Sumcheck bit packing

Matteo Frigo
Google

abhi shelat
Google

November 4, 2026

Abstract

Circuit-based zero-knowledge proofs over large fields waste most of a field element when they commit to a single bit. We study the problem of committing to *packed* witnesses, in which m bits are stored in one element of $GF(2^m)$, while still allowing the proof system to verify linear claims about the individual bits.

Prior approaches to this problem either (a) scale to packing only 3–4 bits [Fs26], (b) require tensor-structured claims and additional sumcheck rounds that incur larger soundness error [DP25, DP26], or (c) rely on Frobenius automorphisms [Koc26, SG26]. In contrast, we present a simpler reduction that exploits only the $GF(2)$-linearity of the field trace and achieves soundness error 2^{-m} for arbitrary bit-linear claims. We then use a characterization of trace-zero field elements to make the reduction zero-knowledge over any hiding linear commitment scheme: whereas previous constructions verify $O(m^2)$ [Bin26, SG26] or $O(m)$ [Koc26] quadratic constraints in the commitment scheme, our protocol requires *only* linear constraints.

1 Introduction

In a circuit-based zero-knowledge (ZK) protocol, the verifier holds a circuit C and a theorem statement x, and the prover aims to convince the verifier that it knows a witness w such that $C(x, w) = 0$ without revealing any other information about w. The most efficient known protocols for this problem rely on the sumcheck protocol for part of the verification. In this setting, the prover first commits to the witness $w = (w_0, \ldots, w_{n-1})$ over a finite field $\mathbb{F}$, runs several protocol steps ending at a claimed value $Y \in \mathbb{F}$, and finishes the protocol with the obligation of showing that

$$Y = \sum_{i=0}^{n-1} R_i w_i,$$

where w_i is the ith witness element and $R_i \in \mathbb{F}$ are public coefficients determined by the verifier's challenges.

A common problem in such protocols concerns the efficiency of operating on *bits*. In order to attain sufficient soundness, the ZK protocol often employs a field $\mathbb{F}$ with $|\mathbb{F}| \geq 2^{128}$, which requires that committed witnesses be encoded as elements of this large field. On the other hand, the circuit C being verified is often best expressed over boolean values $\{0, 1\}$. The standard way to handle this is to commit to *bit* witnesses, add constraints to the circuit that verify the witnesses are bits, and proceed to simulate the boolean AND and OR operations algebraically in $\mathbb{F}$. One substantial cost of this approach is the loss in space efficiency: while each w_i is committed as an element of $\mathbb{F}$, here it is being used to convey only a single bit. In practice, this inflates the cost of committing to the witness by roughly $100\times$.

One approach to overcome this problem is to *pack* k bits into a single field element. Frigo and shelat [Fs26] apply this approach by devising a custom set of sparse polynomials $p_1, \ldots, p_k$ where $p_i(w)$ outputs the ith bit of the integer representation of an element $w \in \mathbb{F}$. These polynomials are first evaluated in the circuit on the packed witness to produce the k bits, after which the original circuit is evaluated. While this technique works over any finite field for small packing values, such as $k = 3$ or 4, it increases the depth of the circuit that is evaluated and still incurs a commitment overhead of roughly $40\times$.

Diamond and Posen [DP25, DP26] introduce a *ring-switching* technique that packs m bits into a single element of a binary extension field $\mathbb{F} = GF(2^m)$, reducing evaluation claims on the unpacked bits to evaluation claims on the packed field elements via tensor-algebra identities and sumcheck. However, their reduction requires tensor-structured coefficients and leaves zero-knowledge as an open problem; subsequent works such as Flock [BRW26] and Bloemen's concurrent dual-packing note [Blo26] likewise omit zero-knowledge, while Diamond [Dia25] adds zero-knowledge only to the underlying large-field commitment scheme. To make ring-switching itself zero-knowledge, the Binius64 implementation [Bin26] commits to the $m \times m$ unpacked bit matrix of a blinding pad and enforces the elements are bits through m^2 (e.g. 16,384 at $m = 128$) hidden-wire quadratic constraints per packed claim. A recent note by Kochergin [Koc26] eliminates the explicit bit-transpose via linearized polynomials and Horner's rule in the Frobenius, reducing the outer proof system's overhead to $m - 1$ quadratic constraints (127 at $m = 128$) and $3(m - 1)$ auxiliary variables. The concurrent GordieHowe system [SG26] blinds the same Frobenius sum without Horner's rule, which costs $d(d - 1)/2$ quadratic constraints to pack d bits and therefore limits it to packing 16 bits at a time.

In this paper, we present a simpler and direct bit-packing reduction together with a compact zero-knowledge extension that requires *no* quadratic constraints to be verified in the commitment scheme.

Trace reduction (§3) Our approach exploits the $GF(2)$-linearity of the trace map $Tr : \mathbb{F} \to GF(2)$ and a dual basis in order to construct an efficient *bit plucker*. We then convert each claim on a bit into an equivalent form that verifies that a linear expression over the

packed witnesses has trace zero. We then batch these claims with soundness error 2^{-m}. Our approach does not require an inner sumcheck, tensor structure on the coefficients, or a Frobenius automorphism. When the coefficients do have tensor structure, as in many applications that use sumcheck, all $O(Mm)$ field multiplications reduce to pure $GF(2)$ additions (XORs).

Zero-knowledge blinding (§4) To achieve the zero-knowledge property, we exploit a Hilbert identity that $Tr(x) = 0$ if and only if there exists $t \in \mathbb{F}$ such that $x = t(t+1)$, combined with a standard *blinding* technique. The prover commits to *pads* t_k and masks each trace-zero claim with $t_k(t_k+1)$, i.e., we simply add a random trace-zero element to each expression being validated. Although $t_k(t_k+1) = t_k^2 + t_k$ is quadratic in the pads t_k, batching the claims with squared challenges δ_k^2 rewrites $\sum_k \delta_k^2 t_k^2 = (\sum_k \delta_k t_k)^2$ in $\mathbb{F}$, which reduces all t_k^2 terms (across any number of batched claims) to a *single* committed mask r and its square r^2. These can be verified in a purely linear commitment scheme.

2 The packing problem

Let $\mathbb{F} = GF(2^m)$ be the finite field chosen for the ZK protocol; let $\beta = \{\beta_0, \ldots, \beta_{m-1}\}$ be a basis for $\mathbb{F}$ over $GF(2)$. Every element $z \in \mathbb{F}$ has a unique coordinate expansion in β:

$$z = \sum_{k=0}^{m-1} d_k(z) \cdot \beta_k, \tag{1}$$

where $d_k : \mathbb{F} \to GF(2) \subset \mathbb{F}$ is the kth coordinate map. For concreteness, one may take $\beta = \{1, x, x^2, \ldots, x^{m-1}\}$ to be the polynomial basis, which makes $d_k(z)$ the kth bit of the standard representation of z; however, all of our results hold for *any* basis β of $\mathbb{F}$ over $GF(2)$—in particular, β need not contain 1. (Only the minor implicit-t_0 optimization in Lemma 4.2 assumes β is indexed so that $d_0(1) = 1$, which is always achievable by re-ordering since $1 = \sum_k d_k(1)\beta_k \neq 0$ has at least one non-zero coordinate.)

Suppose we are given $n = m \cdot M$ witness bits $x_{i,j} \in \{0,1\}$ for $i \in [0, M-1]$ and $j \in [0, m-1]$. We define the *packed witnesses* $X_i \in \mathbb{F}$ as:

$$X_i = \sum_{j=0}^{m-1} \beta_j \cdot x_{i,j} \quad (\forall i \in [0, M-1]), \tag{2}$$

so that $d_j(X_i) = x_{i,j}$. Each X_i packs m witness bits into a single field element.

Remark 2.1. *The packing map $(x_{i,0}, \ldots, x_{i,m-1}) \mapsto \sum_j \beta_j x_{i,j}$ is a bijection from $\{0,1\}^m$ onto $\mathbb{F}$ with inverse $X_i \mapsto (d_0(X_i), \ldots, d_{m-1}(X_i))$. Thus an* arbitrary *committed element $X_i \in \mathbb{F}$ always defines m unique bits $x_{i,j} = d_j(X_i) \in \{0,1\}$, and our protocols require no separate "bit constraints" in the circuit C or in the commitment scheme. In all soundness statements below, $x_{i,j}$ denotes $d_j(X_i)$ for whatever $X_i \in \mathbb{F}$ a (possibly malicious) Prover committed to.*

Assume that the outer protocol (such as a sumcheck) reduces theorem verification to checking a linear identity over the unpacked bits:

$$Y = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} R_{i,j} \cdot d_j(X_i), \tag{3}$$

where $R_{i,j} \in \mathbb{F}$ are public coefficients (often in tensor form $R_{i,j} = C_i D_j$). Our goal is to reduce (3) to verifying linear relations directly over the committed packed vector $\{X_i\}_{i=0}^{M-1}$:

$$Y' = \sum_{i=0}^{M-1} R'_i X_i \tag{4}$$

with public coefficients $R'_i \in \mathbb{F}$ (augmented by $m+1$ blinding elements in the zero-knowledge setting), incurring soundness loss at most 2^{-m}.

Order of events In sumcheck-based arguments (§4.1), the Prover commits to the packed witnesses *first*; Verifier challenges then fix $R_{i,j}$; and the Prover claims Y *afterwards*, as the final message of the sumcheck. All soundness bounds in this paper therefore hold for an arbitrary Y chosen adversarially after $R_{i,j}$ are fixed.

2.1 Preliminaries: coordinates and linear batching

Here we recount basic, folklore results that we use. Because $\mathbb{F}$ is an m-dimensional vector space over $GF(2)$ with basis β, a field equation whose unknowns are $GF(2)$-valued decomposes into m bit-level equations:

Lemma 2.2 (Coordinate decomposition)**.** *For any coefficients $R_{i,j} \in \mathbb{F}$, bits $x_{i,j} \in GF(2)$, and $Y \in \mathbb{F}$, the field equation $Y = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} R_{i,j} x_{i,j}$ holds if and only if*

$$d_k(Y) = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} d_k(R_{i,j})\, x_{i,j} \quad (\forall k \in [0, m-1]).$$

Proof. Expanding $R_{i,j} = \sum_{k=0}^{m-1} d_k(R_{i,j})\beta_k$ and using $x_{i,j} \in GF(2)$ to move scalars past β_k yields

$$\sum_{i,j} R_{i,j} x_{i,j} = \sum_{k=0}^{m-1} \left(\sum_{i,j} d_k(R_{i,j}) x_{i,j} \right) \beta_k.$$

Comparing with $Y = \sum_{k=0}^{m-1} d_k(Y)\beta_k$, uniqueness of coordinates in β gives the claim. □

Standard batching To verify many linear constraints over $\mathbb{F}$ at once, we use standard random linear batching, stated here in an affine form that also covers squared challenges in $GF(2^m)$:

Lemma 2.3 (Linear batching)**.** *Let $\mathbb{F}$ be a finite field and $N \geq 1$. For any non-zero vector $\vec{E} = (E_0, \ldots, E_{N-1}) \in \mathbb{F}^N$, any fixed scalar $E^* \in \mathbb{F}$, and a uniform random challenge vector $\vec{\gamma} = (\gamma_0, \ldots, \gamma_{N-1}) \leftarrow \mathbb{F}^N$:*

$$\Pr\left[\sum_{k=0}^{N-1} \gamma_k E_k = E^*\right] = \frac{1}{|\mathbb{F}|}.$$

When $\mathbb{F} = GF(2^m)$, squaring $\delta \mapsto \delta^2$ is a bijection of $\mathbb{F}$, so the identical bound holds when $\delta_0, \ldots, \delta_{N-1} \leftarrow \mathbb{F}$ are sampled uniformly and $\gamma_k = \delta_k^2$.

Proof. Pick k^* with $E_{k^*} \neq 0$. For any fixed $\{\gamma_k\}_{k \neq k^*}$, the equation $\sum_k \gamma_k E_k = E^*$ holds only for the single value $\gamma_{k^*} = E_{k^*}^{-1}(E^* - \sum_{k \neq k^*} \gamma_k E_k)$. Since γ_{k^*} is uniform on $\mathbb{F}$ (including when $\gamma_{k^*} = \delta_{k^*}^2$ over $GF(2^m)$, as $\delta \mapsto \delta^2$ is an additive homomorphism with kernel $\{0\}$ and hence a bijection), this happens with probability $1/|\mathbb{F}|$. □

3 Trace-Based Bit Packing

The field trace $Tr : \mathbb{F} \to GF(2)$ is a well-studied surjective $GF(2)$-linear map that satisfies the properties described in Lemma 3.1. There are several equivalent ways to define the trace, e.g., algebraically via the Frobenius automorphism as $Tr(z) = \sum_{s=0}^{m-1} z^{2^s}$, or as the linear-algebraic trace of the $GF(2)$-linear multiplication map $u \mapsto z \cdot u$ on $\mathbb{F}$. Although prior work computes with the Frobenius definition, we never need to apply Frobenius powers (z^{2^s}) in either our protocol or its proofs, and there is a much simpler way to evaluate Tr. Exploiting $GF(2)$-linearity over the coordinate expansion (1),

$$Tr(z) = \sum_{\ell=0}^{m-1} d_\ell(z)\, Tr(\beta_\ell).$$

Once the fixed m-bit mask $(Tr(\beta_0), \ldots, Tr(\beta_{m-1})) \in GF(2)^m$ is precomputed for the chosen basis β, evaluating $Tr(z)$ is simply the parity of the bitwise AND of z with that mask—for instance, $d_{121}(z) \oplus d_{127}(z)$ in the standard representation of $GF(2^{128}) \cong GF(2)[x]/(x^{128} + x^7 + x^2 + x + 1)$.

In our proofs below, we exploit only the following properties of the trace:

Lemma 3.1 (Properties of the trace)**.** *On $\mathbb{F} = GF(2^m)$, the field trace $Tr : \mathbb{F} \to GF(2)$ is a surjective $GF(2)$-linear map satisfying:*

1. *$Tr(x + y) = Tr(x) + Tr(y)$ and $Tr(c \cdot x) = c \cdot Tr(x)$ for all $x, y \in \mathbb{F}$ and $c \in GF(2)$;*
2. *$Tr(z^2) = Tr(z)$ for all $z \in \mathbb{F}$;*

3. *There exists $\rho \in \mathbb{F}$ with $Tr(\rho) = 1$, and for any basis $\beta = \{\beta_0, \ldots, \beta_{m-1}\}$ of $\mathbb{F}$ over $GF(2)$, there exists a unique dual basis $\alpha = \{\alpha_0, \ldots, \alpha_{m-1}\}$ such that $Tr(\alpha_k \cdot \beta_j) = \delta_{k,j}$ (the Kronecker delta) for all $k, j \in [0, m-1]$.*

(Note that $Tr(c \cdot x) = c \cdot Tr(x)$ does not hold for general $c \in \mathbb{F}$; below it is only applied with $c \in \{0, 1\}$.)

A bit-plucker In particular, the dual-basis property in Lemma 3.1(3) turns the trace into an algebraic *bit-plucker*: multiplying a packed element $X \in \mathbb{F}$ by α_k before taking the trace projects onto β_k and annihilates every other basis coordinate, extracting the kth bit $d_k(X)$:

Lemma 3.2 (Bit-plucker)**.** *For every $X \in \mathbb{F}$ and $k \in [0, m-1]$, $d_k(X) = Tr(\alpha_k \cdot X)$.*

Proof. Expanding $X = \sum_{\ell=0}^{m-1} \beta_\ell \cdot d_\ell(X)$ with $d_\ell(X) \in GF(2)$ and applying Lemma 3.1(1,3),

$$Tr(\alpha_k \cdot X) = \sum_{\ell=0}^{m-1} Tr(\alpha_k \cdot \beta_\ell) \cdot d_\ell(X) = \sum_{\ell=0}^{m-1} \delta_{k,\ell} \cdot d_\ell(X) = d_k(X). \qquad \square$$

By $GF(2)$-linearity, any bit-level inner product over the unpacked witness bits is equivalent to a single trace-zero condition on a linear combination of the packed witnesses:

Lemma 3.3 (Single bit-claim via trace)**.** *Fix $\rho \in \mathbb{F}$ with $Tr(\rho) = 1$. For any target bit $c \in GF(2)$ and bit weights $a_{i,j} \in GF(2)$ ($i \in [0, M-1]$, $j \in [0, m-1]$), let $Q_i = \sum_{j=0}^{m-1} a_{i,j}\alpha_j \in \mathbb{F}$. Then*

$$c = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} a_{i,j}\, d_j(X_i) \iff Tr\left(\sum_{i=0}^{M-1} Q_i X_i + \rho c \right) = 0.$$

Proof. Since $a_{i,j}, c \in GF(2)$, Lemma 3.1(1) and Lemma 3.2 give

$$Tr\left(\sum_{i=0}^{M-1} Q_i X_i + \rho c \right) = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} a_{i,j} Tr(\alpha_j X_i) + c\, Tr(\rho) = \sum_{i=0}^{M-1} \sum_{j=0}^{m-1} a_{i,j}\, d_j(X_i) + c,$$

which vanishes in $GF(2)$ if and only if $c = \sum_{i,j} a_{i,j} d_j(X_i)$. $\square$

To verify a field evaluation claim $Y = \sum_{i,j} R_{i,j} d_j(X_i)$—or more generally $L \geq 1$ field evaluation claims simultaneously—Lemma 2.2 decomposes each field claim into m coordinate bit-claims, producing $N = Lm$ bit-claims $\{c_k = \sum_{i,j} a_{i,j,k} d_j(X_i)\}_{k=0}^{N-1}$ (for $L = 1$, $N = m$ with $c_k = d_k(Y)$ and $a_{i,j,k} = d_k(R_{i,j})$). Defining

$$Q_{i,k} = \sum_{j=0}^{m-1} a_{i,j,k}\alpha_j \quad \text{and} \quad Z_k = \sum_{i=0}^{M-1} Q_{i,k} X_i + \rho c_k \qquad (\forall k \in [0, N-1]), \tag{5}$$

Public Input: $(Y, R_{i,j})$: Target value $Y \in \mathbb{F}$, coefficients $R_{i,j} \in \mathbb{F}$ for $i \in [0, M-1]$ and $j \in [0, m-1]$, basis β with dual basis α, fixed $\rho \in \mathbb{F}$ with $Tr(\rho) = 1$, and commitments to packed witnesses $X_i \in \mathbb{F}$ for $i \in [0, M-1]$.
Protocol Steps:

1. For each $k \in [0, m-1]$, let $c_k = d_k(Y)$ and $Q_{i,k} = \sum_{j=0}^{m-1} d_k(R_{i,j})\alpha_j$. The Prover computes
$$Z_k = \sum_{i=0}^{M-1} Q_{i,k} X_i + \rho c_k \quad (\forall k \in [0, m-1])$$
and sends $\{Z_k\}_{k=0}^{m-1}$ to the Verifier.
2. The Verifier checks that $Tr(Z_k) = 0$ for every $k \in [0, m-1]$. If any check fails, reject.
3. The Verifier samples $\vec{\gamma} = (\gamma_0, \ldots, \gamma_{m-1}) \leftarrow \mathbb{F}^m$ and sends it to the Prover.
4. Both parties compute
$$Z' = \sum_{k=0}^{m-1} \gamma_k (Z_k + \rho c_k) \quad \text{and} \quad r'_i = \sum_{k=0}^{m-1} \gamma_k Q_{i,k} \quad (\forall i \in [0, M-1]),$$
and verify the linear relation over the committed witnesses:
$$Z' \stackrel{?}{=} \sum_{i=0}^{M-1} r'_i X_i.$$

Figure 1: The Trace-Based Packing Protocol.

all L field claims hold if and only if $Tr(Z_k) = 0$ for every $k \in [0, N-1]$. The Prover sends $\{Z_k\}_{k=0}^{N-1}$, the Verifier checks $Tr(Z_k) = 0$ for each k, and both parties batch the N linear equations $Z_k + \rho c_k = \sum_i Q_{i,k} X_i$ into a single commitment check using $\vec{\gamma} \leftarrow \mathbb{F}^N$. Figure 1 summarizes the protocol for $L = 1$ ($N = m$); since each Z_k has trace 0, the Prover's message $\{Z_k\}_{k=0}^{m-1}$ compresses to $m(m-1)$ bits ($m-1$ field elements, or m elements uncompressed as in ring-switching).

Theorem 3.4. *The protocol in Figure 1 is complete and sound. Specifically, for any committed X_i, any coefficients $R_{i,j}$, and any Y (possibly chosen after $R_{i,j}$ are fixed), if $Y \neq \sum_{i,j} R_{i,j} d_j(X_i)$, then the Verifier rejects with probability at least $1 - 2^{-m}$ over $\vec{\gamma} \leftarrow \mathbb{F}^m$. (The identical bound holds when batching $N = Lm$ bit-claims across L field outputs.)*

Proof. Completeness follows from Lemma 2.2, Lemma 3.3, and $Z_k + \rho c_k = \sum_i Q_{i,k} X_i$ in characteristic 2. For soundness, if $Y \neq \sum_{i,j} R_{i,j} d_j(X_i)$, then by Lemma 2.2 and Lemma 3.3

some index k^* has $Tr(Z_{k^*}) = 1$. Let $\{\tilde{Z}_k\}$ be the Prover's Step 1 message. If $Tr(\tilde{Z}_{k^*}) \neq 0$, Step 2 rejects immediately; otherwise $Tr(\tilde{Z}_{k^*}) = 0 \neq Tr(Z_{k^*})$, so $\tilde{Z}_{k^*} \neq Z_{k^*}$ and Step 4 accepts with probability 2^{-m} by Lemma 2.3 (with $E_k = \tilde{Z}_k + Z_k$ and $E^* = 0$). □

A note on efficiency As noted above, each trace check in Step 2 is a single bit-mask parity check. For arbitrary $R_{i,j}$, both parties perform $M \cdot m$ field multiplications plus $O(Mm^2)$ bit-XORs ($Mm^2/8$ with byte lookup tables): the Prover forms $Q_{i,k} = \sum_{j=0}^{m-1} d_k(R_{i,j})\alpha_j$ via $GF(2)$ additions of dual-basis constants and multiplies $Q_{i,k}X_i$, while the Verifier can avoid forming $Q_{i,k}$ altogether by tabulating $\Gamma(R) = \sum_{k=0}^{m-1} d_k(R)\gamma_k$ from $\vec{\gamma}$ and computing $r'_i = \sum_{j=0}^{m-1} \alpha_j \Gamma(R_{i,j})$.

When $R_{i,j} = C_i D_j$ has tensor form, expanding $C_i = \sum_{\ell=0}^{m-1} d_\ell(C_i)\beta_\ell$ inside $d_k(C_i D_j)$ factors $Q_{i,k}$ as

$$Q_{i,k} = \sum_{\ell=0}^{m-1} d_\ell(C_i)\, \mu_{k,\ell}, \qquad \mu_{k,\ell} = \sum_{j=0}^{m-1} d_k(\beta_\ell D_j)\, \alpha_j. \tag{6}$$

The Verifier precomputes $\{\mu_{k,\ell}\}$ (m^2 field multiplications $\beta_\ell D_j$ plus bit-XORs) and $g'_\ell = \sum_{k=0}^{m-1} \gamma_k \mu_{k,\ell}$ (m^2 multiplications), so each coefficient $r'_i = \sum_{\ell=0}^{m-1} g'_\ell d_\ell(C_i)$ is a pure $GF(2)$ sum of $\{g'_\ell\}$: the $O(Mm)$ per-witness work involves *no* field multiplications (only $Mm/8$ XORs via 8-bit lookup tables). Likewise, the Prover first computes the bit-transposed sums $W_\ell = \sum_{i=0}^{M-1} d_\ell(C_i)X_i$ (or via $Y_j = \sum_i x_{i,j}C_i$ followed by an $m \times m$ bit transpose) using Mm XORs, and then $\sum_i Q_{i,k}X_i = \sum_{\ell=0}^{m-1} \mu_{k,\ell}W_\ell$ in m^2 field multiplications.

4 Zero-Knowledge Packing

The protocol of Figure 1 is not zero-knowledge because $\{Z_k\}$ depends on the witness $\vec{X}$. To make it zero-knowledge while preserving the Verifier's clear-text check $Tr(U_k) = 0$, we mask each Z_k with a random trace-zero element of the form $t(t+1)$. This characterization of trace-zero elements follows from the additive form of Hilbert's Theorem 90 specialized to $GF(2^m)$. Over finite fields, its proof is elementary enough to include here.

Lemma 4.1 (Trace-zero property)**.** *For any $z \in \mathbb{F} = GF(2^m)$,*

$$Tr(z) = 0 \iff \exists t \in \mathbb{F} \text{ such that } t(t+1) = z.$$

Proof. Define $\wp : \mathbb{F} \to \mathbb{F}$ by $\wp(t) = t^2 + t = t(t+1)$. In characteristic 2, $(u+v)^2 = u^2 + v^2$, so $\wp$ is $GF(2)$-linear, and Lemma 3.1(1,2) gives $Tr(\wp(t)) = Tr(t^2) + Tr(t) = 0$ for all $t \in \mathbb{F}$, i.e., $\mathrm{im}(\wp) \subseteq \ker(Tr)$. Since $t(t+1) = 0 \iff t \in \{0,1\}$, we have $|\ker(\wp)| = 2$ and $|\mathrm{im}(\wp)| = |\mathbb{F}|/2 = 2^{m-1}$. Because $Tr : \mathbb{F} \to GF(2)$ is surjective, $|\ker(Tr)| = 2^{m-1}$ as well, so $\mathrm{im}(\wp) = \ker(Tr)$. □

Given Lemma 4.1, we proceed to construct an efficient zero-knowledge protocol for bit packing over any hiding *linear* commitment scheme in two main steps, followed by a minor optimization.

1. Self-enforcing trace-zero pads The Prover masks each Z_k as

$$U_k = Z_k + t_k(t_k + 1) = Z_k + t_k^2 + t_k. \tag{7}$$

At first glance, this masking may seem overly complex. Consider the more natural alternative in which the Prover commits to pads p_k that are supposed to satisfy $Tr(p_k) = 0$, a constraint that the linear commitment scheme *does not* check. A malicious Prover could commit to pads with $Tr(p_k) = 1$, wait to learn $R_{i,j}$, and shift its claimed Y accordingly (recall the order of events in §2). Forcing the pad to take the form $t_k^2 + t_k$ eliminates this attack completely: $Tr(t_k^2 + t_k) = 0$ holds *for every* $t_k \in \mathbb{F}$, so no committed value can introduce a trace offset, and $\{U_k\}$ can be sent in the same round as Y.

2. Collapsing squares via Freshman's Dream It now remains to verify the quadratic terms t_k^2 in (7). Our goal is to avoid using any quadratic constraints in the commitment scheme. To do so, the Verifier samples $\vec{\delta} = (\delta_0, \ldots, \delta_{m-1}) \leftarrow \mathbb{F}^m$ and sets the batching weights to $\gamma_k = \delta_k^2$. By Freshman's Dream in characteristic 2,

$$\sum_{k=0}^{m-1} \gamma_k t_k^2 = \sum_{k=0}^{m-1} \delta_k^2 t_k^2 = \left(\sum_{k=0}^{m-1} \delta_k t_k \right)^2.$$

The Prover commits at the outset to one extra random mask $r \leftarrow \mathbb{F}$ and its square $b = r^2$, and opens the blinded sum $A = r + \sum_k \delta_k t_k$. Then $A^2 + b = \sum_k \gamma_k t_k^2$ is linear in the committed element b. Crucially, the commitment scheme need not enforce $b = r^2$: any discrepancy $\Delta_b = r^2 + b$ is independent of $\vec{\gamma}$ and is caught automatically by the random batching weights $\gamma_k = \delta_k^2$ (Lemma 2.3). (More generally, the same pair $(r, b = r^2)$ and squared challenges δ_j^2 discharge *any* batch of committed squaring constraints $q_j = \ell_j^2$ over $GF(2^m)$ using only linear commitment checks.)

3. Implicit t_0 from the unused kernel bits As a minor commitment-size optimization, we avoid committing to t_0 by making its value implicit in $t_1, \ldots, t_{m-1}$. Specifically, each $U_k \in \ker(Tr)$ has $m - 1$ bits of entropy, so $(U_0, \ldots, U_{m-1})$ has $m(m-1)$ bits in total. Meanwhile, each committed $t_k \in \mathbb{F}$ ($k \in [1, m-1]$) has m bits, of which $\wp(t_k) = t_k^2 + t_k$ leaves 1 bit (its $GF(2)$-coset) unused. Defining $t_0 = \sum_{k=1}^{m-1} \beta_k t_k$ as a public $\mathbb{F}$-linear combination of $t_1, \ldots, t_{m-1}$ recycles those $m - 1$ bits to blind U_0 without committing to t_0:

Lemma 4.2. *Assume $d_0(1) = 1$. Define $\Phi : \mathbb{F}^{m-1} \to (\ker(Tr))^m$ by*

$$\Phi(t_1, \ldots, t_{m-1}) = \big(\wp(t_0),\, \wp(t_1),\, \ldots,\, \wp(t_{m-1})\big), \qquad t_0 = \sum_{k=1}^{m-1} \beta_k t_k, \quad \wp(z) = z^2 + z.$$

Then Φ is a bijection.

Proof. Both $\mathbb{F}^{m-1}$ and $(\ker(Tr))^m$ are $GF(2)$-vector spaces of size $2^{m(m-1)}$, and Φ is $GF(2)$-linear because $(t_1, \ldots, t_{m-1}) \mapsto t_0$ and $\wp$ are $GF(2)$-linear. If $\Phi(t_1, \ldots, t_{m-1}) = \vec{0}$, then $\wp(t_k) = 0$ for all $k \in [0, m-1]$, so $t_0, t_1, \ldots, t_{m-1} \in \{0, 1\} = GF(2)$. If $t_0 = 1$, then $1 = \sum_{k=1}^{m-1} t_k \beta_k$ with $t_k \in GF(2)$, contradicting $d_0(1) = 1$ by uniqueness of coordinates in β. Hence $t_0 = 0$, and linear independence of $\{\beta_1, \ldots, \beta_{m-1}\}$ over $GF(2)$ forces $t_1 = \cdots = t_{m-1} = 0$. Thus $\ker(\Phi) = \{\vec{0}\}$ and Φ is a bijection. □

Figure 2 gives the complete protocol. Because Φ is a bijection, $\Phi^{-1}(\vec{U} + \vec{Z})$ uniquely determines $(t_1, \ldots, t_{m-1})$ once $\{U_k\}$ is sent, which is why the independent mask r in Step 2 is needed to blind $\sum_k \delta_k t_k$. Since each U_k has trace 0, $\{U_k\}_{k=0}^{m-1}$ compresses to $m(m-1)$ bits ($m-1$ field elements); together with $A \in \mathbb{F}$, the Prover sends m field elements in total.

Theorem 4.3. *The protocol in Figure 2 satisfies the following properties:*

- ***Completeness:*** *If $Y = \sum_{i,j} R_{i,j} d_j(X_i)$ and the Prover is honest, the Verifier always accepts.*
- ***Soundness:*** *Fix any committed $\vec{X}_{pad} \in \mathbb{F}^{M+m+1}$ (where b need not equal r^2), any $R_{i,j}$, and any Y chosen before $\vec{\delta}$ is sampled. If $Y \neq \sum_{i,j} R_{i,j} d_j(X_i)$, the Verifier accepts with probability at most 2^{-m} over $\vec{\delta} \leftarrow \mathbb{F}^m$ (or at most $2^{-(m-1)}$ when (8) and (9) are merged via $\eta \leftarrow \mathbb{F}$).*
- ***Zero-Knowledge:*** *For every Verifier strategy, $(\{U_k\}_{k=0}^{m-1}, A)$ is distributed uniformly over $(\ker(Tr))^m \times \mathbb{F}$, independently of $\vec{X}$. Consequently, if the commitment scheme is hiding and its linear-relation check in Step 5 is zero-knowledge (as in randomized Ligero-style commitments), the protocol is zero-knowledge.*

Proof. Completeness: By Lemma 2.2, Lemma 3.3, and Lemma 4.1, $Tr(U_k) = Tr(Z_k) + Tr(t_k^2 + t_k) = 0$. Equation (8) holds by definition of A and t_0. Squaring $A = r + \sum_{k=0}^{m-1} \delta_k t_k$ gives $A^2 = r^2 + \sum_{k=0}^{m-1} \gamma_k t_k^2 = b + \sum_{k=0}^{m-1} \gamma_k t_k^2$. Since $U_k + \rho c_k = \sum_i Q_{i,k} X_i + t_k^2 + t_k$, adding A^2 to $\sum_k \gamma_k (U_k + \rho c_k)$ cancels $\sum_k \gamma_k t_k^2$ and yields the right-hand side of (9).

Soundness: Fix $\vec{X}_{\text{pad}}$, $R_{i,j}$, and $Y \neq \sum_{i,j} R_{i,j} d_j(X_i)$, and set $t_0 = \sum_{k=1}^{m-1} \beta_k t_k$ and $\Delta_b = r^2 + b$. By Lemma 2.2 and Lemma 3.3, there exists $k^* \in [0, m-1]$ such that $Z_k = \sum_i Q_{i,k} X_i + \rho c_k$ has $Tr(Z_{k^*}) = 1$. Let $\{\tilde{U}_k\}$ be the Prover's Step 2 message. If any $Tr(\tilde{U}_k) \neq 0$, Step 3 rejects. Otherwise $Tr(\tilde{U}_{k^*}) = 0$; since $Tr(t_{k^*}^2 + t_{k^*}) = 0$ unconditionally, $E_k = \tilde{U}_k + Z_k + t_k^2 + t_k$ satisfies $Tr(E_{k^*}) = 1$, so $E_{k^*} \neq 0$ and $\vec{E} \neq \vec{0}$ is fixed before $\vec{\delta}$ is sampled.

Public Input: basis β (with $d_0(1) = 1$) and dual basis α for $\mathbb{F} = GF(2^m)$, $\rho \in \mathbb{F}$ with $Tr(\rho) = 1$, and witness count $M \in \mathbb{N}$.
Protocol Steps:

1. **Commitment Phase:** The Prover samples $t_1, \ldots, t_{m-1}, r \leftarrow \mathbb{F}$, sets $b = r^2$, and commits to
$$\vec{X}_{\text{pad}} = (X_0, \ldots, X_{M-1}, t_1, \ldots, t_{m-1}, r, b) \in \mathbb{F}^{M+m+1}.$$
Let $t_0 = \sum_{k=1}^{m-1} \beta_k t_k$ (implicit, not committed).

2. **Claim & Masked Trace Messages:** Public coefficients $R_{i,j} \in \mathbb{F}$ are determined exogenously and the Prover claims $Y \in \mathbb{F}$. Letting $c_k = d_k(Y)$ and $Q_{i,k} = \sum_{j=0}^{m-1} d_k(R_{i,j})\alpha_j$, the Prover computes and sends
$$U_k = \sum_{i=0}^{M-1} Q_{i,k} X_i + \rho c_k + t_k^2 + t_k \quad (\forall k \in [0, m-1]).$$

3. **Trace Check & Batch Challenge:** The Verifier checks $Tr(U_k) = 0$ for all $k \in [0, m-1]$ (rejecting if any fails), samples $\vec{\delta} = (\delta_0, \ldots, \delta_{m-1}) \leftarrow \mathbb{F}^m$, sets $\gamma_k = \delta_k^2$, and sends $\vec{\delta}$ to the Prover.

4. **Linearized Square Opening:** The Prover sends
$$A = r + \sum_{k=0}^{m-1} \delta_k t_k = r + \sum_{k=1}^{m-1} (\delta_k + \delta_0 \beta_k) t_k.$$

5. **Commitment Verification:** Letting $r'_i = \sum_{k=0}^{m-1} \gamma_k Q_{i,k}$, both parties verify two linear relations over $\vec{X}_{\text{pad}}$:
$$A \stackrel{?}{=} r + \sum_{k=1}^{m-1} (\delta_k + \delta_0 \beta_k) t_k, \tag{8}$$
$$\sum_{k=0}^{m-1} \gamma_k (U_k + \rho c_k) + A^2 \stackrel{?}{=} \sum_{i=0}^{M-1} r'_i X_i + \sum_{k=1}^{m-1} (\gamma_k + \gamma_0 \beta_k) t_k + b. \tag{9}$$
(To merge (8) and (9) into one linear commitment query, the Verifier samples $\eta \leftarrow \mathbb{F}$ after receiving A and checks (9) $+ \eta \cdot$ (8).)

Figure 2: The Zero-Knowledge Trace Packing Protocol.

Let $\tilde{A}$ be the Prover's Step 4 message. If $\tilde{A} \neq r + \sum_{k=0}^{m-1} \delta_k t_k$, then (8) fails (or (9) $+ \eta \cdot$ (8) fails with probability $1 - 2^{-m}$ over η). If $\tilde{A} = r + \sum_{k=0}^{m-1} \delta_k t_k$, then $\tilde{A}^2 + b = \Delta_b + \sum_{k=0}^{m-1} \gamma_k t_k^2$, and substituting $\tilde{U}_k + \rho c_k = E_k + \sum_i Q_{i,k} X_i + t_k^2 + t_k$ into (9) reduces (9) to

$$\sum_{k=0}^{m-1} \gamma_k E_k = \Delta_b.$$

Because $\vec{E} \neq \vec{0}$ and Δ_b are fixed before $\vec{\delta} \leftarrow \mathbb{F}^m$ is chosen, Lemma 2.3 bounds the probability that this holds by 2^{-m} (and a union bound gives $2^{-(m-1)}$ when merging via η).

Zero-Knowledge: For fixed $(\vec{X}, R_{i,j}, Y)$, $\vec{Z} = (Z_0, \ldots, Z_{m-1}) \in (\ker(Tr))^m$ is fixed. Since $(t_1, \ldots, t_{m-1}) \leftarrow \mathbb{F}^{m-1}$ is uniform, Lemma 4.2 implies that $\vec{U} = \vec{Z} + \Phi(t_1, \ldots, t_{m-1})$ is uniform on $(\ker(Tr))^m$. Conditioned on $(\vec{X}, \vec{U})$ and any Verifier choice of $\vec{\delta}, r \leftarrow \mathbb{F}$ remains uniform and independent, so $A = r + \sum_k \delta_k t_k$ is uniform on $\mathbb{F}$. The simulator $\mathcal{S}$ samples $U_0, \ldots, U_{m-1} \leftarrow \ker(Tr)$ uniformly, receives $\vec{\delta}$, samples $A \leftarrow \mathbb{F}$ uniformly, and invokes the commitment scheme's zero-knowledge simulator on (8)–(9). □

Remark 4.4 (Batching L field claims with a single square)**.** *Across L field evaluation claims ($N = Lm$ bit-claims), the Prover commits to $L(m-1)$ pad seeds (leaving one seed implicit per block of m via Lemma 4.2) together with a* single *shared pair $(r, b = r^2)$—a total of $L(m-1)+2$ blinding elements—and sends a single scalar $A = r + \sum_{k=0}^{N-1} \delta_k t_k$.*

Remark 4.5 (Extension to general field extensions)**.** *Using the relative trace $Tr : \mathbb{F} \to GF(q)$ (with $Z_k = \sum_i Q_{i,k} X_i - \rho c_k$ and $d_0(1) \neq 0$ in place of $d_0(1) = 1$), the protocol of Figure 2 extends to packing m elements of any finite field $GF(q)$ into $\mathbb{F} = GF(q^m)$ with only two changes: (i) pad (7) as $U_k = Z_k + (t_k - t_k^q)$, which parameterizes* $\ker(Tr)$ *by additive Hilbert's Theorem 90, and (ii) replace squares with q^{th} powers by setting $b = r^q$, $\gamma_k = \delta_k^q$, and $A^2 \mapsto A^q$ in (9). Because q is a power of* $\mathrm{char}(\mathbb{F})$*, $(u+v)^q = u^q + v^q$ still holds without cross terms, and since A^q is evaluated publicly by the Verifier, the commitment scheme still checks only linear relations.*

4.1 Use by sumcheck

In a sumcheck-based argument, the Prover commits to $\vec{X}_{\text{pad}}$ first; the Prover and Verifier then run a sumcheck ending with a claim $\tilde{x}(\vec{r}) = Y$ on the multilinear extension $\tilde{x}$ of the witness bits at a random point $\vec{r}$, fixing $R_{i,j} = \chi_{i,j}(\vec{r})$. Because the pads $t_k^2 + t_k$ in Figure 2 are self-enforcing, the Prover sends $\{U_k\}$ alongside the final sumcheck message, and in the non-interactive setting $\vec{\delta}$ (and η) are derived via Fiat–Shamir.

In a zero-knowledge sumcheck [Fs26], $Y = \tilde{x}(\vec{r})$ cannot be revealed in the clear; instead, the sumcheck outputs a masked claim $Y' = Y + p_Y$, where $p_Y \in \mathbb{F}$ is a committed blinding scalar (or a public $\mathbb{F}$-linear combination of committed sumcheck pads). Since $p_Y = \sum_{j=0}^{m-1} \beta_j d_j(p_Y)$, the masked identity $Y' = \sum_{i<M,j} R_{i,j} d_j(X_i) + \sum_{j=0}^{m-1} \beta_j d_j(p_Y)$ can be verified directly by Figure 2 with no extra messages or rounds: treat $X_M := p_Y$

as one additional packed element with $R_{M,j} = \beta_j$, so $Q_{M,k} = \alpha_k$, $c_k = d_k(Y')$, and (9) simply adds $+\left(\sum_{k=0}^{m-1} \gamma_k \alpha_k\right) p_Y$ to the right-hand side. When $p_Y \leftarrow \mathbb{F}$ is independent of $(\vec{X}, t_1, \ldots, t_{m-1}, r)$, Y' is uniform on $\mathbb{F}$ and Theorem 4.3 preserves soundness 2^{-m} and zero-knowledge.

5 Comparison with Related Work

As mentioned in the introduction, Diamond and Posen [DP25, DP26] formulate bit unpacking as a *ring-switching* reduction over the tensor algebra $\mathbb{F} \otimes_{GF(2)} \mathbb{F}$, an approach also adopted in non-zero-knowledge systems such as Flock [BRW26]. Our trace-based protocol differs from ring-switching in several respects.

First, our approach avoids extra sumcheck rounds and achieves tighter soundness. Ring-switching runs either a $\log_2 m$-round inner sumcheck [DP25] or a multilinear tensor contraction over $\mathbb{F}^{\log_2 m}$ (losing a $\log_2 m$ factor in soundness: $(\log_2 m) \cdot 2^{-m}$) followed by a $\log_2 M$-round closing sumcheck over the packed witness blocks [DP26, Construction 3.1]. Viewing $\mathbb{F}$ directly over $GF(2)$ replaces both steps over a linear commitment scheme with a single linear batching challenge (Lemma 2.3) having exact soundness error $1/|\mathbb{F}| = 2^{-m}$.

Second, our approach handles arbitrary coefficients without tensor structure. Ring-switching (as well as direct vector-space bit-transposition) relies on the evaluation weights factoring as a pure tensor $R_{i,j} = C_i D_j$ so that the j-dimension can be stripped off and transposed against β. Our trace test (Lemma 3.3) requires no structure on $R_{i,j}$, directly batching $N = Lm$ arbitrary bit-linear claims across multiple evaluation points, while still recovering multiplication-free $O(Mm)$ XOR work via (6) whenever $R_{i,j} = C_i D_j$ does factor.

Dual-basis packing [Blo26] In a concurrent note, Bloemen [Blo26] independently develops a non-zero-knowledge "dual packing" reduction for inner-product commitment schemes over general field extensions $\mathbb{E}/\mathbb{F}$. His result characterizes blockwise packings via dual bases under an $\mathbb{F}$-linear output map $H : \mathbb{E} \to \mathbb{F}$. Abstractly, his one-round protocol [Blo26, Protocol 4] is similar to our non-zero-knowledge protocol in Figure 1 when specialized to the binary field, up to small differences (e.g. shifting Z_k by ρc_k so that $Tr(Z_k) = 0$). From that shared starting point, the two works take complementary directions: Bloemen [Blo26] focuses on non-zero-knowledge verifier succinctness by choosing constant-term projection $H = d_0$; his goal is to use this method inside polylogarithmic IPCSs such as WHIR. In contrast, our main goal is zero-knowledge, and in this case, choosing $H = Tr$ is crucial: unlike d_0 (where $d_0(t^2 + t) \neq 0$ in general), $\ker(Tr)$ is directly parameterized by $\wp(t) = t^2 + t$ (Lemma 4.1), enabling self-enforcing masks with zero quadratic constraints.

Zero-knowledge ring-switching wrappers [Bin26,Koc26,SG26] The main advantage of our approach over ring-switching is its path to zero-knowledge. Diamond and Posen [DP26]

leave zero-knowledge ring-switching as future work: blinding the Prover's m partial evaluations with a one-time pad is awkward because transposing the $m \times m$ bit matrix of the pad is $GF(2)$-linear rather than $\mathbb{F}$-linear, so a linear commitment to the pad vector cannot simultaneously open both the untransposed claim and the transposed batch check. Following the outer-verifier blinding paradigm of [Fs26], Binius64 [Bin26], Kochergin [Koc26], and GordieHowe [SG26] certify the pad correction inside an outer proof system using extra constraints, each in a different way.

Binius64 [Bin26] commits directly to the $m \times m$ unpacked coordinate matrix of the pad and uses m^2 quadratic constraints to ensure that each entry is a *bit*. At $m = 128$, this costs 16,384 bit-check constraints ($3m^2 - 2m = 48{,}896$ total multiplications including basis-constant scalings) and 48,768 auxiliary variables [Koc26, §8].

Both Kochergin [Koc26] and GordieHowe [SG26, §5.5, App. F] avoid committing to pad bits by expressing coordinate extraction as a linearized polynomial via Frobenius "twists" (which [SG26] attributes to Hashcaster [Sou24]): blinding the m twisted partial sums with committed field masks $R_0, \ldots, R_{m-1} \in \mathbb{F}$ reduces the pad correction to certifying the Frobenius sum $\sum_{k=0}^{m-1} R_k^{2^k}$. Kochergin [Koc26] merges weights across L claims and evaluates $\sum_{k=0}^{m-1} R_k^{2^k}$ via Horner's rule, $R_0 + (R_1 + (R_2 + \cdots)^2)^2$, which still requires $m - 1$ quadratic squaring constraints, $2(m-1)$ affine constraints, and $3(m-1)$ auxiliary variables on top of the $Lm + 1$ one-time-pad elements (510 hidden blinding variables in total at $L = 1, m = 128$, excluding the incoming sumcheck pad).

GordieHowe [SG26], in contrast, does not apply Horner's rule and instead commits a separate length-k squaring chain for each mask R_k. Packing d bits therefore requires $1 + 2 + \cdots + (d-1) = d(d-1)/2$ quadratic squaring constraints (8,128 constraints and 8,256 mask/chain elements if applied directly at $d = m = 128$). To bound this quadratic blowup, GordieHowe packs only $d = 16$ bits at a time via squaring chains into $GF(2^{16})$ (120 quadratic constraints and 136 elements), and bridges $GF(2^{16}) \subset GF(2^{128})$ via multi-step packing trees that commit 48–64 additional mask coordinates in Ligero-specific Reed–Solomon subfield rows checked by column-wise subfield membership tests.

Our approach instead reduces every bit claim to a *trace-zero* check. The identity $z = t(t+1)$ provides unconditional trace-zero masks, and squared challenges $\gamma_k = \delta_k^2$ collapse any number of masked claims into a single committed square $(r, b = r^2)$. Our protocol (Figure 2) thus requires *zero* quadratic constraints or subfield tests in the commitment scheme, packs all $m = 128$ bits in one step, commits to only $L(m-1) + 2$ blinding elements (129 at $L = 1, m = 128$), and avoids extra sumcheck rounds and inverse Frobenius evaluations. Even if one applied Step 2's Freshman's-Dream batching to discharge Kochergin's $m - 1$ squaring gates linearly, [Koc26] would still require $O(m)$ extra Horner wires, Frobenius inversions, tensor-structured weights, and ring-switching's closing sumcheck. Table 1 summarizes the concrete overhead at $m = 128$.

Scheme ($m = 128, L = 1$)	**Quad. checks**	**Pads / aux**	**Batch err. / SC**	**Any** $R_{i,j}$
Binius64 ZK [Bin26]	16,384	48,768+	$\geq 7 \cdot 2^{-128}$ + SC	No
Kochergin [Koc26]	127	510	$\geq 7 \cdot 2^{-128}$ + SC	No
GordieHowe [SG26] (extrap.)	8,128	8,256+	$\geq 14 \cdot 2^{-128}$ + SC†	No†
This work (Fig. 2)	**0**	**129**	$\mathbf{2^{-128}}$ **(no SC)**	**Yes**

Table 1: Zero-knowledge bit-unpacking overhead for one field evaluation claim over $GF(2^{128})$ (counts for [Bin26,Koc26] from [Koc26, Table 1, §7], excluding the incoming sumcheck pad p_Y, which adds $+L$ to all schemes). Across L batched claims, Kochergin [Koc26] uses 127 quadratic constraints and $128L + 382$ pad/auxiliary elements, whereas Figure 2 uses 0 quadratic constraints and $127L + 2$ pad elements. †Extrapolated to $m = 128$ from GordieHowe's squaring-chain bit gadget [SG26, Table 1, Protocol F.3] ($m(m-1)/2$ quadratic checks and $m(m+1)/2$ elements, plus $2\log_2 m$ pads and error from its masked bit sumcheck [SG26, Protocol D.7]); in practice, [SG26, Cor. F.7] caps squaring chains at $d = 16$ (120 checks, 136 elements) and bridges $GF(2^{16}) \subset GF(2^{128})$ via 48–64 Ligero-specific $GF(2^{16})$-subfield entries.

Relation to Diamond's Zero-Knowledge Binary BaseFold [Dia25] The tiny-field commitment architecture of [DP26] has two layers: (i) *ring-switching*, which reduces bit-level claims over $GF(2)$ to evaluation claims on a packed multilinear over $\mathbb{F} = GF(2^m)$, and (ii) *Binary BaseFold*, a polynomial commitment scheme for multilinears already defined over $\mathbb{F}$. Diamond [Dia25] addresses a different problem from ours: it adds zero-knowledge only to Layer (ii) (incurring a $\approx 4\times$ prover overhead from FRI domain-doubling and folding a full-length random masking codeword), while explicitly leaving Layer (i)—zero-knowledge bit-unpacking—as open future work [Dia25, §1, §5]. Our protocol in Figure 2 supplies that missing Layer (i) reduction with purely additive overhead ($m + 1$ blinding elements, or $L(m-1) + 2$ across L claims), and can be instantiated over any hiding linear commitment scheme (e.g., Ligero) or composed directly on top of [Dia25].

References

[Bin26] Binius Developers. Binius64 blueprint and reference implementation. https://github.com/binius-zk/binius64, 2026.

[Blo26] Remco Bloemen. Dual packing. Math & Engineering Blog, https://xn--2-umb.com/26/dual-packing/, September 2026.

[BRW26] Benedikt Bünz, Ron D. Rothblum, and William Wang. Flock: Fast proving for batch boolean computations. Cryptology ePrint Archive, Paper 2026/1329, 2026. https://eprint.iacr.org/2026/1329.

[Dia25] Benjamin E. Diamond. Zero-knowledge polynomial commitment in binary fields. Cryptology ePrint Archive, Paper 2025/1015, 2025. https://eprint.iacr.org/2025/1015.

[DP25] Benjamin E. Diamond and Jim Posen. Succinct arguments over towers of binary fields. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, pages 93–122, Cham, 2025. Springer Nature Switzerland.

[DP26] Benjamin E. Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 3–32, Cham, 2026. Springer Nature Switzerland.

[Fs26] Matteo Frigo and abhi shelat. Anonymous credentials from ECDSA. *IACR Communications in Cryptology*, 3(1), 2026.

[Koc26] Andrei Kochergin. Zero-knowledge ring-switching without the transpose. Oumuamua Labs Blog, https://oumuamua.dev/blog/zk-ring-switching, September 2026.

[SG26] Chad Sharp and Paul Grubbs. Faster and smaller anonymous credentials from standardized signatures. Cryptology ePrint Archive, Paper 2026/2615, 2026. https://eprint.iacr.org/2026/2615.

[Sou24] Lev Soukhanov. Hashcaster. https://hackmd.io/@levs57/SJ4fuZMD0, September 2024.