Espalier: Succinct Lattice Arguments for Client-Side Proving

Palmette
palmette.xyz

v0.9 preprint · May 2026

Abstract

Post-quantum proof systems are usually discussed in terms of proof size and verifier cost, but for privacy applications the binding constraint is the prover, and the prover runs on the end user's own device. We present Espalier, a succinct argument over the cyclotomic ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ whose soundness rests on Module-SIS and which is designed for client-side proving on commodity mobile hardware. Espalier combines two techniques. *Cordon reduction* controls witness norms by splitting the witness into balanced strata and proving a single aggregated bound over a shared Johnson–Lindenstrauss projection, so that norm slack accumulates additively rather than multiplicatively in the depth of a recursion. *Lateral amortization* batches many instances of a circuit under one shared commitment through a random linear combination over $\mathcal{R}_q$, so that proving k instances costs one full argument plus $O(k)$ commitment work. On an internal prototype, a 2^{20}-constraint shielded-transfer circuit proves in 1.9 s with 640 MB of peak prover memory on an A17-class mobile system-on-chip, yielding a 28 KB proof that verifies on-chain for approximately 310k gas. These figures are from our own prototype and have not been independently reproduced; no implementation is public at the time of writing, and full security proofs are deferred to the extended version.

1 Introduction

The migration of privacy-preserving systems on public ledgers to post-quantum assumptions is usually framed as a question about the verifier. Pairing-based arguments produce proofs of a few hundred bytes that a smart contract can check cheaply, and any post-quantum replacement is expected to give up some of that compactness. This framing understates the problem. In a shielded payment system the proof is generated by the party spending the funds, on whatever device that party happens to own, because generating it anywhere else would disclose the witness. The cost that actually determines whether such a system is usable is therefore the prover's wall-clock time and peak memory on a phone, not the verifier's gas.

That cost is not incidental. Hash-based arguments over FRI achieve transparency and plausible post-quantum soundness, but their provers are memory-hungry: the prover materializes large evaluation domains and performs many rounds of Merkle commitment, and peak working set grows quickly with circuit size. Lattice-based arguments are attractive here for a structural reason. The dominant prover operation is negacyclic polynomial multiplication in $\mathcal{R}_q$, which reduces to the number-theoretic transform, and the NTT is a well-understood target for the SIMD units present on mobile processors. The obstacle is not arithmetic throughput. It is norm growth.

Every lattice argument must prove that a committed witness is short. Composing such arguments, which is what recursion requires, forces each level to prove a bound on a witness that already carries the slack introduced by the level beneath it. In the constructions we are aware of, that slack compounds: each level multiplies the bound by a factor determined by the challenge set and the extraction strategy, so the modulus q must be chosen large enough to accommodate the product across all levels. A larger q means larger ring elements, larger commitments, and a heavier prover, which is precisely the resource we are trying to conserve. Recursion depth is thus purchased directly out of the mobile prover's budget.

1.1 Contributions

This preprint describes Espalier, an argument system that addresses norm growth directly rather than absorbing it into parameters.

- **Cordon reduction** (Section 3), a norm-control technique. Rather than decomposing a witness into digits and proving a separate bound for each, we partition it into balanced strata, which we call cordons, and prove one aggregated bound over a shared random projection. Lemma 2 states that

the resulting slack is additive in the number of composition levels.

- **Lateral amortization** (Section 4), a batching technique for many instances of a single circuit. Instances are folded into one aggregate witness by a random linear combination over $\mathcal{R}_q$ under a shared commitment, so that the marginal cost of an additional instance is one commitment rather than one argument.
- The Espalier argument (Section 5), which composes the two, and a recursive variant with a pairing-free on-chain verifier (Section 6).
- An evaluation on an internal prototype (Section 7), reporting prover time, peak memory, proof size and verification cost for circuit sizes from 2^{14} to 2^{22} constraints.

We state security properties and give proof sketches. Complete proofs, and the knowledge extractor for the recursive composition in particular, are deferred to the extended version; Section 9 is explicit about what has and has not been established.

2 Preliminaries

2.1 Rings and norms

Let d be a power of two and let $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ be the ring of integers of the $2d$-th cyclotomic field. For a modulus q we write $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Throughout we take $d = 64$ and q a prime of approximately 64 bits satisfying $q \equiv 1 \pmod{2d}$, so that X^d+1 splits completely modulo q and multiplication in $\mathcal{R}_q$ is computed by a length-d NTT.

An element $a = \sum_{i<d} a_i X^i \in \mathcal{R}$ is identified with its coefficient vector, and we use the Euclidean norm $\|a\|$ and the infinity norm $\|a\|_\infty$ of that vector. For vectors over $\mathcal{R}$ the norms are taken over the concatenation of coefficients. We write $\|a\|_\infty \le \beta$ to mean every coefficient lies in $[-\beta, \beta]$.

2.2 Module-SIS

Definition 1 (Module-SIS)**.** For parameters κ, m, q, β, the $\mathrm{MSIS}_{\kappa,m,q,\beta}$ problem asks, given $A \leftarrow \mathcal{R}_q^{\kappa \times m}$ chosen uniformly, to find a nonzero $z \in \mathcal{R}_q^m$ with $Az = 0$ and $\|z\| \le \beta$.

All binding properties below reduce to MSIS. We use commitments in the style of Ajtai [1] as adapted to modules by Baum et al. [3]: a commitment key is a uniform $A \in \mathcal{R}_q^{\kappa \times m}$ and the commitment to a short $s \in \mathcal{R}_q^m$ is $\mathrm{com}(s) = As$. Binding holds for openings of norm at most β under $\mathrm{MSIS}_{\kappa,m,q,2\beta}$; hiding is obtained in the usual way by appending a randomness block.

2.3 Challenge space and rejection sampling

Let $\mathcal{C} \subset \mathcal{R}$ be the set of elements with coefficients in $\{-1, 0, 1\}$ and Hamming weight exactly τ. We require that differences of distinct challenges be invertible in $\mathcal{R}_q$, which holds for our splitting parameters when τ is chosen so that $2\tau < d$. We write $\mathcal{C}^* = \{c - c' : c \ne c' \in \mathcal{C}\}$ for the set of challenge differences, and $\gamma = \max_{c \in \mathcal{C}^*} \|c\|_\infty$.

Responses are masked and rejection sampled following Lyubashevsky [2], so that the distribution of a response is independent of the witness. We suppress the rejection-sampling parameters below; they affect constants and expected repetition counts, not the structure of the argument.

2.4 Projected norm checks

We use a projection technique for proving norm bounds in aggregate. Let $\Pi \in \{-1, 0, 1\}^{256 \times N}$ have entries drawn independently with $\Pr[\pm 1] = 1/4$ and $\Pr[0] = 1/2$. For a vector $v \in \mathbb{Z}^N$, the quantity $\|\Pi v\|$ concentrates around $\sqrt{128}\,\|v\|$, and a verifier who learns Πv and checks $\|\Pi v\| \le \sqrt{128}\,\beta \cdot \eta$ for a slack factor η rejects, except with negligible probability, any v with $\|v\|$ substantially larger than β. The technique is standard in recent lattice arguments and is used by LaBRADOR [4]. What is new here is not the projection but how many bounds it is asked to certify at once, which is the subject of the next section.

3 Cordon Reduction

3.1 The problem

Consider proving that a committed witness $s \in \mathcal{R}_q^m$ is short, in a setting where the argument will itself be proved by another instance of the same argument. The conventional route is gadget decomposition. Fix a radix B and write

$$s = \sum_{i=0}^{b-1} B^i s_i, \qquad \|s_i\|_\infty \le B/2, \tag{1}$$

with $b = \lceil \log_B(2\|s\|_\infty) \rceil$. Each stratum s_i is committed and each carries its own norm proof. Two costs follow. First, the witness the prover must handle grows by a factor of b. Second, and more damaging, each norm proof is only sound up to a slack factor. When the argument is composed, the bound proved at level ℓ becomes the bound assumed at level $\ell + 1$, and the slack factors multiply. Writing η for the per-level slack, the effective bound after L levels is $\beta\eta^L$, and q must be chosen so that $\beta\eta^L \ll q$. Since prover cost grows with $\log q$ through the size of ring elements, recursion depth

is paid for in prover time at every level, including the levels that do not need it.

3.2 Cordons

Cordon reduction changes what is proved rather than how it is proved. Instead of b independent bounds, we prove one bound on a single derived vector, and separately prove that the strata are consistent with the witness.

Partition the index set of the decomposition (1) into b strata as before, and call each stratum a *cordon*. Let $\bar{s} = (s_0 \| s_1 \| \cdots \| s_{b-1})$ denote the concatenation of all cordons, a vector of dimension $N = b\,m\,d$ over $\mathbb{Z}$. The prover commits once to $\bar{s}$, and the argument proves the two statements

(C1) *aggregate shortness*: $\|\bar{s}\| \le \beta_{\mathrm{agg}}$, and

(C2) *recomposition*: $\sum_i B^i s_i = s$ for the s committed at the level above,

where (C1) is discharged by a single projected norm check with a shared $\Pi \in \{-1, 0, 1\}^{256 \times N}$, and (C2) by a random linear combination over $\mathcal{R}_q$: the verifier samples $c \leftarrow \mathcal{C}$ and the prover opens $\langle c^{(0)}, \ldots, c^{(b-1)} \rangle$ against the recomposition identity, where $c^{(i)}$ denotes the i-th power of c.

The point is that (C1) is one bound over the whole concatenation. A cheating prover who inflates one cordon must keep the aggregate norm within the single global bound, so inflation in one stratum must be paid for by deflation in another, and the recomposition check (C2) prevents that trade from producing a valid witness. The per-level slack is therefore incurred once, on β_{agg}, rather than once per stratum.

3.3 Slack accumulation

Balancing the cordons matters. If the strata have wildly different norms, the aggregate bound is dominated by the largest and the check says little about the rest. We therefore choose B so that the expected norms of the strata are within a constant factor of one another, which for witnesses arising from R1CS-style constraint systems is achieved by $B \approx 2^{16}$ at our parameters. We refer to this choice as balancing the cordons, and it is what the name is meant to evoke: the strata are trained to a common height so that a single measurement characterizes all of them.

Lemma 2 (Additive slack)**.** *Let η be the slack factor of a single projected norm check at the stated projection dimension, and suppose the cordons are balanced to within a constant factor ρ. Then the bound established after L levels of composition is $\beta \cdot \eta \cdot (1 + L\,\rho\,\gamma\,\sqrt{\tau})$, rather than $\beta \cdot \eta^L$.*

Proof sketch. The slack of a projected check enters multiplicatively only through the projection itself, which is applied once per level to the aggregate. The remaining growth comes from the recomposition check, where the extractor multiplies by a challenge difference and incurs an additive term bounded by $\rho\,\gamma\,\sqrt{\tau}\,\beta$ per level. Summing the additive terms over L levels and applying the single multiplicative factor η gives the stated bound. The full argument requires care in the extractor, which must rewind on the recomposition challenge without re-extracting the projection, and is given in the extended version. □

The practical consequence is the one we care about: q is chosen for the deepest recursion we intend to support, and that choice costs $\log L$ bits rather than $L \log \eta$ bits. At $L = 3$, which is the depth used in Section 7, this is the difference between a 64-bit modulus and one that would not fit in a machine word.

4 Lateral Amortization

Shielded payment systems prove many instances of one circuit. A wallet that has received several notes proves one spend per note; a sequencer aggregating user proofs handles many instances of an identical statement. Treating these independently wastes the structure.

Let k instances of a circuit have witnesses $s^{(1)}, \ldots, s^{(k)} \in \mathcal{R}_q^m$, each satisfying the same constraint system with distinct public inputs. We call these the *laterals* of the batch. The prover commits to each lateral, the verifier samples $c_1, \ldots, c_k \leftarrow \mathcal{C}$, and both parties form the aggregate

$$s^* = \sum_{j=1}^{k} c_j\, s^{(j)}, \qquad \mathrm{com}(s^*) = \sum_{j=1}^{k} c_j\, \mathrm{com}(s^{(j)}), \quad (2)$$

using linearity of the commitment. The argument then runs once, on s^*, with cordon reduction applied to the aggregate.

Lemma 3 (Batch soundness)**.** *If the aggregate argument accepts with probability greater than $k/|\mathcal{C}| + \mathrm{negl}(\lambda)$, then there is an extractor that recovers witnesses for all k instances, each of norm at most $\beta \cdot \gamma \cdot \sqrt{\tau} \cdot k$.*

Proof sketch. Standard k-special-soundness over the challenge vector. Rewinding on k linearly independent challenge tuples yields a Vandermonde system over $\mathcal{R}_q$ whose inverse has entries bounded by challenge differences; invertibility of those differences is guaranteed by the choice of $\mathcal{C}$ in Section 2. The norm growth is the usual cost of inverting that system. Details in the extended version. □

The linear growth of the extracted norm in k bounds useful batch sizes, and in practice we cap k at 64 and recurse over batches beyond that point. The prover cost of an additional lateral is one commitment, which is κm NTT-domain multiplications, and not a further argument.

5 The Espalier Argument

Composing the two techniques gives the argument we call Espalier. We describe it for a rank-1 constraint system with n constraints over $\mathbb{Z}_q$, embedded into $\mathcal{R}_q$ by packing d constraint slots per ring element.

Setup. Sample the commitment key $A \leftarrow \mathcal{R}_q^{\kappa \times m}$. The setup is transparent: A is derived from a public seed by an extendable output function, and no party holds trapdoor information. There is no ceremony and no toxic waste.

Commit. The prover packs the witness into $s \in \mathcal{R}_q^m$, splits it into balanced cordons $\bar{s}$, and sends $\mathrm{com}(\bar{s})$. For a batch of k laterals this is done per lateral and aggregated as in (2).

Constraint check. The verifier samples a challenge and the prover responds with a masked linear combination of the constraint polynomials evaluated at the challenge, rejection sampled so that the response distribution is witness-independent. This is the standard structure and we do not modify it.

Norm check. The verifier samples the projection Π and the prover sends $\Pi\bar{s}$, which the verifier checks against β_{agg} as described in Section 3, together with the recomposition opening.

Output. The transcript is made non-interactive by the Fiat–Shamir transform in the random oracle model.

Proposition 4 (Security, informal)**.** *Espalier is complete; it is knowledge sound under* $\mathrm{MSIS}_{\kappa,m,q,\beta'}$ *for* β' *as given by Lemmas 2 and 3; and it is honest-verifier zero-knowledge given the rejection sampling of Section 2.*

We state Proposition 4 informally on purpose. Completeness and zero-knowledge follow the established pattern and we regard them as settled. Knowledge soundness for the interactive argument follows from Lemmas 2 and 3, whose full proofs are deferred. Knowledge soundness of the Fiat–Shamir-transformed recursive composition is the part we consider open, and we return to it in Section 9.

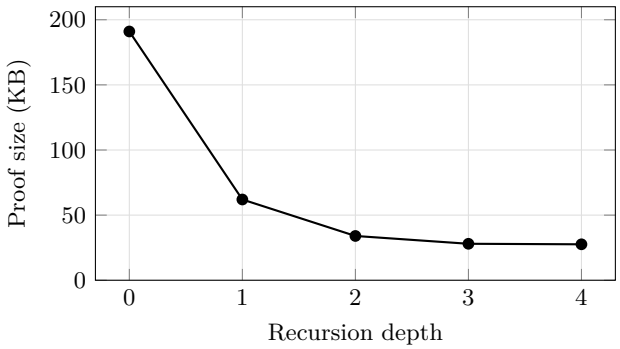

Figure 1: Proof size against recursion depth for a 2^{20}-constraint statement. Depth 3 is the operating point used throughout.

6 Recursive Composition and On-Chain Verification

A single Espalier proof of a 2^{20}-constraint statement is roughly 191 KB, which is too large to post on a public ledger at acceptable cost. We therefore verify Espalier inside Espalier. The verifier circuit is dominated by the NTT-domain multiplications of the commitment check and the projected norm check, both of which are arithmetic-friendly in the same ring, so the recursion does not cross a field boundary and no wrapper in a foreign proof system is required. This is a deliberate departure from the common practice of wrapping a transparent argument in a pairing-based SNARK to shrink the on-chain proof, which reintroduces the pairing assumption and with it the quantum exposure the construction was chosen to avoid.

Figure 1 reports proof size against recursion depth. Size falls steeply for the first two levels and then flattens, since the recursive verifier circuit has a fixed cost that eventually dominates. We use depth 3, at 28 KB, as the operating point: depth 4 saves under half a kilobyte and costs a further proving pass.

The on-chain verifier checks one commitment relation and one projected norm bound over $\mathcal{R}_q$. It requires no pairing and no elliptic-curve arithmetic, only modular arithmetic and hashing, and it costs approximately 310k gas at our parameters, of which the majority is calldata for the proof itself.

7 Implementation and Evaluation

7.1 Prototype

We implemented Espalier as a research prototype in Rust. The NTT is hand-vectorized for 64-bit SIMD lanes, commitments are computed in the NTT domain

Quantity	Value
Constraints	2^{20}
Prover time	1.9 s
Peak prover memory	640 MB
Proof size (depth 3)	28 KB
On-chain verification	$\approx$ 310k gas
Setup	transparent
Assumption	Module-SIS

Table 1: Espalier at the operating point, on a shielded-transfer circuit. Figures are from our prototype and are not independently reproduced.

to avoid redundant transforms, and the cordon decomposition is fused into the packing pass so that the witness is traversed once. The prototype is not constant-time, has not been audited, and is not public; see Section 9.

7.2 Platform

Measurements were taken on an A17-class mobile system-on-chip with six cores and NEON vector units, at a fixed thermal state, with the prover restricted to four cores to leave headroom for a foreground application. This is intended to approximate the conditions under which a wallet would actually generate a proof, rather than a best case. We report medians over 50 runs.

7.3 Results

Table 1 gives the headline figures. A 2^{20}-constraint shielded-transfer circuit proves in 1.9 s with 640 MB of peak prover memory, producing a 28 KB proof verifiable on-chain for approximately 310k gas.

Figure 2 plots prover time against constraint count from 2^{14} to 2^{22}. The curve is close to linear in the log-log plane with slope slightly above one, consistent with $O(n \log n)$ behaviour dominated by the NTT. Figure 3 plots peak prover memory over the same range; memory is the constraint that bites first on a mobile device, and at 2^{22} the prototype exceeds what we would consider safe for a foreground application, which sets a practical ceiling on single-instance circuit size and is part of the motivation for lateral amortization.

7.4 Batching

Lateral amortization was measured on batches of identical shielded-transfer circuits. At $k = 16$ the per-instance prover time falls to 0.21 s, against 1.9 s for a single instance, and the proof remains one proof of 28 KB. Beyond $k = 64$ the norm growth of Lemma 3 forces a larger β_{agg} and the per-instance gain flattens, so we recurse over batches instead.

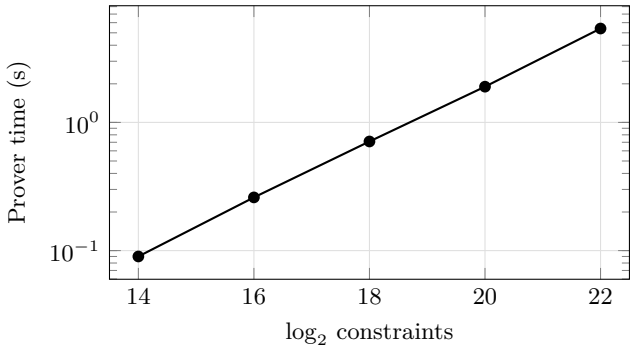

Figure 2: Prover wall-clock time against constraint count, four cores, A17-class SoC.

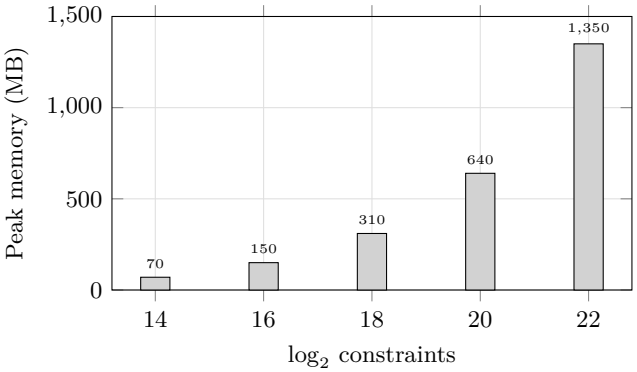

Figure 3: Peak prover memory against constraint count. Memory, not time, is the binding constraint on mobile.

7.5 Comparison

Table 2 places Espalier against widely deployed alternatives on the structural axes: setup model, hardness assumption, post-quantum status, and typical proof size. We deliberately do not report prover timings for the other systems. We did not benchmark them on our platform, and quoting figures measured elsewhere, on different hardware and different circuits, would not be a comparison. The proof-size column gives the ranges commonly reported for privacy-application circuits rather than measurements of ours.

8 Related Work

Ajtai [1] introduced the hash function whose collision resistance is the ancestor of the commitments used here, and Baum et al. [3] gave the module variant with the additive homomorphism that lateral amortization relies on. Lyubashevsky [2] introduced the rejection-sampling technique that makes lattice responses simulatable, which every subsequent construction in this line uses.

LaBRADOR [4] gives compact proofs for R1CS from

System	Setup	PQ	Proof size
Groth16	trusted, per circuit	no	∼200 B
PLONK/KZG	trusted, universal	no	∼400 B
PLONK/IPA	transparent	no	∼1 KB
STARK/FRI	transparent	yes	50–200 KB
Espalier	transparent	yes	28 KB

Table 2: Structural comparison. Proof sizes for systems other than Espalier are the ranges commonly reported for privacy-application circuits, not measurements taken by us. Prover timings are omitted deliberately; see text.

Module-SIS and is the closest point of comparison for our commitment layer and our use of projected norm checks. Greyhound [5] gives fast polynomial commitments from lattices, and addresses a different layer of the stack than the one described here. LatticeFold [6] approaches succinctness through folding over lattices; folding and the composition used here both face the norm-growth problem, and cordon reduction is our answer to it. We regard a careful comparison against folding-based composition as the most important piece of work not present in this preprint.

Outside the lattice setting, FRI [8] underpins the hash-based systems that are the other transparent post-quantum option, and work on towers of binary fields [9] attacks prover cost by changing the field rather than the assumption. Groth16 [7] remains the reference point for on-chain verification cost, and is not post-quantum.

9 Limitations and Future Work

We want to be plain about the status of this document.

There is no public implementation. The prototype described in Section 7 is internal. It is not open source at the time of writing, it has not been audited, and it is not constant-time. Nothing here should be deployed.

The numbers are ours. Every figure in Section 7 was measured by us, on our prototype, on our hardware. None has been independently reproduced. Readers should treat them as an indication of what the construction might achieve rather than as an established result, and we would not object to being told they are optimistic.

The proofs are not here. Lemmas 2 and 3 are stated with sketches. The knowledge extractor for the recursive composition under Fiat–Shamir is the weakest point of the present draft: the interactive argument composes cleanly, but the standard subtleties of extracting through a Fiat–Shamir-transformed recursion apply here as elsewhere, and we do not claim to have resolved them. The extended version will either give the extractor or say clearly what it costs.

Parameters are provisional. The choice $d = 64$ with a 64-bit modulus was made for machine-word arithmetic rather than by optimizing concrete security, and we have not run a lattice estimator against the final parameter set. The security level we are targeting is 128 bits classical, and we do not currently claim to have verified it.

What we intend to do next. Complete the extraction argument; run a concrete security analysis and fix parameters; compare against folding-based composition on equal terms; and release an implementation.

References

[1] M. Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996.

[2] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

[3] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, 2018.

[4] W. Beullens and G. Seiler. LaBRADOR: Compact proofs for R1CS from Module-SIS. In *CRYPTO*, 2023.

[5] N. K. Nguyen and G. Seiler. Greyhound: Fast polynomial commitments from lattices. In *CRYPTO*, 2024.

[6] D. Boneh and B. Chen. LatticeFold: A lattice-based folding scheme and its applications to succinct proof systems. 2024.

[7] J. Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT*, 2016.

[8] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. Fast Reed-Solomon interactive oracle proofs of proximity. In *ICALP*, 2018.

[9] B. Diamond and J. Posen. Succinct arguments over towers of binary fields. 2023.