FROST-GKR: Direct Degree-Seven Poseidon2b Relations over Binary Tower Fields

Unified Hypercubes and Shifted Tables for Sequential Hashing Arguments

Ignotus Nemo
ignotus.nemo@proton.me

July 2026

Abstract

The product-chain GKR implementation of a degree-seven S-box introduces several dependent sumchecks for every permutation. For the 59-permutation Poseidon2b computation considered here, this gives 472 product sumchecks and 4,248 sumcheck rounds before terminal batching. In this implementation, the cost of maintaining the Fiat–Shamir transcript then exceeds the cost of evaluating Poseidon2b itself.

We introduce Frost-GKR (Frobenius Reduction Over Shifted Tables), an arithmetization of sequential Poseidon2b computations over $\mathbb{F} = \mathrm{GF}(2^{128})$. The permutation instances are placed in one Boolean hypercube, and the active S-box relation is imposed directly as $s_{\mathrm{out}} = s_{\mathrm{in}}^7$. Over a binary field, this power is evaluated using two multiplications and two Frobenius squarings. A change of summation variable expresses transitions between adjacent rounds in terms of shifted tables. The Poseidon2b constraints are checked by one degree-nine sumcheck, and a degree-two sumcheck reduces the shifted claims to openings of the three original witness columns.

For 59 width-four permutations, Frost-GKR uses 30 constraint rounds and 75 sumcheck rounds including terminal batching, compared with 4,248 and 4,263, respectively, for the product-chain implementation. On a 12-thread Intel Core i7-1365U, median prover time over 20 release-mode samples decreases from 1,495.629 ms to 142.475 ms (10.50×), and median protocol-verifier time decreases from 929.575 ms to 62.501 ms (14.87×). Raw algebraic proof size decreases from 287,712 bytes to 5,568 bytes (51.67×).

1 Introduction

Transparent proof systems over binary fields align proof arithmetic with native bit operations and carry-less multiplication. Binius [5] demonstrated the practical value of tower fields for succinct arguments, while the GKR protocol [1, 2] remains a natural tool for proving regular, repetitive computations.

A bottleneck appears when a sequential computation contains a high-degree nonlinear layer and the implementation expands that layer into a multiplicative chain. Consider the degree-seven S-box $x \mapsto x^7$ used by the Poseidon2b permutation in ParanO(1)d. The product-chain baseline evaluated in this work introduces intermediate columns such as

$$x_2 = x \cdot x, \qquad x_4 = x_2 \cdot x_2, \qquad x_3 = x_2 \cdot x, \qquad x_7 = x_4 \cdot x_3,$$

then proves each relation through a separate product sumcheck. In the baseline evaluated in this work, each permutation requires eight nine-variable product sumchecks. Across 59 permutations this is 472 sumchecks and $59 \cdot 8 \cdot 9 = 4{,}248$ constraint rounds.

High-degree custom gates in GKR are not new; Belling, Soleimanian, and Bégassat, for example, describe custom GKR gates for MiMC and Poseidon [3]. The problem addressed here is more specific: retaining the degree-seven relation while flattening many sequential permutations into one hypercube and binding the resulting shifted views to the original committed columns.

Frost-GKR proves every permutation instance over one shared Boolean hypercube. Sequential round transitions are expressed through shifted multilinear tables, then tied back to the original committed columns by one additional batched product sumcheck. The resulting Frost-GKR protocol uses two constraint sumcheck invocations. Their depth is the logarithm of the padded batch table, rather than growing linearly with the number of permutations.

1.1 Contributions

1. **Unified degree-seven relation.** We pack the nonlinear and linear constraints of 59 sequential permutations into one relation without materializing the baseline's degree-two multiplication chain. Frobenius squaring evaluates x^7 using two field multiplications and two linear squarings.

2. **Shifted-table reduction.** We place all sequential round constraints on the same hypercube by a change of variable and materialized shifted tables. A degree-two shift gadget reduces the resulting shifted claims to the original committed witness columns.

3. **Flat-basis prover.** Witness tables remain in the polynomial basis modulo $X^{128}+X^7+X^2+X+1$ throughout the sumcheck hot path, permitting field multiplication through carry-less multiplication instructions. Only transcript-facing values cross back to the tower basis.

4. **Reproducible comparison.** We define a like-for-like comparison against the preserved legacy degree-two GKR implementation for the same 59-permutation statement, field, transcript, compiler profile, and machine.

2 Preliminaries

2.1 Binary tower fields and Frobenius squaring

Let $\mathbb{F} = \mathrm{GF}\big(2^{128}\big)$, represented either in a recursive tower basis or in an isomorphic flat polynomial basis. Addition is bitwise XOR. In characteristic two, subtraction and addition coincide.

Lemma 1 (Frobenius endomorphism)**.** *For all $a, b \in \mathbb{F}$,*

$$(a+b)^2 = a^2 + b^2.$$

Consequently, squaring is $\mathbb{F}_2$-linear and

$$x^7 = x \cdot x^2 \cdot x^4 = (x \cdot x^2) \cdot x^4$$

can be evaluated with two multiplications and two squarings.

The linearity of squaring does not lower the algebraic degree of the sumcheck relation: $(a + tb)^7$ remains degree seven in the round variable t. It does, however, reduce the cost of evaluating or constructing that degree-seven polynomial over a binary field.

2.2 Multilinear extensions and sumcheck

For a table $f : \{0,1\}^n \to \mathbb{F}$, let $\widetilde{f} : \mathbb{F}^n \to \mathbb{F}$ denote its multilinear extension. The equality polynomial is

$$\mathrm{eq}(r, x) = \prod_{i=0}^{n-1} \big((1-r_i)(1-x_i) + r_i x_i\big).$$

The sumcheck protocol reduces a claim

$$v = \sum_{x \in \{0,1\}^n} g(x)$$

to one evaluation of g at a random point. If g has individual degree at most d, each round transmits a degree-d univariate polynomial, represented by $d+1$ field elements before protocol-specific compression.

2.3 Poseidon2b

We use the width-four binary-field Poseidon2 instantiation implemented by ParanO(1)d, denoted Poseidon2b. It has 66 nonlinear rounds: eight full rounds and 58 partial rounds. Full rounds apply $x \mapsto x^7$ to every lane; partial rounds apply it only to lane zero. Each round also applies round constants and the corresponding linear layer. Poseidon2 is described by Grassi, Khovratovich, and Schofnegger [6]; the binary-field instantiation and domain separation used here are implementation parameters of ParanO(1)d.

3 The FROST-GKR construction

3.1 Unified hypercube

We prove 59 permutation instances in one 15-variable hypercube with index

$$x = (\mathsf{slot} : 6 \parallel \mathsf{round} : 7 \parallel \mathsf{lane} : 2).$$

Unused slot and round coordinates are zero padded. The witness consists of three multilinear columns:

$$s_{\mathrm{in}}(x), \qquad s_{\mathrm{out}}(x), \qquad \mathsf{state}(x).$$

The public selector $\sigma(x)$ is one exactly where the S-box is active. Public tables also encode the round constants, round mask, and linear-layer coefficients.

For every live cell, the following identities hold:

$$C_1(x) = \sigma(x)\big(s_{\mathrm{out}}(x) + s_{\mathrm{in}}(x)^7\big) + (1+\sigma(x))\big(s_{\mathrm{out}}(x) + s_{\mathrm{in}}(x)\big), \tag{1}$$

$$C_1'(x) = \sigma(x)\big(s_{\mathrm{in}}(x) + \mathsf{state}(x) + \mathsf{RC}(x)\big). \tag{2}$$

Let $y = \mathsf{inc}(x)$ increment only the round coordinate, and let dec be its inverse. The transition identity for output lane e is

$$C_2(y) = \mathsf{state}(y)_e + \sum_{j=0}^{3} M_{\mathsf{kind}(\mathsf{dec}(y))}[e,j]\,\pi(\mathsf{dec}(y) \mid \mathsf{lane} = j), \tag{3}$$

$$\pi(x) = \sigma(x) s_{\mathrm{out}}(x) + (1+\sigma(x))\mathsf{state}(x). \tag{4}$$

On a full round, $\pi = s_{\mathrm{out}}$ in every lane. On a partial round, lane zero uses s_{out} while the remaining lanes pass through the previous state.

3.2 Main degree-nine sumcheck

The prover materializes each table referenced at $\mathsf{dec}(y)$ as a multilinear table indexed by y. For a transcript challenge $\rho \in \mathbb{F}^{15}$ and a public live-cell mask μ, define

$$U(y) = \mathrm{eq}(\rho, \mathsf{dec}(y))\,\mu(\mathsf{dec}(y)),$$

where the right-hand side denotes the multilinear table obtained by evaluating the expression on the Boolean cube.

With independent transcript challenges $\beta, \gamma \in \mathbb{F}$, the main sumcheck proves

$$\sum_{y \in \{0,1\}^{15}} U(y)\big(C_1(\mathsf{dec}(y)) + \beta C_1'(\mathsf{dec}(y)) + \gamma C_2(y)\big) = 0.$$

Every table factor is multilinear in the sumcheck variable. The largest term is $U \cdot \sigma \cdot s_{\mathrm{in}}^7$, of individual degree nine. The main proof therefore consists of 15 degree-nine round polynomials. At its terminal point r', it emits direct and shifted claims on the three witness columns.

3.3 Degree-two shift gadget

The shifted claims must be bound to the original committed columns. For example,

$$s_{\mathrm{in,dec}}(r') = \sum_{x \in \{0,1\}^{15}} w_{\mathrm{dec}}(x)\, s_{\mathrm{in}}(x),$$

where w_{dec} is the multilinear extension of the Boolean table

$$x \longmapsto \mathrm{eq}(r', \mathsf{inc}(x)).$$

The distinction is important: $\mathsf{inc}(x)$ is evaluated while building the public Boolean table. The sumcheck receives its multilinear extension; it does not symbolically substitute a carry polynomial into eq. Thus each shift term is a product of one public multilinear weight and one witness multilinear column, giving individual degree two.

One challenge δ combines all shifted whole-column and lane-projected claims. A single 15-round degree-two sumcheck reduces them to three openings

$$s_{\mathrm{in}}(r''), \qquad s_{\mathrm{out}}(r''), \qquad \mathsf{state}(r'')$$

of the original columns. For the 15-variable table used in this work, the main and shift relations require $15 + 15 = 30$ constraint sumcheck rounds. More generally, the protocol uses two constraint sumcheck invocations and their round count grows with the logarithm of the padded table size.

4 Implementation

4.1 Flat-basis accumulation

The prover maps the witness and public MLE tables to the flat polynomial basis once. Table folds are linear, and field products in the hot loop use carry-less multiplication followed by reduction in the flat field basis. Only round-polynomial coefficients and terminal claims are converted back to the tower basis for transcript absorption.

4.2 Coefficient-form round construction

A table entry folded in one sumcheck variable has the affine form $a + tb$. Rather than evaluating the complete integrand independently at ten points, the prover constructs its degree-nine monomial coefficients directly and XOR-accumulates them across the remaining Boolean cube. In particular, it constructs

$$(a + tb)^7 = \sum_{i=0}^{7} q_i t^i$$

in the flat basis, then combines it with the affine selector and equality factors. This removes repeated table scans and interpolation from the main round loop.

4.3 Proof accounting

The legacy baseline runs eight nine-variable degree-three product sumchecks per permutation. Before terminal batching, this is

$$59 \cdot 8 \cdot 9 = 4{,}248$$

constraint rounds. Frost-GKR runs 15 degree-nine main rounds and 15 degree-two shift rounds. The legacy path then runs one 15-round terminal batch sumcheck, for 4,263 total sumcheck rounds. Frost-GKR runs three 15-round terminal batch sumchecks, for 75 total rounds.

Using the implementation's raw-field-element accounting, the legacy proof has 287,712 bytes and the Frost-GKR proof has 5,568 bytes. These figures include the terminal batch sumchecks but exclude serialization framing and the final polynomial-commitment openings.

5 Security analysis

5.1 Algebraic soundness

Assume the witness polynomials are binding before the corresponding challenges are sampled. A conservative interactive algebraic ledger for this implementation is as follows. The random-point zero test contributes at most $15/|\mathbb{F}|$; combining C_1, C_1', C_2 with β, γ contributes at most $1/|\mathbb{F}|$; the degree-nine main sumcheck contributes $135/|\mathbb{F}|$. Combining the eleven shifted claims with powers of δ contributes at most $10/|\mathbb{F}|$, and its degree-two sumcheck contributes $30/|\mathbb{F}|$. The three terminal claim batches contribute at most $3/|\mathbb{F}|$ for their random linear combinations and $90/|\mathbb{F}|$ for their degree-two sumchecks. By a union bound, the algebraic reduction error is therefore at most

$$\frac{15 + 1 + 135 + 10 + 30 + 3 + 90}{2^{128}} = \frac{284}{2^{128}} < 2^{-119}.$$

This is not an end-to-end security number. The deployed argument must also include the polynomial commitment scheme's binding, proximity, query, and hash assumptions, together with the security loss of its Fiat–Shamir transform.

5.2 Fiat–Shamir transform

The implementation derives $\rho, \beta, \gamma, \delta$, all sumcheck challenges, terminal batching challenges, and PCS challenges from one domain-separated transcript. Reducing 4,248 constraint rounds to 30 reduces transcript work and proof bytes. It is a performance and implementation benefit, not by itself a security theorem for the complete non-interactive protocol.

6 Performance evaluation

6.1 Benchmark setup

We compare the product-chain baseline and Frost-GKR on the same sequence of 59 Poseidon2b permutations, over the same field and Fiat–Shamir transcript. Both implementations and the benchmark harness are available at revision `8e514ff`.

Measurements were taken from native release builds on an Intel Core i7-1365U. After three warm-up runs, we collected 20 interleaved samples for each implementation and report the median and p95. Prover time includes witness construction, MLE materialization, and sumcheck generation; verifier time ends at the terminal MLE claims.

Metric	Legacy product chain	Frost-GKR	Ratio
Prover median	1,495.629 ms	142.475 ms	10.50×
Prover p95	1,552.296 ms	152.656 ms	10.17×
Verifier median	929.575 ms	62.501 ms	14.87×
Verifier p95	950.801 ms	63.971 ms	14.86×
Algebraic proof bytes	287,712	5,568	51.67×
Constraint rounds	4,248	30	141.60×
All sumcheck rounds	4,263	75	56.84×

Table 1: Like-for-like 59-permutation comparison. Verifier timing ends at the terminal MLE claims. Proof size counts raw algebraic field elements before PCS openings and serialization.

6.2 Results

7 Related work

GKR and time-optimal interactive proofs. The GKR protocol delegates layered circuit evaluation with a verifier whose work is sublinear in the circuit size [1]. Thaler developed practical and time-optimal refinements [2]. Belling, Soleimanian, and Bégassat explicitly formulate custom low-degree GKR gates and apply them to MiMC and Poseidon [3]. Thus the generic use of a degree-seven gate is prior art. Frost-GKR contributes the unified sequential relation, shifted-table binding, and its binary-field implementation.

Power arithmetizations. Soukhanov's power circuits replace generic multiplication wiring with affine maps and powering, with particular savings for sparse GKR wiring predicates [4]. That work and Frost-GKR share the observation that the choice of arithmetization matters. Frost-GKR instead targets a fixed iterated Poseidon2b relation, keeps its degree-seven gate, and solves the round-index shift through materialized tables and a terminal degree-two reduction.

Binary-field proof systems. Diamond and Posen's Binius work constructs succinct arguments over towers of binary fields and develops binary-field adaptations of standard polynomial IOP components [5]. Frost-GKR targets a narrower problem: collapsing many repeated high-degree Poseidon2b relations and their sequential round transitions into two sumchecks.

Poseidon2. Poseidon2 introduces efficient linear layers for arithmetization-oriented hashing [6]. The present work concerns a binary-field instantiation with a degree-seven S-box and exploits characteristic-two Frobenius arithmetic in its proof relation.

Polynomial commitments. The terminal MLE claims can be discharged by a transparent multilinear PCS. BaseFold [7] is one relevant field-agnostic construction. The Frost-GKR relation is modular with respect to that final commitment layer, but end-to-end soundness, proof size, and performance must include the selected PCS.

8 Conclusion

Frost-GKR replaces the evaluated multiplicative-chain baseline for repeated Poseidon2b permutations with one degree-nine relation over a unified hypercube and one degree-two shift relation. The Frost-GKR implementation reduces 472 constraint sumchecks to two, reduces constraint rounds from 4,248 to 30, and reduces raw algebraic proof bytes by 51.67×. Reproducible measurements show a 10.50× median prover speedup and a 14.87× median protocol-verifier speedup. Binary-field Frobenius squaring and flat-basis carry-less multiplication make

the direct degree-seven relation practical; the shifted-table reduction is what makes it compose with the sequential round structure.

Acknowledgements

This work originated in the proof system developed for the ParanO(1)d statechain project, `https://noid.network`. Reproducible source is available at `https://github.com/ignotusnemo/paranoid`.

References

[1] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM*, 62(4):27:1–27:64, 2015. Preliminary version in STOC 2008.

[2] J. Thaler. Time-optimal interactive proofs for circuit evaluation. In *Advances in Cryptology—CRYPTO 2013*, pages 71–89, 2013.

[3] A. Belling, A. Soleimanian, and O. Bégassat. Recursion over public-coin interactive proof systems; faster hash verification. Cryptology ePrint Archive, Report 2022/1072, 2022. `https://eprint.iacr.org/2022/1072`.

[4] L. Soukhanov. Power circuits: A new arithmetization for GKR-styled sumcheck. Cryptology ePrint Archive, Report 2023/1611, 2023. `https://eprint.iacr.org/2023/1611`.

[5] B. E. Diamond and J. Posen. Succinct arguments over towers of binary fields. Cryptology ePrint Archive, Report 2023/1784, 2023. `https://eprint.iacr.org/2023/1784`.

[6] L. Grassi, D. Khovratovich, and M. Schofnegger. Poseidon2: A faster version of the Poseidon hash function. Cryptology ePrint Archive, Report 2023/323, 2023. `https://eprint.iacr.org/2023/323`.

[7] H. Zeilberger, B. Chen, and B. Fisch. BaseFold: Efficient field-agnostic polynomial commitment schemes from foldable codes. Cryptology ePrint Archive, Report 2023/1705, 2023. `https://eprint.iacr.org/2023/1705`.