \hat{\rho}$, and one IPA transcript of $2\log_2(N)$ group elements ,plus the scalar $\hat{a}$. Hence

$$\text{Variant A: } (2\log_2(N) + 8)\mathbb{G}_1 + 5\mathbb{F},$$
$$\text{Variant B: } (2\log_2(N) + 9)\mathbb{G}_1 + 5\mathbb{F},$$

against XLA's $(2\log_2(N)+11)\mathbb{G}_1 + 8\mathbb{F}$. With 48 B compressed $\mathbb{G}_1$ elements and 32 B scalars, these become

$$\begin{aligned} \text{XLA: } & 96\log_2(N) + 784 \text{ B}, \\ \text{Variant A: } & 96\log_2(N) + 544 \text{ B}, \\ \text{Variant B: } & 96\log_2(N) + 592 \text{ B}. \end{aligned}$$

Thus Variant A saves 240 B and Variant B saves 192 B for each N.

N	XLA	Variant A	Variant B
2^7	1.42 KiB	1.19 KiB	1.23 KiB
2^{12}	1.89 KiB	1.66 KiB	1.70 KiB
2^{16}	2.27 KiB	2.03 KiB	2.08 KiB
2^{20}	2.64 KiB	2.41 KiB	2.45 KiB

Table 1. Communication costs with BLS12-381, using 1 KiB = 1024 bytes.

6.2 Computation

Let $m = \log_2(N)$. We count base-scalar pairs: one unit for every pair G^α present in a multi-scalar multiplication, one unit per element of a challenge-dependent basis construction, and one unit per fold. Group multiplications, field operations, hash-to-curve, etc. are excluded. This is the "ungrouped" convention of XLA, whose reported counts are $21N$ for the prover and $8N$ for the verifier before Straus-style grouping (then $18N$ and $7N$ after).

Prover: For XLA, the leading terms are six outer commitments ($6N$), the consistency messages $M_a, M_\rho, \mathbf{h}_w, T_1$ ($4N$), the two-family commitment $Q = \mathbf{b}^{\hat{\mathbf{a}}}\mathbf{g}^{\hat{\rho}}$ ($2N$), the basis constructions $\mathbf{G}_{\text{IPA}}, \mathbf{H}_{\text{IPA}}$ ($3N$), and the two-vector recursion, which at length n costs $2n+2$ message pairs plus n folds – these n terms add up to $6N + O(m)$. In total,

$$E_P^{\text{XLA}} \approx 6N + 4N + 2N + 3N + 6N = 21N.$$

Under the same convention for Variant A, the corresponding terms are five outer commitments ($5N$), $M_a, \mathbf{h}_w, M_{ct}$ ($3N$), $Q = \mathbf{b}^{\hat{\mathbf{a}}}$ (N), the single basis $\mathbf{G}_{\text{IPA}}$ ($2N$), and the one-sided recursion, which at length n costs n message pairs plus $n/2$ folds – these n terms may be summed to $3N - 3$. Hence, we tabulate

$$E_P^{\text{A}} \approx 5N + 3N + N + 2N + 3N = 14N, \quad E_P^{\text{B}} \approx 15N,$$

where Variant B replaces the single message M_{ct} by the pair (M_E, M_F). This is a reduction of the leading coefficient by 33% for Variant A and 29% for Variant B. Under XLA's Straus-grouped convention, which merges pairs such as those in Q and $\mathbf{G}_{\text{IPA}}$, the same count becomes roughly $18N$, versus $13N$ for Variant A or $14N$ for Variant B.

Verifier: All three protocols spend $2N$ pairs on V_E, V_F and N on $\mathbf{h}_w$. The variants' one-sided relation expands over the three length-N families $\mathbf{b}, \mathbf{g}, \mathbf{h}_w$, contributing $3N + 2m + O(1)$ pairs. Hence

$$E_V^{\mathrm{A}} = E_V^{\mathrm{B}} \approx 6N + 2m.$$

XLA's final relation expands over five families, in which the two $\mathbf{g}$ occurrences merge, leaving four length-N blocks: $E_V^{\mathrm{XLA}} \approx 7N + 2m$ grouped, or $8N + 2m$ ungrouped.

Protocol	Prover	Prover (Straus)	Verifier	Verifier (Straus)
XLA	$\approx 21N$	$\approx 18N$	$\approx 8N$	$\approx 7N$
Variant A	$\approx 14N$	$\approx 13N$	$\approx 6N$	$\approx 6N$
Variant B	$\approx 15N$	$\approx 14N$	$\approx 6N$	$\approx 6N$

Table 2. Leading base-scalar-pair counts. "Straus" applies XLA's joint-exponentiation convention. XLA values are as reported in [32].

6.3 Setup and Rounds

No new setup ceremony is required. Every committed polynomial keeps its XLA degree; the maximum, $N+3$, is attained by C_q as before, so an existing universal powers-of-τ SRS remains compatible. A specialized deployment can additionally omit the unused generators $\mathbf{k}, u_\rho$ and XLA's extra hash-to-curve challenge.

Variant A preserves the challenge order needed for the nested extractor. Variant B replaces one prover element M_{ct} by two elements (M_E, M_F) and sends (w, u) jointly, which saves one interactive round. As noted in Remark 4, this round saving derives from the protocol redesign, not automatically from the Fiat-Shamir reduction.

7 Application to Verifiable Mix-Net Voting

Anonymity-set privacy and mix-net privacy in a sense aim to solve two halves of the same problem. Firo's ledger privacy rests on Lelantus Spark [23], a descendant of the one-out-of-many paradigm [21], in which a spend proof demonstrates membership of the consumed coin among a large anonymity set, but never opens the coin itself. The announced Spark Assets extension uses this same design to implement private tokens, among other privacy oriented designs. In the case of voting, naturally a ballot cannot stay closed – for an election, the tabulator must eventually decrypt and publish the plaintexts to tally the votes, so a membership proof does not protect the voter's identity. What must be hidden in order to protect identities however is the link between the posted ciphertexts and the opened values; this is what mixing networks provide [10].

Encrypted ballots are sent through a sequence of servers, each of which permutes and rerandomizes the batch independently. So long as at least a single server performs their job honestly and does not reveal its permutation, the correspondence between each voter and their ballot remains hidden. Verifiability may be restored by requiring every server to publish a proof for R_{vs} [27]. Indeed, this exact shuffle-and-prove paradigm is employed practically in blockchain mixing services such as MixEth [29]. Once posted to a bulletin board or a blockchain, these proofs become permanent and publicly auditable, so the per-layer proof size is naturally what must be minimized – this cost is precisely what we reduce in this paper.

7.1 Shuffle-Then-Decrypt Elections

The standard architecture is a shuffle-then-decrypt procedure. In this model, a threshold ElGamal key Y is produced using distributed key generation among n_T trustees, with a reconstruction threshold t, defined by their use-case [28]. Each eligible voter encrypts their ballot $m_i \in \mathbb{G}_1$ as $C_i = \mathsf{Enc}_Y(m_i, r_i)$, then posts it to a public, add-only bulletin board. Ballot validity, voter eligibility, and the final proofs of correct decryption [9] are obligations of the voting system itself, not the shuffling procedure. Let $\mathbf{C}^{(0)}$ be the admitted batch of $N = 2^m$ ballots, which we assume to be a power of 2 for ease of notation. Mixing servers $S_1, \ldots, S_L$ act in series: S_j samples permutation vector $\pi_j \xleftarrow{\$} \Sigma_N$ and rerandomization vector $\boldsymbol{\rho}_j \xleftarrow{\$} \mathbb{F}^N$, then for each $i \in [N]$ publishes

$$C_i^{(j)} = C_{\pi_j(i)}^{(j-1)} \mathsf{Enc}_Y(1, \rho_{j,i}),$$

along with a noninteractive validity proof ϖ_j from Π'_{vs} for $(\mathsf{ck}, Y, \mathbf{C}^{(j-1)}, \mathbf{C}^{(j)}) \in R_{\text{vs}}$. Once $\varpi_1, \ldots, \varpi_L$ have been verified, the trustees decrypt the final batch $\mathbf{C}^{(L)}$ and post each plaintext with its decryption proof, for tallying. Privacy assumes that at least one server was honest and erased its permutation and rerandomization vector, while secrecy before counting assumes that there were fewer than t corrupted trustees. All of these components are verbatim from XLA, with the only alteration being the posted proof. Note that the SRS may be generated once, then reused across elections and mixing rounds.

Each Fiat-Shamir transcript is initialized as per Section 5.8, with the election identifier and round index j as part of the public context, while ck, Y, and both ciphertext batches must be digested by the hash before the first challenge, to ensure that a proof remains bound to its specific election or round. Variant A is the obvious choice, since the round saved in Variant B is only pertinent in an interactive setting.

7.2 Audit Cost

A single Variant-A proof occupies $(2\log_2 N + 8)\mathbb{G}_1 + 5\mathbb{F} = 96\log_2 N + 544$ B (Table 1), so the mixing stage adds

$$L \cdot (96\log_2 N + 544) \text{ B}$$

of proof data to the bulletin board, or $L \cdot (96 \log_2 N + 592)$ B for Variant B. Compared to XLA, we save a constant $240L$ B (Variant A) or $192L$ B (Variant B), independent of the ballot count. Table 3 tabulates the costs for an $L = 4$ election, where we retain XLA's accounting of 48 B compressed $\mathbb{G}_1$ elements and 32 B scalars over BLS12-381, while the HKR column represents $(6 \log_2 N + 8)\mathbb{G}_1 + 4\mathbb{F}$ with 32 B group elements. At $N = 2^{10}$, the four Variant-A proofs occupy 5.88 KiB – roughly 14% below XLA and 35% below the logarithmic shuffle of Hoffmann et al. – with each doubling of the ballot count adding only 96 B per server. Verifying the mixing stage costs L runs of the Π'_{vs} verifier, each of which is dominated by the $\approx 6N$ base-scalar pairs and three pairings under Section 6.2's conventions, compared to $\approx 7N$ with the same number of pairings for XLA. The prover's work is, using grouped counting, $\approx 13N$ per server.

N	Variant A	Variant B	XLA	HKR [22]
2^{10}	5.88	6.06	6.81	9.00
2^{16}	8.13	8.31	9.06	13.5
2^{20}	9.63	9.81	10.56	16.5

Table 3. Shuffle proof sizes in kibibytes, matching XLA's comparison convention, considering $L = 4$ servers, but excluding ballot-validity and decryption proofs.

8 Conclusion

XLA's consistency layer considered the rerandomization vector $\boldsymbol{\rho}$ as a full witness, even though its verification equations only employ the scalar $\rho^* = \langle \mathbf{a}, \boldsymbol{\rho} \rangle$. Removing that branch is safe, so long as one takes care with weak Fiat-Shamir: Proposition 1 shows that a scalar chosen after the ciphertext-merging challenge admits false statements, but binding ρ^* to the linear blinder slot of the shared commitment C_a fixes this. The fresh basis and the one-sided folding argument then tie every remaining commitment to a single response vector, and the AGM extraction recovers the componentwise rerandomizers from the positional challenge x, which permits a scalar witness to certify the vector relation.

The result is two modifications to XLA's consistency argument, all of which prove the same shuffle relation: Variant A costs $(2 \log_2 N + 8)\mathbb{G}_1 + 5\mathbb{F}$, constant 240 B below XLA, with grouped prover cost $\approx 13N$ and verifier cost $\approx 6N$; Variant B requires one additional group element to merge the (w, u) challenges, but attains one fewer interactive round, with Lemma 7 providing the four-point extractor. Both variants retain XLA's setup and SRS ceremony. Because of the constant per-proof reduction, the saving compound anywhere proofs accumulate – for instance, in the shuffle-then-decrypt elections we consider in Section 7. An L-server mix posts proofs that are $240L$ B fewer than XLA's – roughly 14% smaller at practical sizes, such as a 4-server mix with 2^{10} ballots – and 35% smaller than the logarithmic shuffle of Hoffmann et al. [22]

Acknowledgments

This research was supported by the National Science Foundation (Grant Number DMS-2342254). We have no competing interests to declare. To prepare this document for publication, AI was used strictly to refine the phrasing of human-written portions and to polish tables. This work is an accurate representation of the author's effort and intellectual contributions, and the authors accept all liability.

References

1. Attema, T., Cramer, R.: Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 513–543. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-56877-1_18
2. Attema, T., Fehr, S., Klooß, M.: Fiat-Shamir Transformation of Multi-Round Interactive Proofs. Journal of Cryptology **36**(4), 36 (2023)
3. Attema, T., Fehr, S., Klooß, M., Resch, N.: The Fiat-Shamir Transformation of $(\Gamma_1, \ldots, \Gamma_\mu)$-Special-Sound Interactive Proofs. J. Cryptol. **39**(2), 19 (2026). https://doi.org/10.1007/S00145-025-09569-Y
4. Bayer, S., Groth, J.: Efficient Zero-Knowledge Argument for Correctness of a Shuffle. In: Advances in Cryptology – EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 263–280. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_17
5. Bernhard, D., Pereira, O., Warinschi, B.: How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In: Wang, X., Sako, K. (eds.) Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 626–643. Springer (2012). https://doi.org/10.1007/978-3-642-34961-4_38
6. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting. In: Advances in Cryptology – EUROCRYPT 2016, Part II. Lecture Notes in Computer Science, vol. 9666, pp. 327–357. Springer (2016). https://doi.org/10.1007/978-3-662-49896-5_12
7. Bowe, S., Gabizon, A., Miers, I.: Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model. Cryptology ePrint Archive, Paper 2017/1050 (2017), https://eprint.iacr.org/2017/1050
8. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short Proofs for Confidential Transactions and More. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA. pp. 315–334. IEEE Computer Society (2018). https://doi.org/10.1109/SP.2018.00020
9. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: Brickell, E.F. (ed.) Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. Lecture Notes in Computer Science, vol. 740, pp. 89–105. Springer (1992). https://doi.org/10.1007/3-540-48071-4_7

10. Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM **24**(2), 84–90 (February 1981). https://doi.org/10.1145/358549.358563
11. Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter Proofs for a Privacy-Enhanced Distributed Ledger. IEEE Access **10**, 42067–42082 (2022). https://doi.org/10.1109/ACCESS.2022.3167806
12. Dao, Q., Miller, J., Wright, O., Grubbs, P.: Weak Fiat-Shamir Attacks on Modern Proof Systems. In: 44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023. pp. 199–216. IEEE (2023). https://doi.org/10.1109/SP46215.2023.10179408
13. Eagen, L., Kanjalkar, S., Ruffing, T., Nick, J.: Bulletproofs++: Next Generation Confidential Transactions via Reciprocal Set Membership Arguments. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14655, pp. 249–279. Springer (2024). https://doi.org/10.1007/978-3-031-58740-5_9
14. Ethereum Foundation Cryptography Research Team: Curdleproofs: A Shuffle Argument Protocol. https://github.com/asn-d6/curdleproofs (2022)
15. Fuchsbauer, G., Kiltz, E., Loss, J.: The Algebraic Group Model and its Applications. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 33–62. Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_2
16. Furukawa, J., Sako, K.: An Efficient Scheme for Proving a Shuffle. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 368–387. Springer (2001). https://doi.org/10.1007/3-540-44647-8_22
17. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge. Cryptology ePrint Archive, Paper 2019/953 (2019), https://eprint.iacr.org/2019/953
18. Groth, J.: A Verifiable Secret Shuffle of Homomorphic Encryptions. Journal of Cryptology **23**(4), 546–579 (2010). https://doi.org/10.1007/s00145-010-9067-9
19. Groth, J.: On the Size of Pairing-Based Non-interactive Arguments. In: Advances in Cryptology – EUROCRYPT 2016. LNCS, vol. 9666, pp. 305–326. Springer (2016). https://doi.org/10.1007/978-3-662-49896-5_11
20. Groth, J., Ishai, Y.: Sub-Linear Zero-Knowledge Argument for Correctness of a Shuffle. In: Smart, N.P. (ed.) Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4965, pp. 379–396. Springer (2008). https://doi.org/10.1007/978-3-540-78967-3_22
21. Groth, J., Kohlweiss, M.: One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. pp. 253–280. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_9

22. Hoffmann, M., Klooß, M., Rupp, A.: Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019. pp. 2093–2110. ACM (2019). https://doi.org/10.1145/3319535.3354251
23. Jivanyan, A., Feickert, A.: Lelantus Spark: Secure and Flexible Private Transactions. In: Matsuo, S., Gudgeon, L., Klages-Mundt, A., Hernandez, D.P., Werner, S., Haines, T., Essex, A., Bracciali, A., Sala, M. (eds.) Financial Cryptography and Data Security. FC 2022 International Workshops - CoDecFin, DeFi, Voting, WTSC, Grenada, May 6, 2022, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13412, pp. 409–447. Springer (2022). https://doi.org/10.1007/978-3-031-32415-4_28
24. Jones, M.: Logos Research 101: Transforming an Interactive Protocol to a Noninteractive Argument. https://vac.dev/rlog/vac101-fiat-shamir/, accessed: 2026-11-13
25. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: Advances in Cryptology – ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 177–194. Springer (2010). https://doi.org/10.1007/978-3-642-17373-8_11
26. Miller, J.: The Frozen Heart Vulnerability in Bulletproofs. https://blog.trailofbits.com/2022/04/15/the-frozen-heart-vulnerability-in-bulletproofs/, accessed: 2024-10-15
27. Neff, C.A.: A Verifiable Secret Shuffle and its Application to e-Voting. In: Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01). pp. 116–125. ACM (2001). https://doi.org/10.1145/501983.502000
28. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party. In: EUROCRYPT '91: Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques. Lecture Notes in Computer Science, vol. 547, pp. 522–526. Springer-Verlag (1991). https://doi.org/10.1007/3-540-46416-6_47
29. Seres, I.A., Nagy, D.A., Buckland, C., Burcsi, P.: MixEth: Efficient, Trustless Coin Mixing Service for Ethereum. In: Danos, V., Herlihy, M., Potop-Butucaru, M., Prat, J., Piergiovanni, S.T. (eds.) International Conference on Blockchain Economics, Security and Protocols, Tokenomics 2019, Paris, France, May 6-7, 2019. OASIcs, vol. 71, pp. 13:1–13:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/OASICS.TOKENOMICS.2019.13
30. Terelius, B., Wikström, D.: Proofs of Restricted Shuffles. In: Progress in Cryptology – AFRICACRYPT 2010. LNCS, vol. 6055, pp. 100–113. Springer (2010). https://doi.org/10.1007/978-3-642-12678-9_7
31. Wikström, D.: Verificatum Mix-Net. https://www.verificatum.org (2010)
32. Xue, Y., Lu, X., Au, M.H.: Compact Verifiable Shuffles for ElGamal Ciphertexts. Cryptology ePrint Archive, Paper 2026/805 (2026), https://eprint.iacr.org/2026/805