# Succinct Lattice Inner Product

ccertain *

October 21, 2026

## 1 Sketch

The LaBRADOR argument of knowledge [3] introduces the concept of dot product constraints of the form:

$$\tilde{\cdot} \in \mathbb{Z}_q/(X^N+1)$$

$$\sum_{i,j}^{r} \tilde{a}_{ij}\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_j\rangle + \sum_{i}^{r}\langle \tilde{\boldsymbol{\phi}}_i, \tilde{\mathbf{w}}_i\rangle = \tilde{b}$$

We propose a modified protocol operating entirely in the scalar field $\mathbb{Z}_q$ with $q = 2^{61}-1$ (Mersenne61). The protocol has three layers:

1. An SIS (Ajtai) commitment to the rows of a binary witness matrix, $\mathbf{t}_i = \mathbf{A}\mathbf{w}_i$. This is the only source of binding.

2. A sumcheck [2] over the column index of the witness. The constraints on the witness are aggregated with uniform challenges into a single sum over the columns, which the sumcheck reduces to the evaluation of every witness row at one random point. The aggregation challenges are in $\mathbb{Z}_q$ and repeated; the round challenges come from the cubic extension $\mathbb{F} = \mathbb{F}_{q^3}$ so that the rounds resist grinding under Fiat-Shamir (Section 5.4). The round polynomials and the evaluations are sent in the clear and are bound to the witness by the final step.

3. An amortized opening with short integer challenges, $\mathbf{z} = \sum_i c_i\mathbf{w}_i$. The verifier checks $\mathbf{z}$ against the SIS commitment and against the row evaluations.

The opening challenges are binary. A binary challenge space is small, so $k$ independent random combinations are opened, but every difference of two challenges is $\pm 1$ on a single row and the extracted openings are exact short preimages (Section 5.2). Larger short challenge sets would need fewer openings but carry an extraction slack, the property LaBRADOR obtains from short invertible ring challenges, and are not used.

Earlier versions of this document sent Gram matrices of the witness, $r^2$ elements per aggregation challenge, in place of the sumcheck. The sumcheck sends $O(\log n + r)$ elements per aggregation repetition, so the witness can be tall and narrow, which shrinks the openings.

Zero knowledge is not a goal of this protocol: the openings reveal the witness.

*ccertain@tutanota.com

# 2 Protocol

## 2.1 Notation and Parameters

Bold indicates vector, bold uppercase indicates matrix. Vectors are column vectors. $\mathbf{a} \circ \mathbf{b}$ is the element-wise product. Elements of $\mathbb{Z}_q$ are compared and normed through their centered representatives in $(-\frac{q}{2}, \frac{q}{2})$, and $|\mathbf{x}|_\infty$ is the largest centered absolute value of an entry.

- $q = 2^{61} - 1$, $e := \lceil \log_2 q \rceil = 61$.
- Prover knows a witness $\mathbf{W} \in \{0,1\}^{r \times n} \subset \mathbb{Z}_q^{r \times n}$ with rows $\mathbf{w}_i$.
- $\mathbf{A} \in \mathbb{Z}_q^{\kappa \times n}$ is a public uniformly random matrix expanded from a seed. $\kappa$ is chosen for SIS hardness (Section 5.5).
- $\mathcal{C} = \{0,1\}$ is the opening challenge set.
- $k$ is the number of opened random combinations and $k_\gamma$ the number of independent aggregation repetitions.
- $\mathbb{F} = \mathbb{Z}_q[x]/(x^3 - 5)$ is the cubic extension field of $q^3 \approx 2^{183}$ elements ($q \equiv 1 \bmod 3$ and 5 is the smallest non-cube, so $x^3 - 5$ is irreducible). An element is written as three coefficients. A product of an element of $\mathbb{F}$ by one of $\mathbb{Z}_q$ costs 3 multiplications, of two elements of $\mathbb{F}$ about 11.
- $\ell := \lceil \log_2 n \rceil$. A vector $\mathbf{u} \in \mathbb{Z}_q^n$ is read as a function on the boolean hypercube $\{0,1\}^\ell$ through the bits of the column index, low bit first, with the columns $n, \ldots, 2^\ell - 1$ equal to zero. Its multilinear extension is

$$\tilde{u}(\mathbf{x}) = \sum_{l \in \{0,1\}^\ell} eq(\mathbf{x}, l)\, u_l, \qquad eq(\mathbf{x}, l) = \prod_j^\ell (x_j l_j + (1 - x_j)(1 - l_j))$$

  and $eq(\mathbf{x}, \cdot) \in \mathbb{Z}_q^{2^\ell}$ is the vector of $eq(\mathbf{x}, l)$ over the column indices. $\tilde{u}(\boldsymbol{\rho}) = \langle \mathbf{u}, eq(\boldsymbol{\rho}, \cdot) \rangle$ costs $n$ multiplications given the $eq$ vector, which costs $2^\ell$.

**Sumcheck.** For a polynomial $P$ of degree at most $d$ in each of $\ell$ variables, the sumcheck protocol [2] reduces the claim $\sum_{l \in \{0,1\}^\ell} P(l) = S$ to one evaluation of $P$. In round $j \in [\ell]$ the prover sends the univariate polynomial $s_j(t) = \sum_{l' \in \{0,1\}^{\ell - j - 1}} P(r_0, \ldots, r_{j-1}, t, l')$ as $d + 1$ coefficients in $\mathbb{F}$, the verifier checks $s_j(0) + s_j(1) = S_j$ against the current claim $S_0 = S$, samples $r_j \xleftarrow{\$} \mathbb{F}$ and sets $S_{j+1} = s_j(r_j)$. After the last round the verifier checks $S_\ell = P(\mathbf{r})$. A false claim passes with probability at most $d\ell/q^3$. When $P$ is built from multilinear extensions of vectors, $P(\mathbf{r})$ is a function of the vectors' evaluations $\tilde{u}(\mathbf{r}) \in \mathbb{F}$, which the prover sends and the verifier checks against the commitment. The prover's tables are over $\mathbb{Z}_q$ in the first round and over $\mathbb{F}$ after the first fold.

## 2.2 Commitment

Prover commits to each row of the witness with the SIS matrix:

$$\mathbf{A}\mathbf{w}_i \to \mathbf{t}_i \in \mathbf{T} \in \mathbb{Z}_q^{r\times\kappa}$$

Prover sends $\mathbf{T} \to$ verifier.

## 2.3 Aggregation

The relation to argue is given as a polynomial $F$ of degree at most 2 in the row values, typically a random combination of quadratic constraints with coefficients sampled here, and a claimed value $S$ of its sum over the columns. Verifier samples $k_\gamma$ independent repetitions, $m \in [k_\gamma]$, of

$$\boldsymbol{\rho}^{(m)} \xleftarrow{\$} \mathbb{Z}_q^\ell, \qquad \beta^{(m)} \xleftarrow{\$} \mathbb{Z}_q$$

together with the coefficients of $F^{(m)}$, and sends them to prover. $\boldsymbol{\rho}^{(m)}$ weights the columns through $eq(\boldsymbol{\rho}^{(m)}, \cdot)$, the role the uniform vector $\boldsymbol{\gamma}$ played in earlier versions, and $\beta^{(m)}$ separates the rows of the binary witness check. Both parties agree on the claim

$$\sum_{l\in\{0,1\}^\ell} eq(\boldsymbol{\rho}, l)\left[F(\mathbf{w}_1(l),\ldots,\mathbf{w}_r(l)) + \sum_i^r \beta^i\,\mathbf{w}_i(l)\,(1-\mathbf{w}_i(l))\right] = S \tag{1}$$

writing $\mathbf{w}_i(l)$ for the entry of row $i$ in column $l$. The second term is the binary witness check: it vanishes identically when $\mathbf{W}$ is binary, and the random $\beta$ and $\boldsymbol{\rho}$ make a non-binary row detectable (Section 5.4). The prover sends the claimed values $S^{(m)}$ (when they are not implied), then verifier samples a batching challenge $\mu \xleftarrow{\$} \mathbb{F}$ and the $k_\gamma$ claims are combined into one:

$$\sum_{l\in\{0,1\}^\ell}\sum_m^{k_\gamma} \mu^m\, eq(\boldsymbol{\rho}^{(m)}, l)\left[F^{(m)}(\mathbf{w}_1(l),\ldots,\mathbf{w}_r(l)) + \sum_i^r \beta^{(m)i}\,\mathbf{w}_i(l)\,(1-\mathbf{w}_i(l))\right] = \sum_m^{k_\gamma}\mu^m S^{(m)} \tag{2}$$

The repetitions share the witness rows and differ in their $eq$ vectors, so the batched polynomial has the same degree as a single one and the prover folds one copy of the witness.

## 2.4 Sumcheck

Prover and verifier run the sumcheck of 2: the polynomial has degree 3 in each variable, so round $j$ consists of one polynomial of 4 coefficients in $\mathbb{F}$ sent by prover, then one challenge $r_j \in \mathbb{F}$ sent by verifier. After the last round prover sends the row evaluations at the challenge point:

$$\tilde{\mathbf{w}}_i(\mathbf{r}) \to v_i \in \mathbf{v} \in \mathbb{F}^r$$

## 2.5 Opening

Verifier samples a binary challenge matrix $\mathbf{C} \xleftarrow{\$} \{0,1\}^{r\times k}$ with columns $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(k)}$ and sends it to prover. (Non-interactively, the entries are the low $e-1$ bits of $\lceil rk/(e-1)\rceil$ transcript oracle elements, a fixed shape that a circuit can reproduce; each bit is biased by at most $2^{e-1}/q$.) Prover opens $k$ random combinations of the witness rows:

$$\mathbf{W}^\top \mathbf{C} \to \mathbf{Z} \in \mathbb{Z}_q^{n\times k}, \qquad \mathbf{z}^{(j)} = \sum_i^r c_i^{(j)} \mathbf{w}_i$$

Prover sends $\mathbf{Z} \to$ verifier. As integers $0 \le \mathbf{z}^{(j)} \le r$.

## 2.6 Verification

Verifier runs the sumcheck checks: in round $j$, $s_j(0) + s_j(1) \stackrel{?}{=} S_j$, and after the last round

$$S_\ell \stackrel{?}{=} \sum_m^{k_\gamma} \mu^m \, eq(\boldsymbol{\rho}^{(m)}, \mathbf{r}) \left[ F^{(m)}(v_1, \ldots, v_r) + \sum_i^r \beta^{(m)i} \, v_i \, (1 - v_i) \right] \quad \text{sumcheck final check} \tag{3}$$

where $eq(\boldsymbol{\rho}^{(m)}, \mathbf{r})$ costs $\ell$ multiplications in $\mathbb{F}$.
For each opening $j \in [k]$, writing $\mathbf{z} = \mathbf{z}^{(j)}$ and $\mathbf{c} = \mathbf{c}^{(j)}$, verifier checks:

$$\mathbf{A}\mathbf{z} \stackrel{?}{=} \mathbf{T}^\top \mathbf{c} = \sum_i^r c_i \mathbf{t}_i \quad \text{opening of the SIS commitment} \tag{4}$$

$$|\mathbf{z}|_\infty \stackrel{?}{\le} r \quad \text{norm check} \tag{5}$$

$$\langle \mathbf{z}, eq(\mathbf{r}, \cdot) \rangle \stackrel{?}{=} \sum_i^r c_i \cdot v_i \quad \text{evaluation check} \tag{6}$$

The norm check is implemented as the range $z_l + r \in [0, 2^b)$ with $b = \lceil \log_2(2r+1) \rceil$, which a circuit checks with a bit decomposition; it allows positive entries up to about $2r$, which the norm bounds below absorb. Equation 4 with 5 binds every opening to the commitment $\mathbf{T}$, so $\mathbf{z} = \sum_i c_i \bar{\mathbf{w}}_i$ for the extracted rows $\bar{\mathbf{w}}_i$ (Section 5.2). The evaluations $\mathbf{v}$ are fixed before $\mathbf{C}$ is sampled, and the left hand side of 6 is $\sum_i c_i \tilde{\bar{\mathbf{w}}}_i(\mathbf{r})$, so 6 is a degree one check in $\mathbf{c}$ that forces $v_i = \tilde{\bar{\mathbf{w}}}_i(\mathbf{r})$ (Section 5.3). Given correct evaluations, the sumcheck forces 2 to hold for the extracted witness, which forces every repetition's claim 1 unless $\mu$ is a root of a nonzero polynomial of degree $k_\gamma - 1$, and 1 is a random combination over ($\boldsymbol{\rho}, \beta$, coefficients of $F$) of every constraint and every row's binarity, zero for a bad witness with probability about $(\ell + r)/q$ per repetition (Section 5.4).

The verifier's work on the openings is $k\kappa n$ multiplications for 4 and $3kn$ for 6 (products of $\mathbb{Z}_q$ by $\mathbb{F}$), plus one vector $eq(\mathbf{r}, \cdot)$ of which the first $n$ entries are used. The earlier pairwise Gram checks, $k^2 n$ multiplications, are gone.

# 3 R1CS

Given a rank one constraint system (R1CS) defined as the matrices $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbb{Z}_q^{c \times w}$, and which is satisfied by any vector $\mathbf{x} \in \mathbb{Z}_q^w$ such that:

$$\mathcal{A}\mathbf{x} \cdot \mathcal{B}\mathbf{x} - \mathcal{C}\mathbf{x} = \mathbf{0} \in \mathbb{Z}_q^c$$

All indices in this section start at 0. Define a bit decomposition function which accepts a vector and decomposes each element to bits vertically as a matrix, and a vector chunking function which splits a vector into $l$ chunks of equal length, padding the final chunk with zeroes:

$$\mathit{to_bits} : \mathbb{Z}_q^m \to \{0,1\}^{e \times m}, \qquad \mathit{chunks} : (l, \mathbb{Z}_q^m) \to \mathbb{Z}_q^{l \times \lceil \frac{m}{l} \rceil}$$

We construct the argument from two witnesses. All quadratically related variables, the values $\mathcal{A}\mathbf{x}, \mathcal{B}\mathbf{x}, \mathcal{C}\mathbf{x}$, are stored in the quadratic witness $\mathbf{W}^{(s)}$. The variables themselves, $\mathbf{x}$, are stored in the linear witness $\mathbf{W}^{(l)}$. Each is split into column chunks stacked vertically, $c^{(s)}$ for the quadratic and $c^{(l)}$ for the linear witness, which trades the width of the openings against the height of the commitments and evaluations (Section 4). The witness dimensions are:

$$r^{(s)} = 3ec^{(s)}, \quad n^{(s)} = \left\lceil \frac{c}{c^{(s)}} \right\rceil, \qquad r^{(l)} = ec^{(l)}, \quad n^{(l)} = \left\lceil \frac{w}{c^{(l)}} \right\rceil, \qquad \ell_s = \lceil \log_2 n^{(s)} \rceil, \quad \ell_l = \lceil \log_2 n^{(l)} \rceil$$

The program is padded with zero rows to height $c^{(s)}n^{(s)}$ and with zero columns to width $c^{(l)}n^{(l)}$, and $\mathbf{x}$ is padded with zeroes to length $c^{(l)}n^{(l)}$. Padded rows are trivially satisfied, and padded entries of $\mathbf{x}$ do not affect any constraint.

**Witnesses.** Write $\mathbf{a}_a$, $\mathbf{b}_a$, $\mathbf{c}_a$ for chunk $a$ of $\mathcal{A}\mathbf{x}$, $\mathcal{B}\mathbf{x}$, $\mathcal{C}\mathbf{x}$ under $\mathit{chunks}(c^{(s)}, \cdot)$, and $\mathbf{x}_b$ for chunk $b$ of $\mathit{chunks}(c^{(l)}, \mathbf{x})$. The witnesses are stacks of blocks:

$$\mathbf{W}^{(s)} = \begin{bmatrix} \mathbf{W}_0^{(s)} \\ \vdots \\ \mathbf{W}_{c^{(s)}-1}^{(s)} \end{bmatrix} \in \{0,1\}^{r^{(s)} \times n^{(s)}}, \quad \mathbf{W}_a^{(s)} = \begin{bmatrix} \mathit{to_bits}(\mathbf{a}_a) \\ \mathit{to_bits}(\mathbf{b}_a) \\ \mathit{to_bits}(\mathbf{c}_a) \end{bmatrix}$$

$$\mathbf{W}^{(l)} = \begin{bmatrix} \mathit{to_bits}(\mathbf{x}_0) \\ \vdots \\ \mathit{to_bits}(\mathbf{x}_{c^{(l)}-1}) \end{bmatrix} \in \{0,1\}^{r^{(l)} \times n^{(l)}}$$

Block $a$ of the quadratic witness occupies rows $o = 3ea$ to $o + 3e - 1$ and block $b$ of the linear witness occupies rows $eb$ to $eb + e - 1$. The recomposed values of a block are the multilinear combinations of its rows

$$\bar{\mathbf{a}}_a = \sum_i^e 2^i \mathbf{w}_{o+i}^{(s)}, \quad \bar{\mathbf{b}}_a = \sum_i^e 2^i \mathbf{w}_{o+e+i}^{(s)}, \quad \bar{\mathbf{c}}_a = \sum_i^e 2^i \mathbf{w}_{o+2e+i}^{(s)}, \quad \bar{\mathbf{x}}_b = \sum_i^e 2^i \mathbf{w}_{eb+i}^{(l)}$$

which for an honest prover are $\mathbf{a}_a$, $\mathbf{b}_a$, $\mathbf{c}_a$, $\mathbf{x}_b$. Their multilinear extensions are the same combinations of the rows' extensions. The linear witness only enters through linear relations, so it needs to be short (for the SIS opening) but not binary.

**Commitment.** Prover sends

$$\mathbf{T}^{(s)} = \mathbf{W}^{(s)}\mathbf{A}^{(s)\top}, \quad \mathbf{T}^{(l)} = \mathbf{W}^{(l)}\mathbf{A}^{(l)\top}$$

with $\mathbf{A}^{(s)} \in \mathbb{Z}_q^{\kappa \times n^{(s)}}$ and $\mathbf{A}^{(l)} \in \mathbb{Z}_q^{\kappa \times n^{(l)}}$.

**Aggregation.** For $m \in [k_\gamma]$ verifier samples

$$\boldsymbol{\rho}^{(m)} \xleftarrow{\$} \mathbb{Z}_q^{\ell_s}, \quad \boldsymbol{\psi}^{(m)} \xleftarrow{\$} \mathbb{Z}_q^{c^{(s)}}, \quad \delta^{(m)}, \beta^{(m)}, \lambda^{(m)} \xleftarrow{\$} \mathbb{Z}_q$$

and both parties compute $\boldsymbol{\gamma}^{(m)} = eq(\boldsymbol{\rho}^{(m)}, \cdot) \in \mathbb{Z}_q^{n^{(s)}}$, the column weights. The weight can only depend on the column, which is shared by every block of $\mathbf{W}^{(s)}$; $\boldsymbol{\psi}^{(m)}$ separates the blocks and $\delta^{(m)}$ separates $\mathcal{A}, \mathcal{B}, \mathcal{C}$. Each R1CS row $an^{(s)} + l$ gets the coefficient

$$\tau^{(m)}_{an^{(s)}+l} = \psi^{(m)}_a \gamma^{(m)}_l, \qquad \boldsymbol{\tau}^{(m)} \in \mathbb{Z}_q^{c^{(s)}n^{(s)}}$$

and both parties compute the combined linear constraint vector, split into chunks matching the linear witness:

$$\boldsymbol{\tau}^{(m)\top}\mathcal{A} + \delta^{(m)} \cdot \boldsymbol{\tau}^{(m)\top}\mathcal{B} + \delta^{(m)2} \cdot \boldsymbol{\tau}^{(m)\top}\mathcal{C} \to \boldsymbol{\phi}^{(m)\top} \in \mathbb{Z}_q^{c^{(l)}n^{(l)}}, \qquad chunks(c^{(l)}, \boldsymbol{\phi}^{(m)}) \to \boldsymbol{\Phi}^{(m)} \in \mathbb{Z}_q^{c^{(l)} \times n^{(l)}} \tag{7}$$

Prover sends the linear claim of every repetition, the value of the combined linear constraints on the witness:

$$\langle \boldsymbol{\phi}^{(m)}, \bar{\mathbf{x}} \rangle = \sum_b^{c^{(l)}} \langle \boldsymbol{\phi}^{(m)}_b, \bar{\mathbf{x}}_b \rangle \to S^{(m)} \in \mathbb{Z}_q$$

and verifier samples the batching challenge $\mu \xleftarrow{\$} \mathbb{F}$.

**Sumchecks.** Two sumchecks are run, each batching every repetition, with the prover's rows as the tables. The r1cs sumcheck, over the columns of the quadratic witness, has degree 3 per variable and claims

$$\begin{aligned} \sum_{l \in \{0,1\}^{\ell_s}} \sum_m^{k_\gamma} \mu^m \, eq(\boldsymbol{\rho}^{(m)}, l) \Big[ \sum_a^{c^{(s)}} \psi^{(m)}_a \left( \bar{\mathbf{a}}_a(l)\bar{\mathbf{b}}_a(l) - \bar{\mathbf{c}}_a(l) \right) + \sum_i^{r^{(s)}} \beta^{(m)i} \, \mathbf{w}^{(s)}_i(l) \left( 1 - \mathbf{w}^{(s)}_i(l) \right) \\ + \lambda^{(m)} \sum_a^{c^{(s)}} \psi^{(m)}_a \left( \bar{\mathbf{a}}_a(l) + \delta^{(m)}\bar{\mathbf{b}}_a(l) + \delta^{(m)2}\bar{\mathbf{c}}_a(l) \right) \Big] = \sum_m^{k_\gamma} \mu^m \lambda^{(m)} S^{(m)} \end{aligned} \tag{8}$$

The first term is the quadratic constraints of every block, the second the binary witness check, the third the left hand side of the linear consistency with the linear witness, which equals $\langle \boldsymbol{\phi}^{(m)}, \bar{\mathbf{x}} \rangle$ for a consistent witness. The witness dependent products $\bar{\mathbf{a}}_a \bar{\mathbf{b}}_a$ and $\mathbf{w}_i(1 - \mathbf{w}_i)$ are shared by the repetitions, so the batching costs the prover little. The linear sumcheck, over the columns of the linear witness, has degree 2 per variable and claims

$$\sum_{l\in\{0,1\}^{\ell_l}} \sum_{m}^{k_\gamma} \mu^m \sum_{b}^{c^{(l)}} \boldsymbol{\phi}_b^{(m)}(l)\, \bar{\mathbf{x}}_b(l) = \sum_{m}^{k_\gamma} \mu^m S^{(m)} \tag{9}$$

Both sumchecks run in lockstep: transcript round $j$ carries the round polynomial of the r1cs sumcheck (4 coefficients in $\mathbb{F}$, while $j < \ell_s$) then of the linear sumcheck (3 coefficients, while $j < \ell_l$), followed by the challenges $r_j, r'_j \xleftarrow{\$} \mathbb{F}$ in the same order. After the last round prover sends the row evaluations at the two challenge points:

$$\tilde{\mathbf{w}}_i^{(s)}(\mathbf{r}) \to \mathbf{v}^{(s)} \in \mathbb{F}^{r^{(s)}}, \qquad \tilde{\mathbf{w}}_i^{(l)}(\mathbf{r}') \to \mathbf{v}^{(l)} \in \mathbb{F}^{r^{(l)}}$$

**Opening.** Verifier samples $\mathbf{C}^{(s)} \xleftarrow{\$} \{0,1\}^{r^{(s)}\times k}$ and $\mathbf{C}^{(l)} \xleftarrow{\$} \{0,1\}^{r^{(l)}\times k}$ and prover sends

$$\mathbf{Z}^{(s)} = \mathbf{W}^{(s)\top}\mathbf{C}^{(s)} \in \mathbb{Z}_q^{n^{(s)}\times k}, \qquad \mathbf{Z}^{(l)} = \mathbf{W}^{(l)\top}\mathbf{C}^{(l)} \in \mathbb{Z}_q^{n^{(l)}\times k}$$

**Verification.** Verifier runs the round checks of both sumchecks. The final check of the r1cs sumcheck is 3 with the bracket of 8 evaluated at $\mathbf{v}^{(s)}$, with $\bar{\mathbf{a}}_a = \sum_i^e 2^i v_{o+i}^{(s)}$ and so on. The final check of the linear sumcheck is

$$S'_{\ell_l} \stackrel{?}{=} \sum_{m}^{k_\gamma} \mu^m \sum_{b}^{c^{(l)}} \tilde{\boldsymbol{\phi}}_b^{(m)}(\mathbf{r}') \sum_{i}^{e} 2^i v_{eb+i}^{(l)}, \qquad \tilde{\boldsymbol{\phi}}_b^{(m)}(\mathbf{r}') = \langle \boldsymbol{\phi}_b^{(m)}, eq(\mathbf{r}', \cdot)\rangle \tag{10}$$

where the verifier evaluates the public chunks $\boldsymbol{\phi}_b^{(m)}$ itself, $3n^{(l)}$ multiplications per chunk, no more than computing $\boldsymbol{\phi}^{(m)}$ in the first place. For the quadratic witness verifier runs 4, 5 and 6 with $r = r^{(s)}$ and $(\mathbf{A}^{(s)}, \mathbf{T}^{(s)}, \mathbf{Z}^{(s)}, \mathbf{C}^{(s)}, \mathbf{v}^{(s)}, \mathbf{r})$, and for the linear witness with $r = r^{(l)}$ and $(\mathbf{A}^{(l)}, \mathbf{T}^{(l)}, \mathbf{Z}^{(l)}, \mathbf{C}^{(l)}, \mathbf{v}^{(l)}, \mathbf{r}')$.

**Correctness.** With binary $\mathbf{W}^{(s)}$ the binary term of 8 vanishes, the quadratic term of repetition $m$ is $\sum_a \psi_a \sum_l \gamma_l(\bar{a}_{al}\bar{b}_{al} - \bar{c}_{al}) = 0$ for a satisfying witness, and its linear term is $\lambda \sum_{a,l} \tau_{an^{(s)}+l}(\bar{a}_{al} + \delta\bar{b}_{al} + \delta^2\bar{c}_{al}) = \lambda\boldsymbol{\tau}^\top(\mathcal{A} + \delta\mathcal{B} + \delta^2\mathcal{C})\bar{\mathbf{x}} = \lambda\langle\boldsymbol{\phi}, \bar{\mathbf{x}}\rangle = \lambda S^{(m)}$. The sum of repetition $m$ in 9 is $\langle\boldsymbol{\phi}^{(m)}, \bar{\mathbf{x}}\rangle = S^{(m)}$ by definition.

**Soundness.** Section 5: the openings bind $\mathbf{v}^{(s)}, \mathbf{v}^{(l)}$ to the extracted witnesses (error $2\epsilon_1^k$ over both witnesses), given correct evaluations the sumchecks force 8 and 9 for the extracted witness (error $(3\ell_s + 2\ell_l)/q^3$), the batching forces them for every repetition (error $2(k_\gamma - 1)/q^3$), and for a repetition the left hand side of 8 minus $\lambda\langle\boldsymbol{\phi}, \bar{\mathbf{x}}\rangle$ is a nonzero polynomial of degree at most $\ell_s + r^{(s)} + 4$ in the challenges of the repetition whenever the extracted witness is not binary, violates a quadratic constraint, or is inconsistent between the two witnesses. The error per repetition is

$$\epsilon_{agg} = \frac{\ell_s + r^{(s)} + 4}{q}$$

**Proof size.** About

$$(r^{(s)} + r^{(l)})\kappa + k_\gamma + 3\left(4\ell_s + 3\ell_l + r^{(s)} + r^{(l)}\right) + k(n^{(s)} + n^{(l)})$$

elements of $\mathbb{Z}_q$ (three per element of $\mathbb{F}$), linear in the heights and in the widths. Section 4 describes how to choose the dimensions.

## 3.1 As-Is Binary Regions

In many circuits most witness entries and most constraint values are 0 or 1. In SHAKE256 every entry of $\mathbf{x}$ is a bit, and all but one row have $(\mathcal{A}\mathbf{x})_i, (\mathcal{B}\mathbf{x})_i, (\mathcal{C}\mathbf{x})_i \in \{0,1\}$. Bit decomposing such a value costs $e$ witness cells, of which $e-1$ are zero. Instead the prover stores binary values as is, in regions of the witness matrices that are not recomposed with powers of two. Booleanity is not detected from the R1CS (compilers fold it into linear combinations); the prover chooses the regions from the witness values, and soundness does not depend on the choice.

**Masks.** Prover chooses a row mask $\boldsymbol{\mu}^{(s)} \in \{0,1\}^c$ and a witness mask $\boldsymbol{\mu}^{(l)} \in \{0,1\}^w$. The honest choice is $\mu_i^{(s)} = 1$ exactly when $(\mathcal{A}\mathbf{x})_i, (\mathcal{B}\mathbf{x})_i, (\mathcal{C}\mathbf{x})_i \in \{0,1\}$, and $\mu_j^{(l)} = 1$ exactly when $x_j \in \{0,1\}$. The masks are sent with $\mathbf{T}^{(s)}, \mathbf{T}^{(l)}$, before any aggregation challenge, and cost $c+w$ bits.

**Layout.** Rows with $\mu_i^{(s)} = 0$, taken in increasing order, fill the decomposed region of $\mathbf{W}^{(s)}$ exactly as before: the row at position $p$ goes to block $\lfloor p/n^{(s)} \rfloor$, column $p \bmod n^{(s)}$. Rows with $\mu_i^{(s)} = 1$ fill the as-is region in the same way, where as-is block $a$ is the three rows

$$\mathbf{W}^{(s)}_{o'}, \mathbf{W}^{(s)}_{o'+1}, \mathbf{W}^{(s)}_{o'+2} = \bar{\mathbf{a}}_a, \bar{\mathbf{b}}_a, \bar{\mathbf{c}}_a, \qquad o' = 3ec^{(s)} + 3a$$

holding the values themselves. Witness entries are placed the same way in $\mathbf{W}^{(l)}$: entries with $\mu_j^{(l)} = 0$ are bit decomposed into blocks of $e$ rows, entries with $\mu_j^{(l)} = 1$ fill as-is blocks of a single row placed after the decomposed rows. With $u^{(s)} = |\{i : \mu_i^{(s)} = 1\}|$ and $u^{(l)} = |\{j : \mu_j^{(l)} = 1\}|$:

$$c^{(s)} = \left\lceil \frac{c - u^{(s)}}{n^{(s)}} \right\rceil, \quad d^{(s)} = \left\lceil \frac{u^{(s)}}{n^{(s)}} \right\rceil, \quad r^{(s)} = 3ec^{(s)} + 3d^{(s)}$$

$$c^{(l)} = \left\lceil \frac{w - u^{(l)}}{n^{(l)}} \right\rceil, \quad d^{(l)} = \left\lceil \frac{u^{(l)}}{n^{(l)}} \right\rceil, \quad r^{(l)} = ec^{(l)} + d^{(l)}$$

The prover chooses the widths $n^{(s)}, n^{(l)}$ (Section 4 describes how) and sends them with the masks; the block counts follow. Soundness depends on the layout only through the heights, which enter the norm bounds, so the verifier accepts any widths with $r^{(s)}, r^{(l)} \le R_{\max}$, a protocol parameter from which the SIS norm bound is computed (Section 5.5), and does not need to reproduce the prover's choice. Unused cells are zero. With all-zero masks this is the layout above.

**Aggregation.** $\boldsymbol{\psi}^{(m)} \in \mathbb{Z}_q^{c^{(s)}+d^{(s)}}$ has one entry per block of either region, as-is block $a$ using $\psi^{(m)}_{c^{(s)}+a}$. Row $i$ placed in block $\beta$ (of either region), column $l$, gets $\tau_i^{(m)} = \psi_\beta^{(m)}\gamma_l^{(m)}$, and $\boldsymbol{\phi}^{(m)}$ is computed from $\boldsymbol{\tau}^{(m)}$ as in 7. It is split into one chunk $\boldsymbol{\phi}_\beta^{(m)} \in \mathbb{Z}_q^{n^{(l)}}$ per block $\beta$ of $\mathbf{W}^{(l)}$, holding the entries of $\boldsymbol{\phi}^{(m)}$ for the witness entries placed in that block, at their columns, and $S^{(m)}$ sums over all these chunks.

**Sumchecks.** In every repetition the r1cs sumcheck 8 gains the as-is blocks with weight 1 in place of $2^i$: the quadratic term $\sum_a^{d^{(s)}} \psi_{c^{(s)}+a}(\mathbf{w}_{o'}(l)\mathbf{w}_{o'+1}(l) - \mathbf{w}_{o'+2}(l))$ and the linear term $\lambda\sum_a^{d^{(s)}} \psi_{c^{(s)}+a}(\mathbf{w}_{o'}(l) + \delta\mathbf{w}_{o'+1}(l) + \delta^2\mathbf{w}_{o'+2}(l))$, and the binary term already covers every row. The linear sumcheck 9 gains $\sum_b^{d^{(l)}} \boldsymbol{\phi}_{c^{(l)}+b}(l)\, \mathbf{w}^{(l)}_{ec^{(l)}+b}(l)$. The final checks read the as-is rows of $\mathbf{v}$ directly in the same way.

**Soundness.** The masks are fixed before the aggregation challenges are sampled, and the argument of Section 3 goes through for every choice of masks. The binary check forces every row of $\mathbf{W}^{(s)}$ to be binary, so the as-is quadratic term is $\sum_l \gamma_l(\bar{a}_{al}\bar{b}_{al} - \bar{c}_{al})$ on the values themselves. The linear witness needs no binary check: an as-is entry is read directly as $x'_j$, and the extracted $\mathbf{x}'$ satisfies the R1CS whatever its values. A mask choice can therefore only affect completeness: a non-binary as-is value in $\mathbf{W}^{(s)}$ fails the binary check, and a large as-is entry in $\mathbf{W}^{(l)}$ fails the norm check. Small non-binary entries of $\mathbf{W}^{(l)}$ pass the norm check and may also be stored as is. $\epsilon_{agg}$ is unchanged with the heights above.

**Proof size.** About

$$(r^{(s)} + r^{(l)})\kappa + k_\gamma + 3\left(4\ell_s + 3\ell_l + r^{(s)} + r^{(l)}\right) + k(n^{(s)} + n^{(l)}) + \frac{c+w}{e}$$

elements of $\mathbb{Z}_q$, with the heights above. For SHAKE256 the heights shrink by about a factor $e$ for the same widths.

**Witness dependent shape.** The layout depends on the witness values, so the proof shape and verifier work can vary between witnesses of the same circuit. A recursive verifier with a static circuit needs a fixed layout, e.g. masks fixed by the program.

# 4 Witness Layout

The proof size of Section 3.1 is linear in the heights, through the SIS commitments ($\kappa$ per row) and the evaluations (3 per row), and linear in the widths through the openings ($k$ per column). With the cell count of a witness fixed by the program, $rn \approx N$, the size $r(\kappa+3) + kn$ is minimized at $r \approx \sqrt{kN/(\kappa+3)}$, which is tall: with $\kappa = 16$ and $k = 131$ a column costs about seven times a row. For all but small programs this exceeds $R_{\max}$ and the heights are set by $R_{\max}$ instead, the widths following as $N/R_{\max}$. The sumcheck rounds, $4\ell_s + 3\ell_l$ per repetition, are negligible.

The prover chooses the widths $n^{(s)}, n^{(l)}$ by enumerating, for each region of each witness, the widths that give $1, 2, \ldots$ blocks, discarding the combinations with $r^{(s)} > R_{\max}$ or $r^{(l)} > R_{\max}$,

and taking the pair with the smallest estimated size. The estimate is the proof size formula with $k_\gamma$ recomputed for each candidate (it depends on $\ell_s, r^{(s)}$ through $\epsilon_{agg}$) and the masks left out, which are the same for every layout. The search is deterministic in the mask counts so the verifier could reproduce it, but need not: it accepts any widths within $R_{\max}$.

For a random R1CS with decomposed regions only, in Mersenne61 with $R_{\max} = 4096$, $\kappa = 16$:

| $c = w$ | $c^{(s)}, n^{(s)}$ | $c^{(l)}, n^{(l)}$ | $k, k_\gamma$ | elements | Gram version |
|---|---|---|---|---|---|
| $10^4$ | $20,500$ | $35,286$ | $131,3$ | $2.1 \cdot 10^5$ | $1.1 \cdot 10^6$ |
| $5 \cdot 10^5$ | $22,22728$ | $67,7463$ | $131,3$ | $4.1 \cdot 10^6$ | $1.5 \cdot 10^7$ |

The last column is the size the Gram matrix version of this protocol reached with its own optimal layout. The openings are now the dominant term, so a larger $R_{\max}$ (with a correspondingly larger SIS norm bound) is what would shrink the proof further.

# 5 Cryptanalysis

This section analyzes the protocol of Section 2 and its R1CS instantiation in Section 3 over $q = 2^{61} - 1$. Checks on the same challenge do not multiply, the overall soundness error is a sum of the error terms below.

## 5.1 Binding of the Opening

Equation 4 with the norm check 5 is an opening of an SIS commitment. The openings of a column only depend on that column's challenge through the check, so given two accepting transcripts $(\mathbf{C}, \mathbf{Z})$, $(\mathbf{C}', \mathbf{Z}')$ whose challenge matrices satisfy $\mathbf{c}_j - \mathbf{c}'_j = \Delta \mathbf{e}_i$ for some column $j$ and $\Delta \neq 0$, with the other columns arbitrary, the openings of column $j$ satisfy

$$\mathbf{A}(\mathbf{z}_j - \mathbf{z}'_j) = \Delta \cdot \mathbf{t}_i, \qquad |\mathbf{z}_j - \mathbf{z}'_j|_\infty \leq 2r$$

with $\Delta = \pm 1$ for binary challenges, so $\bar{\mathbf{w}}_i = \Delta(\mathbf{z}_j - \mathbf{z}'_j)$ is an exact short opening of $\mathbf{t}_i$. Two such openings of the same $\mathbf{t}_i$ differ by an SIS solution of $\ell_\infty$ norm at most $4r$, so $\bar{\mathbf{w}}_i$ is determined by $\mathbf{t}_i$.

## 5.2 Knowledge Extraction

Fix the prover's randomness and everything up to the opening challenge, so $\mathbf{T}$, the round polynomials and the evaluations $\mathbf{v}^{(m)}$ are fixed. Let $X = \{0,1\}^r$ be the challenge space of one column, $m = |\mathcal{C}| = 2$, $N = |X| = m^r$, and let $S \subseteq X^k$ be the set of challenge matrices the prover answers acceptably, $\epsilon = |S|/N^k$. For a column $j$ and a value $\mathbf{c} \in X$ write $\rho_j(\mathbf{c})$ for the density of $S$ among the matrices with $j$-th column $\mathbf{c}$ (the other columns uniform). A direction-$i$ line is a set $\{\mathbf{c} + t\mathbf{e}_i\}$ of $m$ column values differing only in row $i$.

To extract row $i$ the extractor needs an accepting pair with $\mathbf{c}_j - \mathbf{c}'_j = \Delta \mathbf{e}_i$ for some column $j$, i.e. two accepting matrices whose $j$-th columns lie on the same direction-$i$ line. It samples a base $\mathbf{C} \in S$ and then, for a column $j$ and another value $\mathbf{c}'$ on the direction-$i$ line through $\mathbf{c}_j$, samples matrices with $j$-th column $\mathbf{c}'$ and fresh uniform other columns until one is accepted. This succeeds quickly when some line mate $\mathbf{c}'$ of some column has $\rho_j(\mathbf{c}') \geq \tau$.

**Lemma 1.** *For every row $i$ and $\tau > 0$, the probability over a uniform base $\boldsymbol{C} \in S$ that, for every column $j$, every other value on the direction-$i$ line through $\mathbf{c}_j$ has $\rho_j < \tau$ is at most $(m^{-k} + k\tau)/\epsilon$.*

*Proof.* Call a column value heavy if $\rho_j \geq \tau$ and light otherwise. A base with the stated property has, in every column $j$, either a light $j$-th column or a heavy $j$-th column all of whose line mates are light. Heavy values with only light line mates are at most one per line, so at most $N/m$ per column, and bases of the second kind in every column number at most $(N/m)^k$. Bases of the first kind in some column $j$ carry mass $\sum_{\text{light } \mathbf{c}} \rho_j(\mathbf{c}) N^{k-1} < \tau N^k$, so at most $k\tau N^k$ in total. Divide by $|S| = \epsilon N^k$. □

With $\tau = \epsilon/(4k)$ the probability is at most $m^{-k}/\epsilon + \frac{1}{4}$, so for $\epsilon \geq 4m^{-k}$ at least half of the bases have a line mate of density at least $\epsilon/(4k)$ in some column. Trying the $k(m-1)$ line mates in round robin finds an accepting partner after $O(k^2 m/\epsilon)$ prover invocations in expectation; rows that fail are retried from a fresh base. Every row is extracted independently, so the extractor obtains exact openings $\bar{\mathbf{w}}_i$, with $|\bar{\mathbf{w}}_i|_\infty \leq 2r$, of every row of both witnesses with $O(rk^2/\epsilon)$ invocations, whenever $\epsilon \geq 4\epsilon_1^k$ ($\epsilon_1 = 1/2$). This is the knowledge error of the opening round, of the same order as the evaluation binding terms of Section 5.3.

For an accepting opening $\mathbf{z}^*$ with challenge column $\mathbf{c}^*$, the vector $\mathbf{z}^* - \sum_i c_i^* \bar{\mathbf{w}}_i$ is an integer vector in the kernel of $\mathbf{A}$ of $\ell_\infty$ norm at most $r + 2r^2$, hence zero unless it is an SIS solution of that norm. So $\mathbf{z}^* = \sum_i c_i^* \bar{\mathbf{w}}_i$ in $\mathbb{Z}_q^n$ for every accepting opening, which is what the next subsection assumes. The binary witness check then forces every row of $\bar{\mathbf{W}}^{(s)}$ into $\{0,1\}^n$, and the R1CS relation is over $\mathbb{Z}_q$, so the linear witness $\bar{\mathbf{W}}^{(l)}$ needs no integrality beyond being short.

## 5.3 Binding of the Sumcheck Evaluations

By Section 5.2 every accepting opening equals $\mathbf{z} = \sum_i c_i \bar{\mathbf{w}}_i$ for the extracted rows, unless SIS is broken. The evaluations $\mathbf{v}$ are fixed before $\mathbf{C}$ is sampled, so the evaluation check 6 reads

$$\sum_i^r c_i \left( \tilde{\bar{\mathbf{w}}}_i(\mathbf{r}) - v_i \right) = 0 \in \mathbb{F}$$

Write $\epsilon_1 = \frac{1}{2}$: a uniform column $\mathbf{c} \in \{0,1\}^r$ is orthogonal to a fixed nonzero $\mathbf{u} \in \mathbb{F}^r$ with probability at most $\epsilon_1$, because for a coordinate $i$ with $u_i \neq 0$ and any values of the other coordinates at most one value of $c_i$ satisfies $\langle \mathbf{c}, \mathbf{u} \rangle = 0$. A wrong $\mathbf{v}$ therefore passes one column with probability at most $\epsilon_1$, and all $k$ columns with probability at most $\epsilon_1^k$. The R1CS argument checks two witnesses, so all evaluations are correct except with probability $2\epsilon_1^k$.

The pairwise Gram checks of earlier versions, which needed $(k+1)\epsilon_1^k$ and set $k = 136$, are no longer needed: nothing quadratic in the witness is sent, the quadratic constraints are evaluated inside the sumcheck on the evaluations.

## 5.4 Soundness of the R1CS Argument

Fix the extracted witnesses $\bar{\mathbf{W}}^{(s)}, \bar{\mathbf{W}}^{(l)}$ and suppose the evaluations are correct. The linear sumcheck 9 ends in the verifier's own evaluation of the public $\boldsymbol{\phi}_b^{(m)}$ and the bound $\bar{\mathbf{x}}_b(\mathbf{r}')$, so by the sumcheck soundness its batched claim holds for the extracted witness except with

probability $2\ell_l/q^3$, and then $S^{(m)} = \langle \boldsymbol{\phi}^{(m)}, \bar{\mathbf{x}} \rangle$ for every $m$ unless $\mu$ is a root of a nonzero polynomial of degree $k_\gamma - 1$, probability $(k_\gamma - 1)/q^3$. Likewise the r1cs sumcheck forces the sum of 8 over the extracted witness except with probability $3\ell_s/q^3$, and the batching forces every repetition's sum to equal $\lambda^{(m)} S^{(m)}$ except with probability $(k_\gamma - 1)/q^3$. For a repetition $m$ both together say that

$$F(\boldsymbol{\rho}, \boldsymbol{\psi}, \delta, \beta, \lambda) := \sum_l eq(\boldsymbol{\rho}, l)\big[Q(l) + B(l)\big] + \lambda\Big(\sum_l eq(\boldsymbol{\rho}, l) L(l) - \langle \boldsymbol{\phi}, \bar{\mathbf{x}} \rangle\Big) = 0$$

with $Q, B, L$ the quadratic, binary and linear terms of the bracket of 8 for repetition $m$. $F$ is a polynomial in the $\mathbb{Z}_q$ challenges of the repetition, all sampled after $\mathbf{T}^{(s)}, \mathbf{T}^{(l)}$, i.e. after the extracted witness is fixed. Its degree is at most $\ell_s$ in $\boldsymbol{\rho}$ ($\sum_l eq(\boldsymbol{\rho}, l) u_l$ is the multilinear extension $\tilde{u}(\boldsymbol{\rho})$), 1 in $\boldsymbol{\psi}$, 2 in $\delta$, $r^{(s)} - 1$ in $\beta$ and 1 in $\lambda$, so at most $\ell_s + r^{(s)} + 4$ in total. It is nonzero for a bad witness:

- If row $i$ of $\bar{\mathbf{W}}^{(s)}$ is not binary, the coefficient of $\beta^i \lambda^0$ is the multilinear extension of the nonzero vector $(\bar{w}_{il} - \bar{w}_{il}^2)_l$ at $\boldsymbol{\rho}$, a nonzero polynomial; distinct rows have distinct powers of $\beta$ and cannot cancel.
- If $\bar{\mathbf{W}}^{(s)}$ is binary but block $a$ violates $\bar{\mathbf{a}}_a \circ \bar{\mathbf{b}}_a = \bar{\mathbf{c}}_a$, the coefficient of $\psi_a \lambda^0$ is the multilinear extension of $\bar{\mathbf{a}}_a \circ \bar{\mathbf{b}}_a - \bar{\mathbf{c}}_a$ at $\boldsymbol{\rho}$, nonzero.
- If $\bar{\mathbf{W}}^{(s)}$ is binary and satisfies every block but $(\bar{\mathbf{a}}, \bar{\mathbf{b}}, \bar{\mathbf{c}}) \neq (\mathcal{A}\bar{\mathbf{x}}, \mathcal{B}\bar{\mathbf{x}}, \mathcal{C}\bar{\mathbf{x}})$, the coefficient of $\lambda$ is $\sum_i \tau_i \big[(\bar{a}_i + \delta \bar{b}_i + \delta^2 \bar{c}_i) - ((\mathcal{A} + \delta\mathcal{B} + \delta^2\mathcal{C})\bar{\mathbf{x}})_i\big]$, where the $\tau_i = \psi_{\mathrm{blk}(i)} eq(\boldsymbol{\rho}, l(i))$ of distinct rows are linearly independent polynomials, so it is nonzero when any row's bracket is a nonzero polynomial in $\delta$.

By Schwartz-Zippel $F = 0$ with probability at most

$$\epsilon_{agg} = \frac{\ell_s + r^{(s)} + 4}{q}$$

per repetition, and all $k_\gamma$ independent repetitions pass a bad extracted witness with probability at most $\epsilon_{agg}^{k_\gamma}$. The total soundness error is bounded by

$$\epsilon \leq 6\epsilon_1^k + \epsilon_{agg}^{k_\gamma} + \frac{3\ell_s + 2\ell_l + 2(k_\gamma - 1)}{q^3} + \mathrm{Adv}^{\mathrm{SIS}}_{\kappa, n, q, \beta}$$

with the $4\epsilon_1^k$ of the extractor (Lemma 1), $2\epsilon_1^k$ of the evaluation binding, and the SIS norm bound $\beta \approx 2r^2$ in $\ell_\infty$ from Section 5.2.

**Fiat-Shamir.** After the Fiat-Shamir transform an attacker can grind whole transcripts, so the error of every squeeze has to be below $2^{-128}$ on its own, and errors of different squeezes do not compose. The squeezes and their errors are: the aggregation challenges of all repetitions, $\epsilon_{agg}^{k_\gamma}$ (every repetition sits behind the one squeeze); the batching challenge $\mu$, $(k_\gamma - 1)/q^3$ per sumcheck; each r1cs sumcheck round, $3/q^3$; each linear sumcheck round, $2/q^3$; the opening challenge, $6\epsilon_1^k$. The round challenges are why $\mathbb{F}$ is needed: with $r_j \in \mathbb{Z}_q$ a prover holding a bad witness could, in round $j$, send a polynomial with the correct sum $s_j(0) + s_j(1)$ but otherwise

free coefficients and grind them until $r_j$ is a root of $s_j - s_j^*$, which takes about $q/3 \approx 2^{59}$ squeezes and puts the sumcheck on a true claim from then on. Repeating the sumcheck in $\mathbb{Z}_q$ does not help, since nothing forces the rescues of different repetitions into the same round: they add work instead of multiplying the error. With $r_j \in \mathbb{F}$ the same attack costs $q^3/3 \approx 2^{181}$ squeezes. The aggregation challenges can stay in $\mathbb{Z}_q$ because the repetitions sit behind a single squeeze and the claims of all of them are batched before any round challenge is sampled.

**Row separators.** The powers $\beta^i$ cost one squeeze and $r^{(s)}$ multiplications but put the degree $r^{(s)} - 1$ into $\epsilon_{agg}$. Sampling an independent $\beta_i$ per row would reduce the degree to 1 and $\epsilon_{agg}$ to about $(\ell_s + 6)/q$ at the cost of $r^{(s)}$ squeezes per repetition; for Mersenne61 both give $k_\gamma = 3$.

## 5.5 Parameters for Mersenne61

**Openings.** Every extracted row is an exact opening. The norm check is $|\mathbf{z}|_\infty \leq r$, two openings of the same $\mathbf{t}_i$ differ by at most $2r$ in $\ell_\infty$, and the kernel vector $\mathbf{z}^* - \sum_i c_i^* \bar{\mathbf{w}}_i$ has $\ell_\infty$ norm at most $r + 2r^2$. The number of openings is the smallest $k$ with $6 \cdot 2^{-k} \leq 2^{-128}$,

$$k = 131$$

so about $131(n^{(s)} + n^{(l)})$ field elements of openings.

**Repetitions.** The prover chooses the witness widths, so the heights are only known to satisfy $r \leq R_{\max}$. With $R_{\max} = 2^{12}$ and widths up to $2^{15}$, $\epsilon_{agg} \leq (15 + 4096 + 4)/q \approx 2^{-49}$, so $k_\gamma = 3$ gives $\epsilon_{agg}^3 \approx 2^{-147}$; $k_\gamma = 2$ gives only about $2^{-98}$. The implementation computes $k_\gamma$ from the actual $\ell_s, r^{(s)}$ of the layout, which for Mersenne61 is 3 for every layout within $R_{\max}$. The extension field terms are at most $(3 \cdot 15 + 2 \cdot 15 + 4)/q^3 \approx 2^{-176}$.

**SIS.** The size optimizer drives the heights to $R_{\max}$ for all but small programs (Section 4), so the norm bound is computed at $R_{\max}$: $2R_{\max}^2 = 2^{25}$ in $\ell_\infty$, and with $n \leq 2^{20}$ about $\beta \approx 2^{35}$ in $\ell_2$ (the norm check range adds at most another factor 4). A larger $R_{\max}$ costs SIS security directly. $\kappa$ is not fixed here: it should be set with a lattice estimator for $\mathrm{SIS}_{\kappa,n,q,\beta}$ at the witness width $n$. The proof size is about

$$(r^{(s)} + r^{(l)})\kappa + k_\gamma + 3\left(4\ell_s + 3\ell_l + r^{(s)} + r^{(l)}\right) + k(n^{(s)} + n^{(l)})$$

elements of $\mathbb{Z}_q$.

## 5.6 Open Items

1. **SIS parameters.** $\kappa$ for each witness width, from a lattice estimator. With the heights at $R_{\max}$ the trade between $R_{\max}$, $\kappa$ and the openings should be revisited.

2. **Appendix A** has not been revised to the SIS opening or the sumcheck; it still uses uniform $\boldsymbol{\nu}$ challenges, short sum-of-rows commitments and Gram matrices, which are not binding on their own.

3. **Fiat-Shamir.** Section 5.4 bounds the error of every squeeze; a formal knowledge soundness proof of the non-interactive protocol remains to be written down.

4. **Prover memory.** The prover folds the tables of both witnesses, one copy for all repetitions; after the first fold they live in $\mathbb{F}$ (three elements per cell) at half the width. The first round is computed from the $\mathbb{Z}_q$ tables.

# 6 (Non)-Existence of Infinite and Finite Oracles

Infinite length oracles do not exist by combinatorial bounds on information and computation. Finite oracles can be proven neither to exist nor not to exist, but a lower bound on the probability of non-existence can be realized by taking the number of distinguishable states of reality as the maximum oracle output length. Denote $\lambda^{(M)}$, the distance of a simulated oracle from a theoretical oracle for an output equal in length to the number of distinguishable states of reality $M$. Concretely this may be taken as something like the Lloyd bound $M \approx 2^{399}$ or the holographic storage bound $M \approx 2^{406}$. We apply a conservative bound and take $M = |energy| \cdot |matter|$, defined as the product of the number of distinguishable states of energy and the distinguishable states of matter. Taking each as $2^{256}$ we get $M = 2^{256} \cdot 2^{256} = 2^{512}$.

## 6.1 Probable Finite Oracles

Given $\mathbf{x} \in \mathbb{Z}_q^m$ an output vector of length $m$ from a simulated oracle. Call $\boldsymbol{\chi} \in \mathbb{Z}_q^m$ output from a corresponding theoretical oracle. Call $\lambda_{\ell_1}^{(m)} = \langle \mathbf{1}, |\mathbf{x} - \boldsymbol{\chi}| \rangle$ the linear distance from the simulated oracle to the theoretical oracle for an output of length $m$. A simulated oracle is statistically likely to exist provided $\Pr[\lambda_{\ell_1}^{(m)} = 0] > 1 - \frac{1}{M}$. Oracles with $\Pr[\lambda_{\ell_1}^{(M)} = 0] > 1 - \frac{1}{M}$ are statistically indistinguishable from theoretical oracles for all representable simulations of the oracle.

## 6.2 Infinite Oracles do not Exist

The maximum length of a theoretical oracle is bounded by emergence and dissolution of free energy by big bang and heat death respectively. Oracles exist as a function of information, therefore time is finite relative to oracles as information ceases to exist after heat death, as entropy exists maximally. Time continues to exist, but the arrow of time (a product of information and entropy), ceases to exist. This is analagous to time becoming an algebraic object without total order: points in time become comparable only by equality and relative distance, not by direction. By the existence of entropy, at a point in time forward relative to time itself, total order will emerge in time, at which point entropy is locally minimized and information emerges (a big bang event). Therefore, non-existence of information does not preclude knowledge extraction by quantum observers, which are capable of observing time relative to time itself as a function of their own existence and non-existence.

# A Hypersphere Relations

**Not yet revised to the SIS opening.** The subsections below are the earlier hypersphere constructions. On their own they are not binding: their responses are checked only through a few scalar inner products, so a prover can solve for them after seeing the challenge. Composed with the SIS commitment of Section 2 they become meaningful, because every opening is pinned to the committed witness. In particular the anti-podal point of an opening is an opening of the bitwise complement: for $\bar{c} = \sum_i c_i$,

$$\mathbf{A}(\bar{c} \cdot \mathbf{1} - \mathbf{z}) = \bar{c} \cdot \mathbf{A}\mathbf{1} - \mathbf{T}^\top \mathbf{c}$$

so the verifier can check openings of $\mathbf{W}^! = \mathbf{1} - \mathbf{W}$ without a second commitment.

## A.1 Space Efficient Linear Relations From Short Vectors

Given a witness matrix $\mathbf{W} \in \{0,1\}^{r \times n}$ prover forms two commitments representing the sums of rows and sums of columns respectively:

$$\sum_i^r \mathbf{w}_i \to \mathbf{t} \in \mathbb{Z}^n$$

$$\langle \mathbf{w}_i, \mathbf{1} \rangle \to t'_i \in \mathbf{t}' \in \mathbb{Z}^r$$

Prover sends $\mathbf{t}, \mathbf{t}' \to$ verifier and verifier samples $\boldsymbol{\nu} \xleftarrow{\$} \mathbb{Z}_q^r$ and $\boldsymbol{\nu}' \xleftarrow{\$} \mathbb{Z}_q^n$. Prover computes challenge commitments:

$$\mathbf{W}^\top \boldsymbol{\nu} \to \mathbf{z} \in \mathbb{Z}_q^n$$

$$\mathbf{W}\boldsymbol{\nu}' \to \mathbf{z}' \in \mathbb{Z}_q^r$$

Prover sends $\mathbf{z}, \mathbf{z}' \to$ verifier and verifier checks the following relations:

$$|\mathbf{t}|_\infty \overset{?}{\leq} r \quad |\mathbf{t}'|_\infty \overset{?}{\leq} n$$

$$\langle \mathbf{t}, \mathbf{1} \rangle \overset{?}{=} \langle \mathbf{t}', \mathbf{1} \rangle \quad \text{transposition linear structure check}$$

$$\langle \mathbf{z}', \mathbf{1} \rangle \overset{?}{=} \langle \mathbf{t}, \boldsymbol{\nu}' \rangle \quad \text{challenge linear cross check (1)}$$

$$\langle \mathbf{z}, \mathbf{1} \rangle \overset{?}{=} \langle \mathbf{t}', \boldsymbol{\nu} \rangle \quad \text{challenge linear cross check (2)}$$

$$\langle \mathbf{z}, \boldsymbol{\nu}' \rangle \overset{?}{=} \langle \mathbf{z}', \boldsymbol{\nu} \rangle \quad \text{argument of transposition relation}$$

Iff $\boldsymbol{\nu}, \boldsymbol{\nu}'$ are sampled randomly after $\mathbf{t}, \mathbf{t}'$ commitment, and $r, n \ll q$, the inner products form two SIS instances on lattice bases that are random relative to the preimage. Computing the inner product is an evaluation of a randomly selected member of the universal hash function family $\mathcal{H}_A(\mathbf{x}) = \mathbf{A}\mathbf{x}$. Thus the evaluation of $\langle \mathbf{t}, \boldsymbol{\nu}' \rangle$ and $\langle \mathbf{t}', \boldsymbol{\nu} \rangle$ are each concrete random oracle queries by the leftover hash lemma.

## A.2 Space Efficient Additive Inverse Linear Relations

The previous section describes representing a witness matrix as a relation between the matrix and its transposition. This introduces problems with dimensions and multiplicative combinations. A more efficient approach uses two witness matrices such that $\mathbf{A}+\mathbf{B}=\mathbf{1}$ and $\mathbf{A}\cdot\mathbf{B}=\mathbf{0}$. Given a binary witness matrix $\mathbf{W}\in\{0,1\}^{r\times n}$ define the bitwise not matrix $\mathbf{W}^{!}\in\{0,1\}^{r\times n}$ such that $\mathbf{W}+\mathbf{W}^{!}=\mathbf{1}$ and $\mathbf{W}\cdot\mathbf{W}^{!}=0$.

Prover begins by forming short commitments to the witness matrix:

$$\mathbf{W}^{\top}\cdot\mathbf{1}\rightarrow\mathbf{t}\in|\mathbb{Z}_q^n|_\infty\leq r$$
$$\mathbf{W}\cdot\mathbf{1}\rightarrow\mathbf{g}\in|\mathbb{Z}_q^r|_\infty\leq n$$

Prover sends $\mathbf{t},\mathbf{g}\rightarrow$ verifier and verifier samples $\boldsymbol{\nu}\xleftarrow{\$}\mathbb{Z}_q^r$. Prover forms a vertical challenge commitment:

$$\mathbf{W}^{\top}\cdot\boldsymbol{\nu}\rightarrow\mathbf{z}\in\mathbb{Z}_q^n$$

Prover sends $\mathbf{z}\rightarrow$ verifier and verifier samples $\boldsymbol{\tau}\xleftarrow{\$}\mathbb{Z}_q^n$. Prover forms a horizontal challenge commitment:

$$\mathbf{W}\cdot\boldsymbol{\tau}\rightarrow\boldsymbol{\zeta}\in\mathbb{Z}_q^r$$

Prover sends $\boldsymbol{\zeta}\rightarrow$ verifier and verifier checks the following relations:

$$\langle\mathbf{t},\mathbf{1}\rangle\stackrel{?}{=}\langle\mathbf{g},\mathbf{1}\rangle\quad\text{binary structure check}$$
$$\langle\mathbf{z},\mathbf{1}\rangle\stackrel{?}{=}\langle\mathbf{g},\boldsymbol{\nu}\rangle\quad\text{linear structure check}$$
$$\langle\boldsymbol{\zeta},\mathbf{1}\rangle\stackrel{?}{=}\langle\mathbf{t},\boldsymbol{\tau}\rangle\quad\text{linear structure check}$$
$$\langle\boldsymbol{\zeta},\boldsymbol{\nu}\rangle\stackrel{?}{=}\langle\mathbf{z},\boldsymbol{\tau}\rangle\quad\text{argument of }r\cdot n\text{ independent linear relations}$$

Verifier then interprets $\mathbf{z}$ and $\boldsymbol{\zeta}$ as points on two boolean hyperspheroids with basis vectors $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$ respectively, each of cardinality $r\cdot n$. Verifier selects the antipodal point to get element-wise boolean negation of the witness on the hypersphere:

$$r\cdot\boldsymbol{\nu}-\mathbf{z}\rightarrow\mathbf{z}^{!}\in\mathbb{Z}_q^n\quad\text{anti-podal point on }\boldsymbol{\nu}\text{ boolean hypersphere}$$

$$n\cdot\boldsymbol{\tau}-\boldsymbol{\zeta}\rightarrow\boldsymbol{\zeta}^{!}\in\mathbb{Z}_q^r\quad\text{anti-podal point on }\boldsymbol{\tau}\text{ boolean hypersphere}$$

Verifier can then compute the expected SIS preimages using linear operations:

$$r\cdot\mathbf{1}-\mathbf{t}\rightarrow\mathbf{t}^{!}\in\mathbb{Z}^n\quad\text{bitwise negation in a one-dimensional space}$$

$$n\cdot\mathbf{1}-\mathbf{g}\rightarrow\mathbf{g}^{!}\in\mathbb{Z}^r\quad\text{bitwise negation in a one-dimensional space}$$

Verifier then checks the following relations:

$$\langle\mathbf{z}^{!},\mathbf{1}\rangle\stackrel{?}{=}\langle\mathbf{g}^{!},\boldsymbol{\nu}\rangle\quad\text{bitwise inversion linear check}$$

$$\langle\boldsymbol{\zeta}^{!},\mathbf{1}\rangle\stackrel{?}{=}\langle\mathbf{t}^{!},\boldsymbol{\tau}\rangle\quad\text{bitwise inversion linear check}$$

$$\langle \mathbf{z}^!, \boldsymbol{\tau} \rangle \stackrel{?}{=} \langle \boldsymbol{\zeta}^!, \boldsymbol{\nu} \rangle \quad \text{bitwise inversion random linear relation}$$

The short commitments $\mathbf{t}$ and $\mathbf{g}$ represent a binary matrix as $r + n$ short elements, formed as two vectors: the sums of rows and the sums of columns.

TODO: formally analyze the cardinality of a theoretical solution space given the verifier equations and some entropy. Cardinality of solution space should be equal to cardinality of domain.

### A.3 Geometric Verification of Anti-Podal Points as Second Pre-image

Define an $m$ dimension hypersphere as a function family $\mathcal{H}_{\mathbf{a}}(\mathbf{x} \in \mathbb{Z}_q^m) = \langle \mathbf{a}, \mathbf{x} \rangle$ with prime $q$. Representing relations between hyperspheres and hypercubes allows for geometric information theoretic security. e.g. if I argue knowledge of two points on a circle, I've implied knowledge of one point on a line. If I argue knowledge of two points on a sphere, I've implied knowledge of one point on a plane. If I argue knowledge of one point on a hypercube, I've implied knowledge of two points on any higher dimension hypersphere. We'll use the terms point and pre-image interchangeably, where pre-image additionally implies $|\mathbf{x}|_\infty \ll q$. In this section we'll examine only binary points/pre-images such that internal sums are structure preserving for later linear projection, transformation, and comparison with hypercube commitments, also known as Ajtai commitments: $\mathcal{H}_{\mathbf{A}}(\mathbf{x} \in \mathbb{Z}_q^m) = \mathbf{A}\mathbf{x} \to \mathbf{c} \in \mathbb{Z}_q^n$. Hypercubes, by definition, have the additional structure $n$, which hyperspheres do not.

For any point $\mathbf{x} \in \{0, 1\}^m$ define a theoretical hypersphere with basis vector $\mathbf{a} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^m$. The probability of any $m$ dimension random hypersphere having a binary point in a field with cardinality $q$ is $\frac{2^m}{q^m}$. This becomes negligible for e.g. $m = 10$ and $q \approx 2^{32} : \frac{2^{10}}{2^{320}} \approx 2^{-310}$. In practice we'll use $100 < m < 1000$ for most programs. This is exactly the statistical binding argument from BDLOP [1], but with additional verification of a second pre-image as linear transformations of oracle independent lattices and short commitments/pre-images/points.

That is, with high probability there exists no points on a random hypersphere which can be represented in $\mathbb{Z}_q$ using random linear combinations of $\{0, 1\}$. Knowledge of a single point on such a hypersphere implies knowledge of an anti-podal point, which a verifier can compute and verify. Concretely this should be linear transformations on oracle separated terms. Given a vector $\mathbf{x}$ and a hypersphere with basis $\mathbf{a}$ define the bitwise not/anti-podal point $\mathbf{a} - \mathbf{x} \to \mathbf{x}^! \in \mathbb{Z}_q^m$.

## References


[1] Carsten Baum and Ivan Damgård and Vadim Lyubashevsky and Sabine Oechsner and Chris Peikert, More Efficient Commitments from Structured Lattice Assumptions, Cryptology ePrint Archive, Paper 2016/997, 2016, `https://eprint.iacr.org/2016/997`

[2] Carsten Lund and Lance Fortnow and Howard Karloff and Noam Nisan, Algebraic Methods for Interactive Proof Systems, Journal of the ACM 39(4), 1992

[3] Ward Beullens and Gregor Seiler, LaBRADOR: Compact Proofs for R1CS from Module-SIS, Cryptology ePrint Archive, Paper 2022/1341, 2022, `https://eprint.iacr.org/2022/1341`