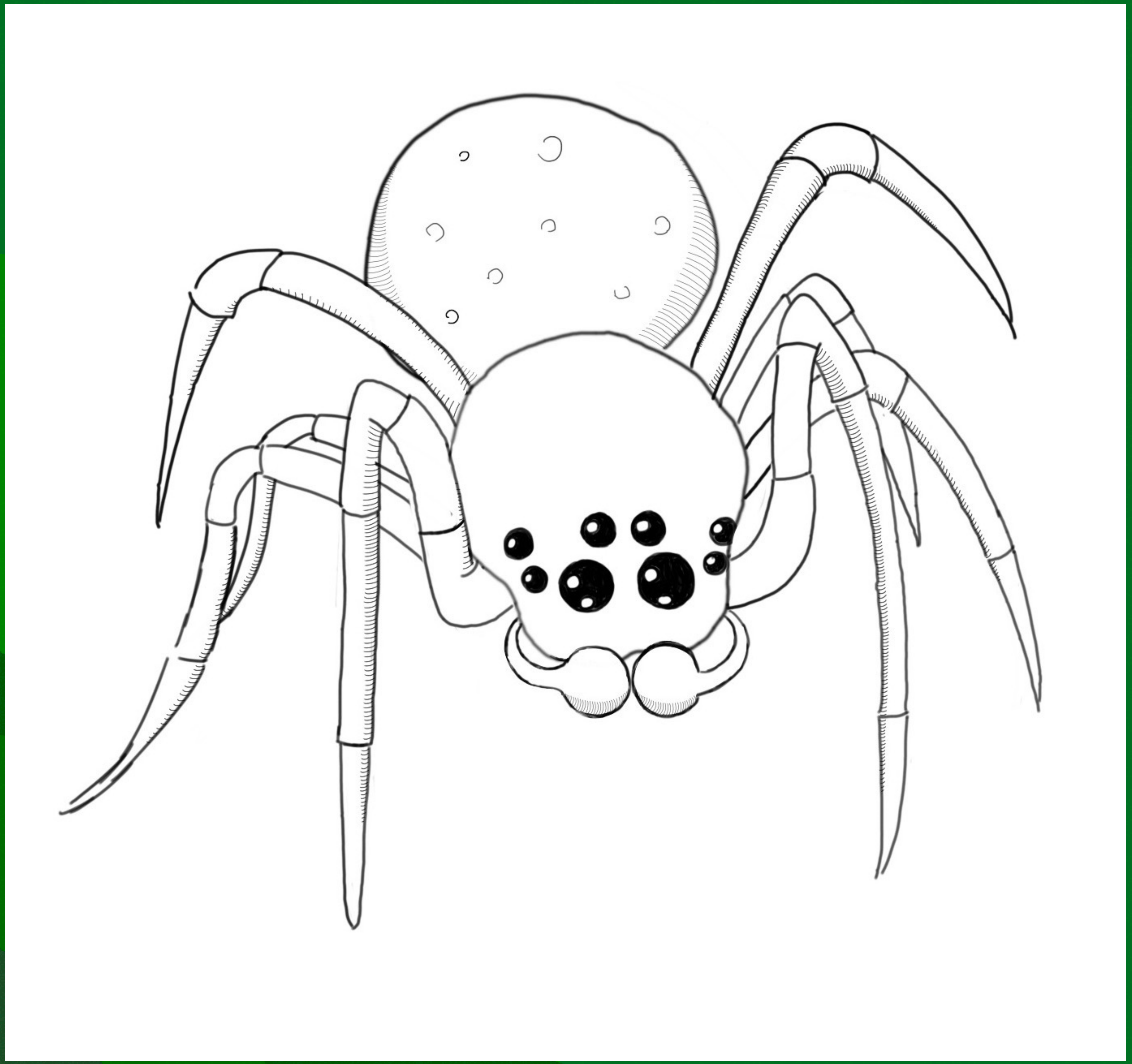
Orbweaver

Succinct Linear Functional Commitments from Lattices



Lattice Orbweaver Spider by Jackie P (CC BY 4.0)

Ben Fisch, Zeyu Liu, and Psi Vesely

Yale University

Results

- Lattice arguments with $O(\log n \log \log n)$ complexity verifier*

Results

- Lattice arguments with $O(\log n \log \log n)$ complexity verifier* ($O(\log^{1.58} n)$ with Karatsuba)

Results

- Lattice arguments with $O(\log n \log \log n)$ complexity verifier* ($O(\log^{1.58} n)$ with Karatsuba)
- Constructions for both cyclotomic rings R_q and integers $\mathbb{Z}_q$ of:
    - Linear map functional commitments/ inner product argument
    - Polynomial commitments
    - SNARK for R1CS

Results

- Lattice arguments with $O(\log n \log \log n)$ complexity verifier* ($O(\log^{1.58} n)$ with Karatsuba)
- Constructions for both cyclotomic rings R_q and integers $\mathbb{Z}_q$ of:
  - Linear map functional commitments/ inner product argument
  - Polynomial commitments
  - SNARK for R1CS
- All extractable, preprocessing, mostly structure-preserving

Results

- Lattice arguments with $O(\log n \log\log n)$ complexity verifier* ($O(\log^{1.58} n)$ with Karatsuba)
- Constructions for both cyclotomic rings R_q and integers $\mathbb{Z}_q$ of:
  - Linear map functional commitments/ inner product argument
  - Polynomial commitments
  - SNARK for R1CS
- All extractable, preprocessing, mostly structure-preserving

Abstract linear map equation

$$\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1,\\ i\neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

Form used in [Gro10,LRY16,AC20]

Evaluation verification equation

$\mathbf{f}, \mathbf{x}$ short

$$\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i = -n+1, \\ i \neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]

Evaluation verification equation

$\mathbf{f}, \mathbf{x}$ short

$$\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1,\\ i \neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

$$c_{\mathbf{f}} \cdot c_{\mathbf{x}}$$

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]

Ring Vandermonde SIS (R-V-SIS) commitment

$$c := \sum_{i=1}^{n} x_i \cdot v^i \mod q \,, \text{ where } v \xleftarrow{\$} R_q \text{ is } \textit{public}$$

Ring Vandermonde SIS (R-V-SIS) commitment

$$c := \sum_{i=1}^{n} x_i \cdot v^i \mod q\,, \text{ where } v \overset{\$}{\leftarrow} R_q \text{ is } \textit{public}$$

- Ajtai's R-SIS commitment, with a Vandermonde key

Ring Vandermonde SIS (R-V-SIS) commitment

$$c := \sum_{i=1}^{n} x_i \cdot v^i \mod q\,, \text{ where } v \xleftarrow{\$} R_q \text{ is } \textit{public}$$

- Ajtai's R-SIS commitment, with a Vandermonde key
- Similar to assumption used in PASS Sign. If we pick v instead from the primitive roots of unity binding reduces to Vandermonde R-SIS [HS15,LZA18,BSS22]

Evaluation verification equation

$\mathbf{f}, \mathbf{x}$ short

$$\underbrace{\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right)}_{c_{\mathbf{f}}} \cdot \underbrace{\left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right)}_{c_{\mathbf{x}}} \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1,\\ i \neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

$c_{\mathbf{f}} \cdot c_{\mathbf{x}}$

(preprocessed)

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]

Evaluation verification equation

$\mathbf{f}, \mathbf{x}$ short

$$\underline{\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right)} \cdot \underline{\left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right)} \equiv \underline{\langle \mathbf{f}, \mathbf{x} \rangle} + \sum_{\substack{i=-n+1,\\ i \neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

$$c_{\mathbf{f}} \quad \cdot \quad c_{\mathbf{x}} \quad \equiv \quad y$$

(preprocessed)

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]

Evaluation verification equation

$\mathbf{f}, \mathbf{x}$ short

$$\left(\sum_{i=1}^{n} f_i \cdot Y^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot Y^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1,\\ i \neq 0}}^{n-1} b_i \cdot Y^i \mod q$$

$$c_{\mathbf{f}} \cdot c_{\mathbf{x}} \equiv y + \langle \mathbf{a}, \pi \rangle \mod q$$

(preprocessed)

π short

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]

Prover key

Generate short preimages $\mathbf{u}_i$ for $i \in \{-n+1,\ldots,n-1\}\backslash\{0\}$ such that

$$\langle \mathbf{a}, \mathbf{u}_i \rangle \equiv v^i \mod q$$

Using a trapdoor public SIS matrix $\mathbf{a}$ [MP12]

Computing the proof

- Given $\langle \mathbf{a}, \mathbf{u}_i \rangle \equiv v^i \mod q$ except for $i = 0$
- Where b_i is the sum of cross terms corresponding to the coefficient of v^i compute

$$\pi := \sum_{\substack{i=-n+1,\\ i\neq 0}}^{n-1} b_i \cdot \mathbf{u}_i \mod q$$

Computing the proof

- Given $\langle \mathbf{a}, \mathbf{u}_i \rangle \equiv v^i \mod q$ except for $i = 0$
- Where b_i is the sum of cross terms corresponding to the coefficient of v^i compute

$$\pi := \sum_{\substack{i=-n+1, \\ i \neq 0}}^{n-1} b_i \cdot \mathbf{u}_i \mod q$$

- Then

$$\langle \mathbf{a}, \pi \rangle \equiv \sum_{\substack{i=-n+1, \\ i \neq 0}}^{n-1} b_i \cdot v_i \mod q$$

Computing the proof

- Given $\langle \mathbf{a}, \mathbf{u}_i \rangle \equiv v^i \mod q$ except for $i = 0$
- Where b_i is the sum of cross terms corresponding to the coefficient of v^i compute

$$\pi := \sum_{\substack{i=-n+1, \\ i \neq 0}}^{n-1} b_i \cdot \mathbf{u}_i \mod q$$

- Then

$$\langle \mathbf{a}, \pi \rangle \equiv \sum_{\substack{i=-n+1, \\ i \neq 0}}^{n-1} b_i \cdot v^i \mod q$$

- $\mathbf{f}, \mathbf{x}$ short $\Longrightarrow b_i$ short, u_i short $\Longrightarrow \pi$ short

Evaluation binding

$$\left(\sum_{i=1}^{n} f_i \cdot v^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot v^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1, \\ i \neq 0}}^{n-1} b_i \cdot v^i \mod q$$

$$c_{\mathbf{f}} \cdot c_{\mathbf{x}} \equiv y + \langle \mathbf{a}, \pi \rangle \mod q$$

Evaluation binding

$$\left(\sum_{i=1}^{n} f_i \cdot v^{-i}\right) \cdot \left(\sum_{i=1}^{n} x_i \cdot v^{i}\right) \equiv \langle \mathbf{f}, \mathbf{x} \rangle + \sum_{\substack{i=-n+1,\\ i \neq 0}}^{n-1} b_i \cdot v^i \mod q$$

$$c_{\mathbf{f}} \cdot c_{\mathbf{x}} \equiv y + \langle \mathbf{a}, \pi \rangle \mod q$$

$$\langle \mathbf{a}, \pi - \pi' \rangle \equiv y' - y \mod q$$

Evaluation binding

$$\langle \mathbf{a}, \pi - \pi' \rangle \equiv y' - y \mod q$$

- k-R-ISIS family of assumptions: can only generate short preimages for targets within a short linear span of the v^i or for random targets [ACLMT22]

Evaluation binding

$$\langle \mathbf{a}, \pi - \pi' \rangle \equiv y' - y \mod q$$

- k-R-ISIS family of assumptions: can only generate short preimages for targets within a short linear span of the v^i or for random targets [ACLMT22]
- $y' - y$ is short, while for $v \xleftarrow{\$} R_q$ all $v^i \mod q$ will be long whp, as will the random targets

Multiple outputs

Can prove $\langle \mathbf{f}_i, \mathbf{x} \rangle = y_i$ for $i \in [t]$ with a single evaluation proof:

$$\langle \mathbf{a}, \pi \rangle \equiv c \cdot \sum_{i=1}^{t} h_i \cdot \mathsf{ck}_{\mathbf{f}_i} - \sum_{i=1}^{t} h_i \cdot y_i \mod q$$

Multiple outputs

Can prove $\langle \mathbf{f}_i, \mathbf{x} \rangle = y_i$ for $i \in [t]$ with a single evaluation proof:

$$\langle \mathbf{a}, \pi \rangle \equiv c \cdot \sum_{i=1}^{t} h_i \cdot \mathsf{ck}_{\mathbf{f}_i} - \sum_{i=1}^{t} h_i \cdot y_i \mod q$$

Key observation: the prover submits a separate knowledge proof π' for c from which we extract $\mathbf{x}$. It's thus unnecessary to extract the hypothetical π_i s.t.

$$\pi = \sum_{i=1}^{t} h_i \cdot \pi_i$$

Multiple outputs

Using extracted $\mathbf{x}$ we get

$$\langle \mathbf{a}, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot \left(\langle \mathbf{f}_i, \mathbf{x} \rangle - y_i \right) - \sum_{i=1}^{t} h_i \cdot \sum_{\substack{j = -n+1, \\ j \neq 0}}^{n-1} b_{i,j} \cdot Y^i \mod q$$

Multiple outputs

$$\langle \mathbf{a}, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot \left(\langle \mathbf{f}_i, \mathbf{x} \rangle - y_i \right) - \sum_{i=1}^{t} h_i \cdot \sum_{\substack{j=-n+1, \\ j \neq 0}}^{n-1} b_{i,j} \cdot Y^i \mod q$$

$$p(h_1, \ldots, h_t)$$

- For $h_1, \ldots, h_t \leftarrow \mathscr{H}$ want $p(h_1, \ldots, h_t) = 0$ only with negligible probability if p is not the zero polynomial

Multiple outputs

$$\langle \mathbf{a}, \pi \rangle \equiv \underline{\sum_{i=1}^{t} h_i \cdot \left(\langle \mathbf{f}_i, \mathbf{x} \rangle - y_i \right)} - \sum_{i=1}^{t} h_i \cdot \sum_{\substack{j = -n+1, \\ j \neq 0}}^{n-1} b_{i,j} \cdot Y^i \mod q$$

$$p(h_1, \ldots, h_t)$$

- For $h_1, \ldots, h_t \leftarrow \mathcal{H}$ want $p(h_1, \ldots, h_t) = 0$ only with negligible probability if p is not the zero polynomial
- Can pick *exponential size* “exceptional set” $\mathcal{H}$ over R_q for large q [LS18] and invoke Generalized Alon-Füredi Theorem [BCPS18]

Multiple outputs

$$\langle \mathbf{a}, \pi \rangle \equiv \underline{\sum_{i=1}^{t} h_i \cdot \left(\langle \mathbf{f}_i, \mathbf{x} \rangle - y_i \right)} - \sum_{i=1}^{t} h_i \cdot \sum_{\substack{j=-n+1, \\ j \neq 0}}^{n-1} b_{i,j} \cdot Y^i \mod q$$

$$p(h_1, \ldots, h_t)$$

- For $h_1, \ldots, h_t \leftarrow \mathscr{H}$ want $p(h_1, \ldots, h_t) = 0$ only with negligible probability if p is not the zero polynomial
- Can pick *exponential size* “exceptional set” $\mathscr{H}$ over R_q for large q [LS18] and invoke Generalized Alon-Füredi Theorem [BCPS18]
- Better to perform ternary decomposition on $\mathbf{f}, \mathbf{x}$ and batch verification

Proof and SRS sizes for $\mathbb{Z}_{2^{32}}$

log2(x)	18	22	26	30
|c| (B)	293	347	422	505
total proof size (KiB)	845	1,081	1,315	1,571
verifier key (MiB)	12	17	23	30
prover key (GiB)	0.3	6	111	2,070

- These are maximum proof sizes. When $\mathbf{f}$ or $\mathbf{x}$ are sparse or have entries much smaller than the norm bound this is reflected by the proof size.

Proof and SRS sizes for $\mathbb{Z}_{2^{32}}$

log2(x)	18	22	26	30
|c| (B)	293	347	422	505
total proof size (KiB)	845	1,081	1,315	1,571
verifier key (MiB)	12	17	23	30
prover key (GiB)	0.3	6	111	2,070

- These are maximum proof sizes. When $\mathbf{f}$ or $\mathbf{x}$ are sparse or have entries much smaller than the norm bound this is reflected by the proof size.
- Binding only version reduces proof size by ~65%, prover key size by ~75%

Proof and SRS sizes for $\mathbb{Z}_{2^{32}}$

log2(x)	18	22	26	30
|c| (B)	293	347	422	505
total proof size (KiB)	845	1,081	1,315	1,571
verifier key (MiB)	12	17	23	30
prover key (GiB)	0.3	6	111	2,070

- These are maximum proof sizes. When $\mathbf{f}$ or $\mathbf{x}$ are sparse or have entries much smaller than the norm bound this is reflected by the proof size.
- Evaluation binding only version (no extractability) reduces proof size by ~65%, prover key size by ~75%
- Smallest compressing proofs start around 165 KiB (binding) and 668 KiB (extractable) — recursion threshold

Lattice-based Succinct Arguments from Vanishing Polynomials

Valerio Cini[1], Russell W. F. Lai[2], Giulio Malavolta[3]

[1] AIT Austrian Institute of Technology, Austria
[2] Aalto University, Finland
[3] Max Planck Institute for Security and Privacy, Germany

CRYPTO, Santa Barbara, CA, U.S., 2023

Lattice-based Succinct Arguments

Approach	Publicly verifiable	Sublinear-verifier (preprocessing)	Linear-prover
PCP/IOP + linear-only enc. [BCIOP13; BISW17; BISW18; GMNO18]	✘	✓	✓
Linearisation + folding [BLNS20; AL21; ACK21; BS22]	✓	✘ $\tilde{O}_\lambda(\lvert\text{stmt}\rvert)$	✓
Direct [ACLMT22]	✓	✓	✘ $\tilde{O}_\lambda(\lvert\text{stmt}\rvert^2)$
This work (and [BCS23])	✓	✓	✓

Lattice-based Succinct Arguments

Approach	Publicly verifiable	Sublinear-verifier (preprocessing)	Linear-prover
PCP/IOP + linear-only enc. [BCIOP13; BISW17; BISW18; GMNO18]	✘	✓	✓
Linearisation + folding [BLNS20; AL21; ACK21; BS22]	✓	✘ $\tilde{O}_\lambda(\lvert\mathsf{stmt}\rvert)$	✓
Direct [ACLMT22]	✓	✓	✘ $\tilde{O}_\lambda(\lvert\mathsf{stmt}\rvert^2)$
This work (and [BCS23])	✓	✓	✓

Our Results

- New assumption: Vanishing Short Integer Solution (vSIS)
  - generalization of SIS
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small ($\mathrm{polylog}(|\mathsf{stmt}|)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}(|\mathsf{stmt}|)$-prover $\mathrm{polylog}(|\mathsf{stmt}|)$-verifier arguments for commitment openings
- New lattice-based succinct arguments for NP $\Leftarrow$ Succinct arguments for vSIS commitment openings

Our Results

- New assumption: Vanishing Short Integer Solution (vSIS)
  - generalization of SIS
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small ($\text{polylog}(|\text{stmt}|)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}(|\text{stmt}|)$-prover $\text{polylog}(|\text{stmt}|)$-verifier arguments for commitment openings
- New lattice-based succinct arguments for NP $\Leftarrow$ Succinct arguments for vSIS commitment openings

Our Results

- New assumption: Vanishing Short Integer Solution (vSIS)
  - generalization of SIS
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small (polylog$\big(|\text{stmt}|\big)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}\big(|\text{stmt}|\big)$-prover polylog$\big(|\text{stmt}|\big)$-verifier arguments for commitment openings
- New lattice-based succinct arguments for NP $\Leftarrow$ Succinct arguments for vSIS commitment openings

Our Results

Instantiations	$\|\pi\|$	Time($\mathcal{P}$)	Time($\mathcal{V}$)	Setup	Assumptions
Folding	$\tilde{O}_\lambda(1)$	$\tilde{O}_\lambda(\|\text{stmt}\|)$	$\tilde{O}_\lambda(1)$	Transparent	vSIS (+ RO for NI)
Knowledge assumption	$\tilde{O}_\lambda(1)$	$\tilde{O}_\lambda(\|\text{stmt}\|)$	$\tilde{O}_\lambda(1)$	Trusted	vSIS + Knowledge-kRISIS

Roadmap

1. vSIS assumptions and commitments
2. Quadratic Relations using vSIS commitments
3. Succinct arguments for vSIS commitment openings

Short Integer Solution (SIS) Assumption

- Parameters: # rows n, # columns m, modulus q.
- Instance: A matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$.
- Problem: Find a short vector $\mathbf{u} \in \mathcal{R}^m$ such that

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \approx 0.$$

- Shorthand: If $\mathbf{u}$ is a short non-zero vector satisfying $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$, write

$$\mathbf{u} \in \mathbf{A}^{-1}(\mathbf{v}).$$

Vanishing SIS as SIS Generalisations

SIS

Find short solution to linear equations

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{0} \bmod q \quad \text{and} \quad 0 < \|\mathbf{u}\| \approx 0.$$

SIS (Alternative Interpretation)

Find linear function with short coefficients which vanishes at all given points

Vanishing SIS (vSIS)

Find polynomial (from some class) with short coefficients which vanishes at all given points

Vanishing SIS as SIS Generalisations

SIS

Find short solution to linear equations

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \approx 0.$$

SIS (Alternative Interpretation)

Find linear function with short coefficients which vanishes at all given points

Vanishing SIS (vSIS)

Find polynomial (from some class) with short coefficients which vanishes at all given points

Vanishing SIS as SIS Generalisations

SIS

Find short solution to linear equations

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \approx 0.$$

SIS (Alternative Interpretation)

Find linear function with short coefficients which vanishes at all given points

Vanishing SIS (vSIS)

Find polynomial (from some class) with short coefficients which vanishes at all given points

Vanishing Short Integer Solution (vSIS) Assumption

Example: Univariate

- Problem: Find short degree m polynomial without constant term

$$p(X) = p_1 X + \ldots + p_m X^m \in \mathcal{R}[X]$$

which vanishes at $v \in \mathcal{R}_q^\times$ modulo q, i.e.

$$p(v) = 0 \bmod q \qquad \text{and} \qquad 0 < \|p\| \approx 0.$$

In other words, find short vector $\mathbf{p} \in \mathcal{R}^m$ such that

$$\begin{bmatrix} v & v^2 & \ldots & v^m \end{bmatrix} \cdot \mathbf{p} = 0 \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{p}\| \approx 0.$$

Simple vSIS Commitments (or Hash Functions)

- Domain: Polynomials $p \in \mathcal{R}[X, X^{-1}]$ (of some class) with short coefficients.
- Public parameters: Random unit $v \leftarrow\$ \mathcal{R}_q^\times$.
- Commitment of polynomial p:

$$\mathrm{com}(p) = p(v) \bmod q.$$

- Binding: If $p(v) = p'(v) \bmod q$, then we break vSIS, i.e.

$$(p - p')(v) = 0 \bmod q \qquad \left\|p - p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0.$$

- (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):

$$p(v) + p'(v) = (p + p')(v) \bmod q \qquad \left\|p + p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0$$
$$p(v) \cdot p'(v) = (p \cdot p')(v) \bmod q \qquad \left\|p \cdot p'\right\| \lessapprox \|p\| \cdot \left\|p'\right\| \approx 0.$$

Simple vSIS Commitments (or Hash Functions)

† Domain: Polynomials $p \in \mathcal{R}[X, X^{-1}]$ (of some class) with short coefficients.

† Public parameters: Random unit $v \leftarrow\$ \mathcal{R}_q^{\times}$.

† Commitment of polynomial p:

$$\mathrm{com}(p) = p(v) \bmod q.$$

† Binding: If $p(v) = p'(v) \bmod q$, then we break vSIS, i.e.

$$(p - p')(v) = 0 \bmod q \qquad \left\|p - p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0.$$

† (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):

$$p(v) + p'(v) = (p + p')(v) \bmod q \qquad \left\|p + p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0$$

$$p(v) \cdot p'(v) = (p \cdot p')(v) \bmod q \qquad \left\|p \cdot p'\right\| \lessapprox \|p\| \cdot \left\|p'\right\| \approx 0.$$

Simple vSIS Commitments (or Hash Functions)

- Domain: Polynomials $p \in \mathcal{R}[X, X^{-1}]$ (of some class) with short coefficients.
- Public parameters: Random unit $v \leftarrow\$ \mathcal{R}_q^\times$.
- Commitment of polynomial p:

$$\mathrm{com}(p) = p(v) \bmod q.$$

- Binding: If $p(v) = p'(v) \bmod q$, then we break vSIS, i.e.

$$(p - p')(v) = 0 \bmod q \qquad \|p - p'\| \leq \|p\| + \|p'\| \approx 0.$$

- (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):

$$p(v) + p'(v) = (p + p')(v) \bmod q \qquad \|p + p'\| \leq \|p\| + \|p'\| \approx 0$$

$$p(v) \cdot p'(v) = (p \cdot p')(v) \bmod q \qquad \|p \cdot p'\| \lessapprox \|p\| \cdot \|p'\| \approx 0.$$

Simple vSIS Commitments (or Hash Functions)

- Domain: Polynomials $p \in \mathcal{R}[X, X^{-1}]$ (of some class) with short coefficients.
- Public parameters: Random unit $v \leftarrow\$ \mathcal{R}_q^\times$.
- Commitment of polynomial p:

$$\mathrm{com}(p) = p(v) \bmod q.$$

- Binding: If $p(v) = p'(v) \bmod q$, then we break vSIS, i.e.

$$(p - p')(v) = 0 \bmod q \qquad \left\|p - p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0.$$

- (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):

$$p(v) + p'(v) = (p + p')(v) \bmod q \qquad \left\|p + p'\right\| \leq \|p\| + \left\|p'\right\| \approx 0$$

$$p(v) \cdot p'(v) = (p \cdot p')(v) \bmod q \qquad \left\|p \cdot p'\right\| \lessapprox \|p\| \cdot \left\|p'\right\| \approx 0.$$

Encoding Vectors as (Laurent) Polynomials

$$\mathbf{a} := (a_1, \ldots, a_m) \in \mathcal{R}^m \qquad \bar{p}_{\mathbf{a}}(X) := p_{\mathbf{a}}(X^{-1}) := a_1 X^{-1} + a_2 X^{-2} + \ldots + a_m X^{-m}$$

$$\mathbf{b} := (b_1, \ldots, b_m) \in \mathcal{R}^m \qquad p_{\mathbf{b}}(X) := b_1 X + b_2 X^2 + \ldots + b_m X^m$$

Note that

$$\bar{p}_{\mathbf{a}}(X) \cdot p_{\mathbf{b}}(X) = \hat{p}_{\mathbf{a}*\mathbf{b}}(X) \implies \hat{p}_{\mathbf{a}*\mathbf{b}} \text{ has } O(m) \text{ terms (lots of collisions!)}$$

where

- † $\mathbf{a} * \mathbf{b} := \left(\sum_{j-i=k} a_i \cdot b_j\right)_{k=-m}^{m}$ “convolution”, and
- † constant term is given by $\langle \mathbf{a}, \mathbf{b} \rangle$.

Encoding Vectors as (Laurent) Polynomials

$$\mathbf{a} := (a_1, \ldots, a_m) \in \mathcal{R}^m \qquad \bar{p}_{\mathbf{a}}(X) := p_{\mathbf{a}}(X^{-1}) := a_1 X^{-1} + a_2 X^{-2} + \ldots + a_m X^{-m}$$

$$\mathbf{b} := (b_1, \ldots, b_m) \in \mathcal{R}^m \qquad p_{\mathbf{b}}(X) := b_1 X + b_2 X^2 + \ldots + b_m X^m$$

Note that

$$\bar{p}_{\mathbf{a}}(X) \cdot p_{\mathbf{b}}(X) = \hat{p}_{\mathbf{a}*\mathbf{b}}(X) \implies \hat{p}_{\mathbf{a}*\mathbf{b}} \text{ has } O(m) \text{ terms (lots of collisions!)}$$

where

- † $\mathbf{a} * \mathbf{b} := \left(\sum_{j-i=k} a_i \cdot b_j\right)_{k=-m}^{m}$ "convolution", and
- † constant term is given by $\langle \mathbf{a}, \mathbf{b} \rangle$.

Key Example

Want to prove that $\mathbf{x}$ is binary (i.e. $x_i \cdot (1 - x_i) = 0$ for all i).

- † $\mathbf{x}$ is committed in vSIS commitment as $c_{\mathbf{x}} := p_{\mathbf{x}}(v)$.
- † $\mathbf{x}$ is committed also in dual vSIS commitment as $\bar{c}_{\mathbf{x}} := \bar{p}_{\mathbf{x}}(v)$,
- † $\mathbf{1}$ is committed in dual vSIS commitment as $\bar{c}_{\mathbf{1}} := \bar{p}_{\mathbf{1}}(v)$.

Observe that

$$\underbrace{\underbrace{\sum_i x_i \cdot v^i}_{c_{\mathbf{x}}} \cdot \left(\underbrace{\sum_j x_j \cdot v^{-j}}_{\bar{c}_{\mathbf{x}}} - \underbrace{\sum_j 1 \cdot v^{-j}}_{\bar{c}_{\mathbf{1}}} \right)}_{\hat{p}_{\mathbf{x}*(\mathbf{1}-\mathbf{x})}(v)} = \underbrace{\sum_i x_i \cdot (x_i - 1)}_{\langle \mathbf{x}, \mathbf{x}-\mathbf{1} \rangle} + \text{mixed terms}$$

Key Example

Want to prove that $\mathbf{x}$ is binary (i.e. $x_i \cdot (1 - x_i) = 0$ for all i).

- † $\mathbf{x}$ is committed in vSIS commitment as $c_{\mathbf{x}} := p_{\mathbf{x}}(v)$.
- † $\mathbf{x}$ is committed also in dual vSIS commitment as $\bar{c}_{\mathbf{x}} := \bar{p}_{\mathbf{x}}(v)$,
- † $\mathbf{1}$ is committed in dual vSIS commitment as $\bar{c}_{\mathbf{1}} := \bar{p}_{\mathbf{1}}(v)$.

Observe that

$$\underbrace{\underbrace{\sum_i x_i \cdot v^i}_{c_{\mathbf{x}}} \cdot \left(\underbrace{\sum_j x_j \cdot v^{-j}}_{\bar{c}_{\mathbf{x}}} - \underbrace{\sum_j 1 \cdot v^{-j}}_{\bar{c}_{\mathbf{1}}} \right)}_{\hat{p}_{\mathbf{x}*(\mathbf{1}-\mathbf{x})}(v)} = \underbrace{\sum_i x_i \cdot (x_i - 1)}_{\langle \mathbf{x}, \mathbf{x}-\mathbf{1} \rangle} + \text{mixed terms}$$

Key Example

Want to prove that **x** is binary (i.e. $x_i \cdot (1 - x_i) = 0$ for all i).

- † **x** is committed in vSIS commitment as $c_{\mathbf{x}} := p_{\mathbf{x}}(v)$.
- † **x** is committed also in dual vSIS commitment as $\bar{c}_{\mathbf{x}} := \bar{p}_{\mathbf{x}}(v)$,
- † **1** is committed in dual vSIS commitment as $\bar{c}_{\mathbf{1}} := \bar{p}_{\mathbf{1}}(v)$.

Observe that

$$\underbrace{\underbrace{\sum_i x_i \cdot v^i}_{c_{\mathbf{x}}} \cdot \left(\underbrace{\sum_j x_j \cdot v^{-j}}_{\bar{c}_{\mathbf{x}}} - \underbrace{\sum_j 1 \cdot v^{-j}}_{\bar{c}_{\mathbf{1}}} \right)}_{\hat{p}_{\mathbf{x}*(\mathbf{1}-\mathbf{x})}(v)} = \underbrace{\sum_i x_i \cdot (x_i - 1)}_{\langle \mathbf{x}, \mathbf{x}-\mathbf{1} \rangle} + \text{mixed terms}$$

Key Example

Want to prove that $\mathbf{x}$ is binary (i.e. $x_i \cdot (1 - x_i) = 0$ for all i).

- $\mathbf{x}$ is committed in vSIS commitment as $c_{\mathbf{x}} := p_{\mathbf{x}}(v)$.
- $\mathbf{h} \circ \mathbf{x}$ is committed also in dual vSIS commitment as $\bar{c}_{\mathbf{h}\circ\mathbf{x}} := \bar{p}_{\mathbf{h}\circ\mathbf{x}}(v)$,
- $\mathbf{h}$ is committed in dual vSIS commitment as $\bar{c}_{\mathbf{h}} := \bar{p}_{\mathbf{h}}(v)$.

Observe that

$$\underbrace{\underbrace{\sum_i x_i \cdot v^i}_{c_{\mathbf{x}}} \cdot \left(\underbrace{\sum_j h_j \cdot x_j \cdot v^{-j}}_{\bar{c}_{\mathbf{h}\circ\mathbf{x}}} - \underbrace{\sum_j h_j \cdot v^{-j}}_{\bar{c}_{\mathbf{h}}} \right)}_{\hat{p}_{\mathbf{x}*\mathbf{h}\circ(\mathbf{x}-\mathbf{1})}(v)} = \underbrace{\sum_i h_i \cdot x_i \cdot (x_i - 1)}_{\langle \mathbf{h}, \mathbf{x}\circ(\mathbf{x}-\mathbf{1})\rangle} + \text{mixed terms}$$

To prove that a vSIS commitment is committing to a (Laurent) polynomial without constant term:

$$\begin{bmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{bmatrix} \cdot \mathbf{x} = \begin{bmatrix} c_{\mathbf{x}} \\ \bar{c}_{\mathbf{x}} \end{bmatrix} \bmod q \ \wedge \ \|\mathbf{x}\| \approx 0,$$

and

$$\begin{bmatrix} v^{-m} & \dots & v^{-1} & v^1 & \dots & v^m \end{bmatrix} \cdot \mathbf{w} = \underbrace{c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{c}_{\mathbf{1}})}_{\hat{c}} \bmod q \ \wedge \ \|\mathbf{w}\| \approx 0,$$

1. using knowledge-kRISIS [ACLMT22], or
2. using folding arguments "Bulletproofs" [BLNS20]

To prove that a vSIS commitment is committing to a (Laurent) polynomial without constant term:

$$\begin{bmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{bmatrix} \cdot \mathbf{x} = \begin{bmatrix} c_{\mathbf{x}} \\ \bar{c}_{\mathbf{x}} \end{bmatrix} \bmod q \ \wedge \ \|\mathbf{x}\| \approx 0,$$

and

$$\begin{bmatrix} v^{-m} & \dots & v^{-1} & v^1 & \dots & v^m \end{bmatrix} \cdot \mathbf{w} = \underbrace{c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{c}_1)}_{\hat{c}} \bmod q \ \wedge \ \|\mathbf{w}\| \approx 0,$$

1. using knowledge-kRISIS [ACLMT22], or
2. using folding arguments “Bulletproofs” [BLNS20]

Knowledge-kRISIS Assumption(s) [ACLMT22] (a Member of)

- Parameters:
  - SIS parameters (n, m, q),
  - submodule rank $t < n$, and
  - t-tuples of Laurent monomials $\mathcal{G}$.
- Assumption: If a PPT (quantum) algorithm $\mathcal{A}$, which on input

$$(\mathbf{A}, \mathbf{T}, v, (\mathbf{u_g})_{\mathbf{g} \in \mathcal{G}})$$

where $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{T} \in (\mathcal{R}_q^\times)^{n \times t}$, $v \in \mathcal{R}_q^\times$, and $\mathbf{u}_g \in \mathbf{A}^{-1}(\mathbf{T} \cdot \mathbf{g}(v))$,

can find $(\mathbf{u}, \mathbf{c})$ where

$$\mathbf{u} \in \mathbf{A}^{-1}(\mathbf{T} \cdot \mathbf{c}),$$

then it must “know” short linear combination $\mathbf{x}$ such that

$$\mathbf{c} = \sum_{g \in \mathcal{G}} \mathbf{g}(v) \cdot x_g \bmod q.$$

Knowledge-kRISIS Assumption(s) [ACLMT22] (a Member of)

- Parameters:
  - SIS parameters (n, m, q),
  - submodule rank $t < n$, and
  - t-tuples of Laurent monomials $\mathcal{G}$.
- Assumption: If a PPT (quantum) algorithm $\mathcal{A}$, which on input

$$(\mathbf{A}, \mathbf{T}, v, (\mathbf{u_g})_{\mathbf{g}\in\mathcal{G}})$$

where $\quad \mathbf{A} \in \mathcal{R}_q^{n\times m}, \quad \mathbf{T} \in (\mathcal{R}_q^{\times})^{n\times t}, \quad v \in \mathcal{R}_q^{\times}, \quad$ and $\quad \mathbf{u}_g \in \mathbf{A}^{-1}(\mathbf{T}\cdot\mathbf{g}(v)),$

can find $(\mathbf{u}, \mathbf{c})$ where

$$\mathbf{u} \in \mathbf{A}^{-1}(\mathbf{T}\cdot\mathbf{c}),$$

then it must “know” short linear combination $\mathbf{x}$ such that

$$\mathbf{c} = \sum_{g\in\mathcal{G}} \mathbf{g}(v)\cdot x_g \bmod q.$$

Using Knowledge-kRISIS

Want to prove $\hat{c}$ and $\mathbf{w} \in \mathcal{R}^{2m+1}$ satisfies:

$$w_0 = 0 \qquad \hat{c} = \hat{\rho}_{\mathbf{w}}(v) \qquad \|\mathbf{w}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{t}, v, (\mathbf{u}_i)_{i \in \pm[m]})$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}(\mathbf{t} \cdot v^i).$$

- Prover: Output $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot w_i$.
- Verifier: Check that $\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot \hat{c} \bmod q$ and $\|\mathbf{u}\| \approx 0$.
- Knowledge-soundness follows immediately from the knowledge-kRISIS assumption.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time.

Using Knowledge-kRISIS

Want to prove $\hat{c}$ and $\mathbf{w} \in \mathcal{R}^{2m+1}$ satisfies:

$$w_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{w}}(v) \qquad \|\mathbf{w}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{t}, v, (\mathbf{u}_i)_{i \in \pm[m]})$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}(\mathbf{t} \cdot v^i).$$

- Prover: Output $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot w_i$.
- Verifier: Check that $\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot \hat{c} \bmod q$ and $\|\mathbf{u}\| \approx 0$.
- Knowledge-soundness follows immediately from the knowledge-kRISIS assumption.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time.

Using Knowledge-kRISIS

Want to prove $\hat{c}$ and $\mathbf{w} \in \mathcal{R}^{2m+1}$ satisfies:

$$w_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{w}}(v) \qquad \|\mathbf{w}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{t}, v, (\mathbf{u}_i)_{i \in \pm[m]})$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}(\mathbf{t} \cdot v^i).$$

- Prover: Output $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot w_i$.
- Verifier: Check that $\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot \hat{c} \bmod q$ and $\|\mathbf{u}\| \approx 0$.
- Knowledge-soundness follows immediately from the knowledge-kRISIS assumption.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time.

Lattice-based Bulletproofs

Goal: Prove SIS relation with $O(\log m)$ communication:

$$\mathbf{x} \in \mathcal{R}^m : \ \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q \land \|\mathbf{x}\| \approx 0$$

where $m = 2^\ell$, $\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2]$, $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$.

Prover $\mathcal{P}((\mathbf{M}, \mathbf{y}), \mathbf{x})$		Verifier $\mathcal{V}(\mathbf{M}, \mathbf{y})$
$\mathbf{y}_{12} := \mathbf{M}_1 \cdot \mathbf{x}_2$		$c \leftarrow\$ \mathcal{C}$
$\mathbf{y}_{21} := \mathbf{M}_2 \cdot \mathbf{x}_1$	$\xrightarrow{\mathbf{y}_{12}, \mathbf{y}_{21}}$	$\hat{\mathbf{M}}_c := \mathbf{M}_1 + c \cdot \mathbf{M}_2$
$\hat{\mathbf{x}}_c := c \cdot \mathbf{x}_1 + \mathbf{x}_2$	$\xleftarrow{c}$	$\hat{\mathbf{y}}_c := \mathbf{y}_{12} + \mathbf{y} \cdot c + \mathbf{y}_{21} \cdot c^2 \bmod q$
	$\xrightarrow{\hat{\mathbf{x}}_c}$	**return** $\begin{cases} \hat{\mathbf{M}}_c \cdot \hat{\mathbf{x}}_c = \hat{\mathbf{y}}_c \\ \|\hat{\mathbf{x}}_c\| \approx 0 \end{cases}$

Just another SIS relation but with only $m/2$ columns $\implies$ Recursion

Lattice-based Bulletproofs

Goal: Prove SIS relation with $O(\log m)$ communication:

$$\mathbf{x} \in \mathcal{R}^m : \ \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q \wedge \|\mathbf{x}\| \approx 0$$

where $m = 2^\ell$, $\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2]$, $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$.

Prover $\mathcal{P}((\mathbf{M}, \mathbf{y}), \mathbf{x})$		Verifier $\mathcal{V}(\mathbf{M}, \mathbf{y})$
$\mathbf{y}_{12} := \mathbf{M}_1 \cdot \mathbf{x}_2$		$c \leftarrow\$ \mathcal{C}$
$\mathbf{y}_{21} := \mathbf{M}_2 \cdot \mathbf{x}_1$	$\xrightarrow{\mathbf{y}_{12}, \mathbf{y}_{21}}$	$\hat{\mathbf{M}}_c := \mathbf{M}_1 + c \cdot \mathbf{M}_2$
$\hat{\mathbf{x}}_c := c \cdot \mathbf{x}_1 + \mathbf{x}_2$	$\xleftarrow{c}$	$\hat{\mathbf{y}}_c := \mathbf{y}_{12} + \mathbf{y} \cdot c + \mathbf{y}_{21} \cdot c^2 \bmod q$
	$\xrightarrow{\hat{\mathbf{x}}_c}$	**return** $\underbrace{\begin{cases} \hat{\mathbf{M}}_c \cdot \hat{\mathbf{x}}_c = \hat{\mathbf{y}}_c \\ \|\hat{\mathbf{x}}_c\| \approx 0 \end{cases}}$

Just another SIS relation but with only $m/2$ columns $\implies$ Recursion

Lattice-based Bulletproofs

After ℓ-fold recursive composition:

Prover $\mathcal{P}((\mathbf{M}, \mathbf{y}), \mathbf{x})$		Verifier $\mathcal{V}(\mathbf{M}, \mathbf{y})$
	$\xrightarrow{\mathbf{y}_{12}^{(1)}, \mathbf{y}_{21}^{(1)}}$	
	$\xleftarrow{c_1}$	$(\hat{\mathbf{M}}_{c_1}, \hat{\mathbf{y}}_{c_1}) := \ldots$
$\vdots$		$\vdots$
	$\xrightarrow{\mathbf{y}_{12}^{(\ell)}, \mathbf{y}_{21}^{(\ell)}}$	
	$\xleftarrow{c_\mu}$	$(\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}, \hat{\mathbf{y}}_{c_1,\ldots,c_\ell}) := \ldots$
	$\xrightarrow{\hat{\mathbf{x}}_{c_1,\ldots,c_\ell}}$	**return** $\begin{cases} \hat{\mathbf{M}}_{c_1,\ldots,c_\ell} \cdot \hat{\mathbf{x}}_{c_1,\ldots,c_\ell} = \hat{\mathbf{y}}_{c_1,\ldots,c_\mu} \\ \lVert \hat{\mathbf{x}}_{c_1,\ldots,c_\ell} \rVert \approx 0 \end{cases}$

Main verifier bottleneck: Computing $\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}$. In general, this requires $\Omega_\lambda(m)$ time.

Lattice-based Bulletproofs

After ℓ-fold recursive composition:

Prover $\mathcal{P}((\mathbf{M}, \mathbf{y}), \mathbf{x})$		Verifier $\mathcal{V}(\mathbf{M}, \mathbf{y})$
	$\xrightarrow{\mathbf{y}_{12}^{(1)}, \mathbf{y}_{21}^{(1)}}$	
	$\xleftarrow{c_1}$	$(\hat{\mathbf{M}}_{c_1}, \hat{\mathbf{y}}_{c_1}) := \ldots$
$\vdots$		$\vdots$
	$\xrightarrow{\mathbf{y}_{12}^{(\ell)}, \mathbf{y}_{21}^{(\ell)}}$	
	$\xleftarrow{c_\mu}$	$(\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}, \hat{\mathbf{y}}_{c_1,\ldots,c_\ell}) := \ldots$
	$\xrightarrow{\hat{\mathbf{x}}_{c_1,\ldots,c_\ell}}$	**return** $\begin{cases} \hat{\mathbf{M}}_{c_1,\ldots,c_\ell} \cdot \hat{\mathbf{x}}_{c_1,\ldots,c_\ell} = \hat{\mathbf{y}}_{c_1,\ldots,c_\mu} \\ \|\hat{\mathbf{x}}_{c_1,\ldots,c_\ell}\| \approx 0 \end{cases}$

Main verifier bottleneck: Computing $\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}$. In general, this requires $\Omega_\lambda(m)$ time.

Structured Folding for vSIS

Core Idea

For $\mathbf{M}$ corresponding to vSIS instance, computing $\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}$ takes $\tilde{O}_\lambda(\log m) = \tilde{O}_\lambda(1)$ time.

Example for $\ell = 3$

$$
\begin{aligned}
\mathbf{M} &= \begin{pmatrix} v & v^2 & v^3 & v^4 & v^5 & v^6 & v^7 & v^8 \end{pmatrix} \\
\hat{\mathbf{M}}_{c_1} &= \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} + \begin{pmatrix} v^5 & v^6 & v^7 & v^8 \end{pmatrix} \cdot c_1 \\
&= \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} \cdot (1 + v^4 \cdot c_1) \\
\hat{\mathbf{M}}_{c_1,c_2} &= \begin{pmatrix} v & v^2 \end{pmatrix} \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \\
\hat{\mathbf{M}}_{c_1,c_2,c_3} &= v \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \cdot (1 + v \cdot c_3) \\
&= v \cdot \prod_{i=1}^{3} (1 + v^{2^{3-i}} \cdot c_i)
\end{aligned}
$$

Structured Folding for vSIS

Core Idea

For $\mathbf{M}$ corresponding to vSIS instance, computing $\hat{\mathbf{M}}_{c_1,\ldots,c_\ell}$ takes $\tilde{O}_\lambda(\log m) = \tilde{O}_\lambda(1)$ time.

Example for $\ell = 3$

$$\begin{aligned}
\mathbf{M} &= \begin{pmatrix} v & v^2 & v^3 & v^4 & v^5 & v^6 & v^7 & v^8 \end{pmatrix} \\
\hat{\mathbf{M}}_{c_1} &= \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} + \begin{pmatrix} v^5 & v^6 & v^7 & v^8 \end{pmatrix} \cdot c_1 \\
&= \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} \cdot (1 + v^4 \cdot c_1) \\
\hat{\mathbf{M}}_{c_1,c_2} &= \begin{pmatrix} v & v^2 \end{pmatrix} \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \\
\hat{\mathbf{M}}_{c_1,c_2,c_3} &= v \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \cdot (1 + v \cdot c_3) \\
&= v \cdot \prod_{i=1}^{3} (1 + v^{2^{3-i}} \cdot c_i)
\end{aligned}$$

Conclusion

- Vanishing Short Integer Solution (vSIS) assumption and commitments
- Succinct arguments for vSIS commitment openings
- Used to construct succinct arguments for NP
  - Lattice-based
  - Quasi-linear-time prover
  - Public and Polylogarithmic-time verifier (after preprocessing)
  - Transparent setup (RO instantiation)

Valerio Cini
AIT Austrian Institute of Technology