LaBRADOR: Compact proofs for R1CS from Module-SIS

Ward Beullens, Gregor Seiler

IBM Research Europe

August 23, 2023

Motivation for Lattice-Based Proof Systems

- Quantum Security

Motivation for Lattice-Based Proof Systems

- Quantum Security
- Suitability for building quantum-safe privacy-preserving protocols

Motivation for Lattice-Based Proof Systems

- Quantum Security
- Suitability for building quantum-safe privacy-preserving protocols
- Better proof sizes than hash-based STARKs

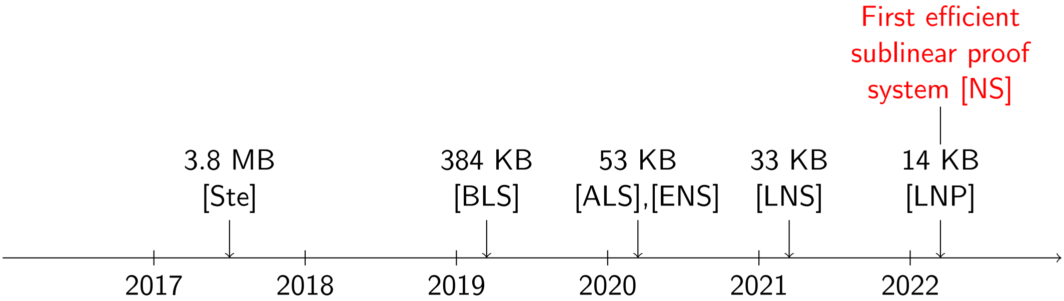
Motivation for Lattice-Based Proof Systems

- Quantum Security
- Suitability for building quantum-safe privacy-preserving protocols
- Better proof sizes than hash-based STARKs
- High computational performance

Motivation for Lattice-Based Proof Systems

- Quantum Security
- Suitability for building quantum-safe privacy-preserving protocols
- Better proof sizes than hash-based STARKs
- High computational performance

Evolution of Linear-Sized Lattice-Based Proof Systems



LaBRADOR

Lattice-Based Recursively Amortized Demonstrations Of R1CS

Highlights:

- Proof size of < 60 KB for large statements
- Recursive structure

~~R1CS~~ Principal Relation

Parameterized by a rank n and a multiplicity r

Witness consists of r (polynomial) vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r$ of rank n that fulfill many dot-product constraints

$$f^{(k)}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r) = \sum_{i,j} \boldsymbol{a}_{ij}^{(k)} \langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle + \sum_i \langle \boldsymbol{\varphi}_i^{(k)}, \boldsymbol{s}_i \rangle + \boldsymbol{b}^{(k)} = \boldsymbol{0},$$

and a norm constraint

$$\|\boldsymbol{s}_1\|^2 + \cdots + \|\boldsymbol{s}_r\|^2 \leq \beta^2$$

Protocol can be seen as a chain of sub-protocols that transform the relation into new instances with smaller parameters

Inner Commitments

Need commitments to $\boldsymbol{s}_i$ for sound transformations of relation

- Prover sends $\boldsymbol{t}_i = \boldsymbol{A}\boldsymbol{s}_i$ for $i = 1, \ldots, r$

Note: All commitments share same matrix $\boldsymbol{A}$ (commitment key)

Outer Commitments

Sending lattice commitments is very expensive ($\approx$ 4KB per commitment)

Idea: Hide inner commitments $\boldsymbol{t}_i$ in an outer commitment

$$\boldsymbol{u} = \sum_{i,k} \boldsymbol{B}_{ik} \boldsymbol{t}_i^{(k)} \quad \text{where} \quad \boldsymbol{t}_i = \boldsymbol{t}_i^{(0)} + b\boldsymbol{t}_i^{(1)} \cdots + b^{(t-1)} \boldsymbol{t}_i^{(t-1)} \text{ with } \left\| \boldsymbol{t}_i^{(k)} \right\|_\infty \leq \frac{b}{2}$$

Outer commitment needs to account for less slack; hence much smaller

Johnson-Lindenstrauss Projection

Recall: Certain random linear maps from a high-dimensional into a low-dimensional vector space preserve the ℓ_2-norm up to small constants

- Verifier sends random matrices Π_i
- Prover sends projection $\vec{p} = \sum_i \Pi_i \vec{s}_i \in \mathbb{Z}_q^{256}$
- Verifier checks $\|\vec{p}\| \leq \sqrt{128}\beta$

Aggregation

Randomly linear-combine dot-product constraints $f^{(k)}$ with uniform challenges

- Verifier sends challenges α_k
- Prover and verifier compute aggregated constraint

$$f(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r) = \sum_{i,j} \boldsymbol{a}_{ij} \langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle + \sum_i \langle \varphi_i, \boldsymbol{s}_i \rangle + \boldsymbol{b} = \boldsymbol{0} \quad \text{where} \quad \begin{cases} \boldsymbol{a}_{ij} = \sum_k \alpha_k \boldsymbol{a}_{ij}^{(k)}, \\ \varphi_i = \sum_k \alpha_k \varphi_i^{(k)}, \\ \boldsymbol{b} = \sum_k \alpha_k \boldsymbol{b}^{(k)} \end{cases}$$

Amortization

Amortize over witness vectors $\boldsymbol{s}_i$

- Prover sends garbage polynomials $\boldsymbol{g}_{ij} = \langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle$ and $\boldsymbol{h}_{ij} = \langle \boldsymbol{\varphi}_i, \boldsymbol{s}_j \rangle$
- Verifier sends challenge polynomials $\boldsymbol{c}, \dots, \boldsymbol{c}_r$
- Prover sends amortized opening

$$\boldsymbol{z} = \boldsymbol{c}_1 \boldsymbol{s}_1 + \boldsymbol{c}_2 \boldsymbol{s}_2 + \cdots + \boldsymbol{c}_r \boldsymbol{s}_r$$

Target Relation of Multiplicity 2

Witness:

$$\boldsymbol{z}, \boldsymbol{v} = \left(\boldsymbol{t}_i^{(k)}\right)$$

Constraints:

$$\sum_{i,j} \boldsymbol{a}_{ij}\boldsymbol{g}_{ij} + \sum_i \boldsymbol{h}_{ii} + \boldsymbol{b} = \boldsymbol{0} \qquad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{c}_1\boldsymbol{t}_1 + \cdots + \boldsymbol{c}_r\boldsymbol{t}_r$$

$$\langle \boldsymbol{z}, \boldsymbol{z} \rangle = \sum_{i,j} \boldsymbol{c}_i\boldsymbol{c}_j\boldsymbol{g}_{ij} \qquad \sum_{i,k} \boldsymbol{B}_{ik}\boldsymbol{t}_i^{(k)} = \boldsymbol{u}$$

$$\langle \varphi, \boldsymbol{z} \rangle = \sum_{i,j} \boldsymbol{c}_i\boldsymbol{c}_j\boldsymbol{h}_{ij} \qquad \|\boldsymbol{z}\|^2 + \|\boldsymbol{v}\|^2 \leq \beta''^2$$

Decomposition in Rank

Before recursing the protocol, want to increase multiplicity and decrease rank

Decomposition in rank: Split vectors of rank n into r vectors of rank n/r:

$$\boldsymbol{z} = \boldsymbol{z}_1 \parallel \cdots \parallel \boldsymbol{z}_r$$

Quadratic term $\langle \boldsymbol{z}, \boldsymbol{z} \rangle$ transforms as

$$\langle \boldsymbol{z}, \boldsymbol{z} \rangle = \langle \boldsymbol{z}_1, \boldsymbol{z}_1 \rangle + \cdots + \langle \boldsymbol{z}_r, \boldsymbol{z}_r \rangle$$

Decomposition in Width

Amortization blows up standard deviation due to multiplication by challenge polynomials; consequently, lattice parameters need to increase to retain SIS-hardness

Decomposition in width:

$$\boldsymbol{z} = \boldsymbol{z}_0 + b\boldsymbol{z}_1 \text{ with } \|\boldsymbol{z}_0\|_\infty \leq \frac{b}{2}$$

Quadratic term $\langle \boldsymbol{z}, \boldsymbol{z} \rangle$ transforms as

$$\langle \boldsymbol{z}, \boldsymbol{z} \rangle = \langle \boldsymbol{z}_0, \boldsymbol{z}_0 \rangle + 2b\langle \boldsymbol{z}_0, \boldsymbol{z}_1 \rangle + b^2 \langle \boldsymbol{z}_1, \boldsymbol{z}_1 \rangle$$

Lattice Bulletproofs?

Want to prove commitment $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s} = \boldsymbol{A}_0\boldsymbol{s}_0 + \boldsymbol{A}_1\boldsymbol{s}_1$ using folding $\boldsymbol{z} = \boldsymbol{s}_0 + \boldsymbol{c}\boldsymbol{s}_1$

Bulletproofs: Quadratic verification using bilinearity of commitment:

$$(\boldsymbol{A}_0 + \boldsymbol{c}\boldsymbol{A}_1)(\boldsymbol{s}_0 + \boldsymbol{c}\boldsymbol{s}_1) = \boldsymbol{A}_0\boldsymbol{s}_0 + \boldsymbol{c}(\boldsymbol{A}_0\boldsymbol{s}_1 + \boldsymbol{A}_1\boldsymbol{s}_0) + \boldsymbol{c}^2\boldsymbol{A}_1\boldsymbol{s}_1 = \boldsymbol{t} + \boldsymbol{c}\boldsymbol{t}_1 + \boldsymbol{c}^2\boldsymbol{t}_2$$

Generalization to n parts needs $O(n^2)$ garbage commitments

Amortization: Linear verification

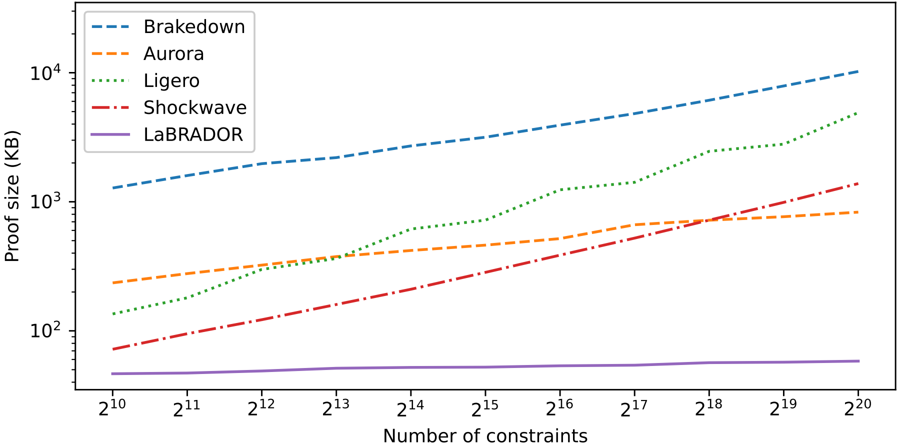
$$\boldsymbol{A}_0(\boldsymbol{s}_0 + \boldsymbol{c}\boldsymbol{s}_1) = \boldsymbol{A}_0\boldsymbol{s}_0 + \boldsymbol{A}_0\boldsymbol{s}_1 = \boldsymbol{t}_0 + \boldsymbol{c}\boldsymbol{t}_1$$

using only n "garbage commitments". Doesn't prove initial commitment $\boldsymbol{t}$. But can collapse ("aggregate") initial commitment to single polynomial and prove with $O(n^2)$ garbage polynomials.

Results I: Proof sizes in Kilobytes for binary R1CS

No. of constraints	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}
Proof Size (KB)	49.02	49.37	51.47	51.6	52.7	53.84

Results II: R1CS mod $2^{64} + 1$



Thank you!