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

@Lattices Meet Hashes, Lausanne, Switzerland, 2023

Succinct Arguments

Let R be an NP relation.

- Completeness: If $(\mathsf{stmt}, \mathsf{wit}) \in R$, then $b = 1$ w.h.p.
- Soundness: If $(\mathsf{stmt}, \mathsf{wit}) \notin R$, then $b = 0$ w.h.p.
- Knowledge-soundness: If $b = 1$ w.h.p., then $\mathcal{P}$ must "know" wit such that $(\mathsf{stmt}, \mathsf{wit}) \in R$.
- Succinctness: $|m_0| + |m_1| + \ldots + |m_\mu| \ll |\mathsf{stmt}|$.
- Preprocessing: (Part of) stmt can be preprocessed by $\mathcal{V}$ before talking to $\mathcal{P}$.
- Non-interactive (NI): $\mu = 0$.

$\underline{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)}$

$\underline{\mathcal{P}(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})}$ $\qquad$ $\underline{\mathcal{V}(\mathsf{pp}, \mathsf{stmt})}$

$$\xrightarrow{m_0}$$

$$\xleftarrow{c_1}$$

$$\xrightarrow{m_1}$$

$$\vdots \qquad\qquad \vdots$$

$$\xleftarrow{c_\mu}$$

$$\xrightarrow{m_\mu}$$

return b

Succinct Arguments

Let R be an NP relation.

- † Completeness: If $(\text{stmt}, \text{wit}) \in R$, then $b = 1$ w.h.p.
- † Soundness: If $(\text{stmt}, \text{wit}) \notin R$, then $b = 0$ w.h.p.
- † Knowledge-soundness: If $b = 1$ w.h.p., then $\mathcal{P}$ must "know" wit such that $(\text{stmt}, \text{wit}) \in R$.
- † Succinctness: $|m_0| + |m_1| + \ldots + |m_\mu| \ll |\text{stmt}|$.
- † Preprocessing: (Part of) stmt can be preprocessed by $\mathcal{V}$ before talking to $\mathcal{P}$.
- † Non-interactive (NI): $\mu = 0$.

$\underline{\text{pp} \leftarrow \text{Setup}(1^\lambda)}$

$\underline{\mathcal{P}(\text{pp}, \text{stmt}, \text{wit})}$ $\qquad\qquad$ $\underline{\mathcal{V}(\text{pp}, \text{stmt})}$

$\xrightarrow{m_0}$

$\xleftarrow{c_1}$

$\xrightarrow{m_1}$

$\vdots \qquad\qquad \vdots$

$\xleftarrow{c_\mu}$

$\xrightarrow{m_\mu}$ $\qquad$ **return** b

Succinct Arguments

Let R be an NP relation.

- † Completeness: If $(\mathsf{stmt}, \mathsf{wit}) \in R$, then $b = 1$ w.h.p.
- † Soundness: If $(\mathsf{stmt}, \mathsf{wit}) \notin R$, then $b = 0$ w.h.p.
- † Knowledge-soundness: If $b = 1$ w.h.p., then $\mathcal{P}$ must "know" wit such that $(\mathsf{stmt}, \mathsf{wit}) \in R$.
- † Succinctness: $|m_0| + |m_1| + \ldots + |m_\mu| \ll |\mathsf{stmt}|$.
- † Preprocessing: (Part of) stmt can be preprocessed by $\mathcal{V}$ before talking to $\mathcal{P}$.
- † Non-interactive (NI): $\mu = 0$.

$\underline{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)}$

$\underline{\mathcal{P}(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})}$		$\underline{\mathcal{V}(\mathsf{pp}, \mathsf{stmt})}$
	$\xrightarrow{m_0}$	
	$\xleftarrow{c_1}$	
	$\xrightarrow{m_1}$	
$\vdots$		$\vdots$
	$\xleftarrow{c_\mu}$	
	$\xrightarrow{m_\mu}$	**return** b

Succinct Arguments

Let R be an NP relation.

- † Completeness: If $(\text{stmt}, \text{wit}) \in R$, then $b = 1$ w.h.p.
- † Soundness: If $(\text{stmt}, \text{wit}) \notin R$, then $b = 0$ w.h.p.
- † Knowledge-soundness: If $b = 1$ w.h.p., then $\mathcal{P}$ must "know" wit such that $(\text{stmt}, \text{wit}) \in R$.
- † Succinctness: $|m_0| + |m_1| + \ldots + |m_\mu| \ll |\text{stmt}|$.
- † Preprocessing: (Part of) stmt can be preprocessed by $\mathcal{V}$ before talking to $\mathcal{P}$.
- † Non-interactive (NI): $\mu = 0$.

$\underline{\text{pp} \leftarrow \text{Setup}(1^\lambda)}$

$\underline{\mathcal{P}(\text{pp}, \text{stmt}, \text{wit})}$		$\underline{\mathcal{V}(\text{pp}, \text{stmt})}$
	$\xrightarrow{\quad m_0 \quad}$	
	$\xleftarrow{\quad c_1 \quad}$	
	$\xrightarrow{\quad m_1 \quad}$	
$\vdots$		$\vdots$
	$\xleftarrow{\quad c_\mu \quad}$	
	$\xrightarrow{\quad m_\mu \quad}$	**return** b

Succinct Arguments

Let R be an NP relation.

- † Completeness: If $(\text{stmt}, \text{wit}) \in R$, then $b = 1$ w.h.p.
- † Soundness: If $(\text{stmt}, \text{wit}) \notin R$, then $b = 0$ w.h.p.
- † Knowledge-soundness: If $b = 1$ w.h.p., then $\mathcal{P}$ must "know" wit such that $(\text{stmt}, \text{wit}) \in R$.
- † Succinctness: $|m_0| + |m_1| + \ldots + |m_\mu| \ll |\text{stmt}|$.
- † Preprocessing: (Part of) stmt can be preprocessed by $\mathcal{V}$ before talking to $\mathcal{P}$.
- † Non-interactive (NI): $\mu = 0$.

$\underline{\text{pp} \leftarrow \text{Setup}(1^\lambda)}$

$\underline{\mathcal{P}(\text{pp}, \text{stmt}, \text{wit})}$		$\underline{\mathcal{V}(\text{pp}, \text{stmt})}$
	$\xrightarrow{\quad m_0 \quad}$	
	$\xleftarrow{\quad c_1 \quad}$	
	$\xrightarrow{\quad m_1 \quad}$	
$\vdots$		$\vdots$
	$\xleftarrow{\quad c_\mu \quad}$	
	$\xrightarrow{\quad m_\mu \quad}$	**return** b

Lattice-based Succinct Arguments

Approach	Publicly verifiable	$\tilde{O}_\lambda(1)$-verifier (preprocessing)	$\tilde{O}_\lambda(\lvert\text{stmt}\rvert)$-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]	✓	✓	✘ $\tilde{O}_\lambda(\lvert\text{stmt}\rvert^2)$
Direct (this work)	✓	✓	✓

Lattice-based Succinct Arguments

Approach	Publicly verifiable	$\tilde{O}_\lambda(1)$-verifier (preprocessing)	$\tilde{O}_\lambda(\lvert \mathrm{stmt} \rvert)$-prover
PCP/IOP + linear-only enc. [BCIOP13; BISW17; BISW18; GMNO18]	✘	✓	✓
Linearisation + folding [BLNS20; AL21; ACK21; BS22]	✓	✘ $\tilde{O}_\lambda(\lvert \mathrm{stmt} \rvert)$	✓
Direct [ACLMT22]	✓	✓	✘ $\tilde{O}_\lambda(\lvert \mathrm{stmt} \rvert^2)$
Direct (this work)	✓	✓	✓

Our Results

- New assumption: Vanishing Short Integer Solution (vSIS)
  - Implied by kRISIS assumption [ACLMT22]
  - Implies kRISIS assumption conditioned on knowledge-kRISIS assumption [ACLMT22]
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small ($\tilde{O}_\lambda(1)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}_\lambda(|\mathsf{stmt}|)$-prover $\tilde{O}_\lambda(1)$-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)
  - Implied by kRISIS assumption [ACLMT22]
  - Implies kRISIS assumption conditioned on knowledge-kRISIS assumption [ACLMT22]
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small ($\tilde{O}_\lambda(1)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}_\lambda(|\mathsf{stmt}|)$-prover $\tilde{O}_\lambda(1)$-verifier arguments for commitment openings
- New lattice-based succinct arguments for NP $\Longleftarrow$ Succinct arguments for vSIS commitment openings

Our Results

- New assumption: Vanishing Short Integer Solution (vSIS)
  - Implied by kRISIS assumption [ACLMT22]
  - Implies kRISIS assumption conditioned on knowledge-kRISIS assumption [ACLMT22]
- New tool: vSIS commitment for committing to polynomials with short coefficients
  - Very small ($\tilde{O}_\lambda(1)$) commitment key
  - (Almost) additively and *multiplicatively* homomorphic
  - Admit $\tilde{O}_\lambda(|\mathsf{stmt}|)$-prover $\tilde{O}_\lambda(1)$-verifier arguments for commitment openings
- New lattice-based succinct arguments for NP $\Leftarrow$ Succinct arguments for vSIS commitment openings

Our Results

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

Roadmap

1. Preliminaries
2. vSIS assumptions and commitments
3. Succinct arguments for vSIS commitment openings
4. Succinct arguments for NP

Number Rings

- Everything we discuss will be over a cyclotomic ring $\mathcal{R} = \mathbb{Z}[\zeta]$.
- For intuition, it is mostly okay to treat $\mathcal{R} = \mathbb{Z}$.

- Quotient ring: For $q \in \mathbb{N}$, $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$.
- Units: Denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ sets of units (invertible elements) $\mathcal{R}$ and $\mathcal{R}_q$ respectively.
- We assume $1/|\mathcal{R}_q^\times| = \mathsf{negl}(\lambda)$.
- Norm: For $a \in \mathcal{R}$, $\|a\|$ is some (geometric) norm, e.g. the ∞-norm.

Number Rings

- † Everything we discuss will be over a cyclotomic ring $\mathcal{R} = \mathbb{Z}[\zeta]$.
- † For intuition, it is mostly okay to treat $\mathcal{R} = \mathbb{Z}$.

- † Quotient ring: For $q \in \mathbb{N}$, $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$.
- † Units: Denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ sets of units (invertible elements) $\mathcal{R}$ and $\mathcal{R}_q$ respectively.
- † We assume $1/|\mathcal{R}_q^\times| = \mathsf{negl}(\lambda)$.
- † Norm: For $a \in \mathcal{R}$, $\|a\|$ is some (geometric) norm, e.g. the ∞-norm.

Matrix and Vector Notation

- † Matrix and vector are bold upper and lower case: **M** and **v**.
- † We usually don't distinguish between row and column vectors.
- † When we do, we write transpose, e.g. $\mathbf{v}^\top$, for row vectors.
- † Let $\mathbf{a} = (a_1, \ldots, a_m), \mathbf{b} = (b_1, \ldots, b_m)$ be vectors.
- † Inner product: $\langle \mathbf{a}, \mathbf{b} \rangle := \sum_{i=1}^{m} a_i \cdot b_i$.
- † Hadamard product: $\mathbf{a} \circ \mathbf{b} := (a_i \cdot b_i)_{i=1}^{m}$.

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

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

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

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 1: Univariate

- Parameters: Class of univariate degree-m polynomials, modulus q.
- Instance: A unit $v \in \mathcal{R}_q^\times$.
- 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 modulo q, i.e.

$$p(v) = 0 \bmod q \qquad \text{and} \qquad 0 < \|p\| := \max_{i \in [m]} \|p_i\| \approx 0.$$

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

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

Vanishing Short Integer Solution (vSIS) Assumption

Example 2: Univariate Laurent

- Parameters: Class of univariate "degree-m" Laurent polynomials, modulus q.
- Instance: A unit $v \in \mathcal{R}_q^\times$.
- Problem: Find short "degree m" Laurent polynomial without constant term

$$p(X) = p_{-m}X^{-m} + \ldots + p_{-1}X^{-1} + p_1X + \ldots + p_mX^m \in \mathcal{R}[X, X^{-1}]$$

which vanishes at v modulo q.

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:

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

- 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:

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

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

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

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

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

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

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):

$$\begin{aligned} p(v) + p'(v) &= (p + p')(v) \bmod q & \left\|p + p'\right\| &\leq \|p\| + \left\|p'\right\| \approx 0 \\ p(v) \cdot p'(v) &= (p \cdot p')(v) \bmod q & \left\|p \cdot p'\right\| &\lessapprox \|p\| \cdot \left\|p'\right\| \approx 0. \end{aligned}$$

Encoding Vectors as (Laurent) Polynomials

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

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

$$\mathbf{c} := (c_{-m}, \ldots, c_{-1}, c_0, c_1, \ldots, c_m) \in \mathcal{R}^{2m+1}$$

$$\hat{p}_{\mathbf{c}}(X) := c_{-m} X^{-m} + \ldots + c_{-1} X^{-1} + c_0 + c_1 X + c_2 X^2 + \ldots + c_m X^m$$

Note that

$$\bar{p}_{\mathbf{a}}(X) \cdot p_{\mathbf{b}}(X) = \hat{p}_{\mathbf{a}*\mathbf{b}}(X),$$

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 $\langle \mathbf{a}, \mathbf{b} \rangle$.

If $\langle \mathbf{a}, \mathbf{b} \rangle = c_0$, then $\hat{p}_{\mathbf{a}*\mathbf{b}-\mathbf{c}}$ has no constant term.

Encoding Vectors as (Laurent) Polynomials

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

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

$$\mathbf{c} := (c_{-m}, \ldots, c_{-1}, c_0, c_1, \ldots, c_m) \in \mathcal{R}^{2m+1}$$

$$\hat{p}_{\mathbf{c}}(X) := c_{-m} X^{-m} + \ldots + c_{-1} X^{-1} + c_0 + c_1 X + c_2 X^2 + \ldots + c_m X^m$$

Note that

$$\bar{p}_{\mathbf{a}}(X) \cdot p_{\mathbf{b}}(X) = \hat{p}_{\mathbf{a}*\mathbf{b}}(X),$$

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 $\langle \mathbf{a}, \mathbf{b} \rangle$.

If $\langle \mathbf{a}, \mathbf{b} \rangle = c_0$, then $\hat{p}_{\mathbf{a}*\mathbf{b}-\mathbf{c}}$ has no constant term.

Terminologies for Moving Forward

- Dual vSIS commitment of $\mathbf{a} \in \mathcal{R}^m$:

$$\bar{c}_{\mathbf{a}} = \bar{p}_{\mathbf{a}}(v) = a_1 v^{-1} + \ldots + a_m v^{-m} \bmod q$$

- (Primal) vSIS commitment of $\mathbf{b} \in \mathcal{R}^m$:

$$c_{\mathbf{b}} = p_{\mathbf{b}}(v) = b_1 v + \ldots + b_m v^m \bmod q$$

- Balanced vSIS commitment of $\mathbf{c} \in \mathcal{R}^{2m+1}$:

$$\hat{c}_{\mathbf{c}} = \hat{p}_{\mathbf{c}}(v) = c_{-m} v^{-m} + \ldots + c_{-1} v^{-1} + c_0 + c_1 v + c_2 v^2 + \ldots + c_m X^m \bmod q$$

A Taste of Applications

Suppose

- † $\mathbf{a}$ is committed in dual vSIS commitment as $\bar{c}_{\mathbf{a}} := \bar{p}_{\mathbf{a}}(v)$,
- † $\mathbf{b}$ is committed in vSIS commitment as $c_{\mathbf{b}} := p_{\mathbf{b}}(v)$, and
- † c is some given value.

To succinctly prove that $\langle \mathbf{a}, \mathbf{b} \rangle = c$:

- † Prove that $\bar{c}_{\mathbf{a}}$ is a dual vSIS commitment.
- † Prove that $c_{\mathbf{b}}$ is a vSIS commitment.
- † Prove that $\bar{c}_{\mathbf{a}} \cdot c_{\mathbf{b}} - c$ is a balanced vSIS commitment of a polynomial without constant term.

A Taste of Applications

Suppose

- † $\mathbf{a}$ is committed in dual vSIS commitment as $\bar{c}_{\mathbf{a}} := \bar{p}_{\mathbf{a}}(v)$,
- † $\mathbf{b}$ is committed in vSIS commitment as $c_{\mathbf{b}} := p_{\mathbf{b}}(v)$, and
- † c is some given value.

To succinctly prove that $\langle \mathbf{a}, \mathbf{b} \rangle = c$:

- † Prove that $\bar{c}_{\mathbf{a}}$ is a dual vSIS commitment.
- † Prove that $c_{\mathbf{b}}$ is a vSIS commitment.
- † Prove that $\bar{c}_{\mathbf{a}} \cdot c_{\mathbf{b}} - c$ is a balanced vSIS commitment of a polynomial without constant term.

Coming up

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

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

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

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

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

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

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

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

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

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

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

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{T}, v, (\mathbf{u}_i)_{i=1}^{m})$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}\left(\mathbf{T} \cdot \begin{pmatrix} v^i \\ v^{-i} \end{pmatrix}\right).$$

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

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{T}, v, (\mathbf{u}_i)_{i=1}^m)$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}\left(\mathbf{T} \cdot \begin{pmatrix} v^i \\ v^{-i} \end{pmatrix}\right).$$

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

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Public parameters: kRISIS instance $(\mathbf{A}, \mathbf{T}, v, (\mathbf{u}_i)_{i=1}^m)$ where

$$\mathbf{u}_i \in \mathbf{A}^{-1}\left(\mathbf{T} \cdot \begin{pmatrix} v^i \\ v^{-i} \end{pmatrix}\right).$$

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

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- 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 $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot x_i$.
- 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 clearly runs in $\tilde{O}_\lambda(m)$ time.
- Verifier clearly runs in $\tilde{O}_\lambda(1)$ time.

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- 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 $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot x_i$.
- 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 clearly runs in $\tilde{O}_\lambda(m)$ time.
- Verifier clearly runs in $\tilde{O}_\lambda(1)$ time.

Succinct Argument for vSIS Commitment (Knowledge-kRISIS)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- 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 $\mathbf{u} = \sum_{i \in \pm[m]} \mathbf{u}_i \cdot x_i$.
- 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 clearly runs in $\tilde{O}_\lambda(m)$ time.
- Verifier clearly runs in $\tilde{O}_\lambda(1)$ time.

Crash Course on (Lattice-based) Bulletproofs

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

$$\left\{(\mathbf{A}, \mathbf{y}) \in \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^n : \exists\, \mathbf{x} \in \mathcal{R}^m,\ \mathbf{A}\cdot\mathbf{x} = \mathbf{y} \bmod q \wedge \|\mathbf{x}\| \approx 0\right\}$$

where $m = 2^\mu$, $\mathbf{A} = (\mathbf{A}_1 \mid \mathbf{A}_2)$, $\mathbf{x} = (\mathbf{x}_1 \mid \mathbf{x}_2)$.

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

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

Crash Course on (Lattice-based) Bulletproofs

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

$$\left\{(\mathbf{A}, \mathbf{y}) \in \mathcal{R}_q^{n \times m} \times \mathcal{R}_q^n : \exists\, \mathbf{x} \in \mathcal{R}^m,\ \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q \wedge \|\mathbf{x}\| \approx 0\right\}$$

where $m = 2^\mu$, $\mathbf{A} = (\mathbf{A}_1 \mid \mathbf{A}_2)$, $\mathbf{x} = (\mathbf{x}_1 \mid \mathbf{x}_2)$.

$$\begin{array}{lcl}
\underline{\text{Prover } \mathcal{P}((\mathbf{A}, \mathbf{y}), \mathbf{x})} & & \underline{\text{Verifier } \mathcal{V}(\mathbf{A}, \mathbf{y})} \\
\mathbf{y}_{12} := \mathbf{A}_1 \cdot \mathbf{x}_2 & & c \leftarrow\!\!\$\, \mathcal{C} \\
\mathbf{y}_{21} := \mathbf{A}_2 \cdot \mathbf{x}_1 & \xrightarrow{\quad \mathbf{y}_{12}, \mathbf{y}_{21} \quad} & \hat{\mathbf{A}}_c := \mathbf{A}_1 + c \cdot \mathbf{A}_2 \\
\hat{\mathbf{x}}_c := c \cdot \mathbf{x}_1 + \mathbf{x}_2 & \xleftarrow{\quad c \quad} & \hat{\mathbf{y}}_c := \mathbf{y}_{12} + \mathbf{y} \cdot c + \mathbf{y}_{21} \cdot c^2 \bmod q \\
 & \xrightarrow{\quad \hat{\mathbf{x}}_c \quad} & \textbf{return } \underbrace{\begin{cases} \hat{\mathbf{A}}_c \cdot \hat{\mathbf{x}}_c = \hat{\mathbf{y}}_c \\ \|\hat{\mathbf{x}}_c\| \approx 0 \end{cases}}
\end{array}$$

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

Crash Course on (Lattice-based) Bulletproofs

After μ-fold recursive composition:

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

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

Crash Course on (Lattice-based) Bulletproofs

After μ-fold recursive composition:

$$
\begin{array}{lcl}
\underline{\text{Prover } \mathcal{P}((\mathbf{A},\mathbf{y}),\mathbf{x})} & & \underline{\text{Verifier } \mathcal{V}(\mathbf{A},\mathbf{y})} \\
 & \xrightarrow{\mathbf{y}_{12}^{(1)},\, \mathbf{y}_{21}^{(1)}} & \\
 & \xleftarrow{c_1} & (\hat{\mathbf{A}}_{c_1}, \hat{\mathbf{y}}_{c_1}) := \ldots \\
\vdots & & \vdots \\
 & \xrightarrow{\mathbf{y}_{12}^{(\mu)},\, \mathbf{y}_{21}^{(\mu)}} & \\
 & \xleftarrow{c_\mu} & (\hat{\mathbf{A}}_{c_1,\ldots,c_\mu}, \hat{\mathbf{y}}_{c_1,\ldots,c_\mu}) := \ldots \\
 & \xrightarrow{\hat{\mathbf{x}}_{c_1,\ldots,c_\mu}} & \textbf{return} \begin{cases} \hat{\mathbf{A}}_{c_1,\ldots,c_\mu} \cdot \hat{\mathbf{x}}_{c_1,\ldots,c_\mu} = \hat{\mathbf{y}}_{c_1,\ldots,c_\mu} \\ \left\| \hat{\mathbf{x}}_{c_1,\ldots,c_\mu} \right\| \approx 0 \end{cases}
\end{array}
$$

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

Structured Folding for vSIS

Core Idea

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

Example for $\mu = 3$

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

Structured Folding for vSIS

Core Idea

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

Example for $\mu = 3$

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

Succinct Argument for vSIS Commitment (Folding)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

- After folding:

$$\hat{\mathbf{A}}_{c_1,\dots,c_\mu} = \begin{pmatrix} v \cdot \prod_{i=1}^{\mu}(1 + v^{2^{\mu-i}} \cdot c_i) \\ v^{-1} \cdot \prod_{i=1}^{\mu}(1 + v^{-2^{\mu-i}} \cdot c_i) \end{pmatrix}$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_1,\dots,c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

- After folding:

$$\hat{\mathbf{A}}_{c_1,\ldots,c_\mu} = \begin{pmatrix} v \cdot \prod_{i=1}^{\mu}(1 + v^{2^{\mu-i}} \cdot c_i) \\ v^{-1} \cdot \prod_{i=1}^{\mu}(1 + v^{-2^{\mu-i}} \cdot c_i) \end{pmatrix}$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_1,\ldots,c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

- After folding:

$$\hat{\mathbf{A}}_{c_1,\dots,c_\mu} = \begin{pmatrix} v \cdot \prod_{i=1}^{\mu}(1 + v^{2^{\mu-i}} \cdot c_i) \\ v^{-1} \cdot \prod_{i=1}^{\mu}(1 + v^{-2^{\mu-i}} \cdot c_i) \end{pmatrix}$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_1,\dots,c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

Want to prove $(c, \bar{c})$ and $\mathbf{x} \in \mathcal{R}^m$ satisfies:

$$c = p_{\mathbf{x}}(v) \qquad \bar{c} = p_{\mathbf{x}}(v^{-1}) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v & v^2 & \dots & v^m \\ v^{-1} & v^{-2} & \dots & v^{-m} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

- After folding:

$$\hat{\mathbf{A}}_{c_1,\dots,c_\mu} = \begin{pmatrix} v \cdot \prod_{i=1}^{\mu}(1 + v^{2^{\mu-i}} \cdot c_i) \\ v^{-1} \cdot \prod_{i=1}^{\mu}(1 + v^{-2^{\mu-i}} \cdot c_i) \end{pmatrix}$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_1,\dots,c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v^{-m} & \dots & v^{-1} & v & \dots & v^{m} \end{pmatrix} \qquad \mathbf{y} = \hat{c}$$

- After folding:

$$\hat{\mathbf{A}}_{c_0,c_1,\dots,c_\mu} = v \cdot \prod_{i=1}^{\mu} (1 + v^{2^{\mu-i}} \cdot c_i) \cdot (v^{-m-1} + c_0)$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_0,c_1,\dots,c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v^{-m} & \dots & v^{-1} & v & \dots & v^{m} \end{pmatrix} \qquad \mathbf{y} = \hat{c}$$

- After folding:

$$\hat{\mathbf{A}}_{c_0, c_1, \dots, c_\mu} = v \cdot \prod_{i=1}^{\mu} (1 + v^{2^{\mu-i}} \cdot c_i) \cdot (v^{-m-1} + c_0)$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_0, c_1, \dots, c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v^{-m} & \ldots & v^{-1} & v & \ldots & v^{m} \end{pmatrix} \qquad \mathbf{y} = \hat{c}$$

- After folding:

$$\hat{\mathbf{A}}_{c_0, c_1, \ldots, c_\mu} = v \cdot \prod_{i=1}^{\mu} (1 + v^{2^{\mu - i}} \cdot c_i) \cdot (v^{-m-1} + c_0)$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_0, c_1, \ldots, c_\mu}$ consists of product of $O(\log m)$ sums).

Succinct Argument for vSIS Commitment (Folding)

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

$$x_0 = 0 \qquad \hat{c} = \hat{p}_{\mathbf{x}}(v) \qquad \|\mathbf{x}\| \approx 0.$$

- Equivalent to proving $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$ where

$$\mathbf{A} = \begin{pmatrix} v^{-m} & \dots & v^{-1} & v & \dots & v^{m} \end{pmatrix} \qquad \mathbf{y} = \hat{c}$$

- After folding:

$$\hat{\mathbf{A}}_{c_0,c_1,\dots,c_\mu} = v \cdot \prod_{i=1}^{\mu} (1 + v^{2^{\mu-i}} \cdot c_i) \cdot (v^{-m-1} + c_0)$$

- Knowledge-soundness follows from existing Bulletproofs analysis.
- Prover runs in $\tilde{O}_\lambda(m)$ time.
- Verifier runs in $\tilde{O}_\lambda(1)$ time (since $\hat{\mathbf{A}}_{c_0,c_1,\dots,c_\mu}$ consists of product of $O(\log m)$ sums).

Two NP-Complete Examples

1. Subset Sum
2. Rank-1 Constraint Satisfiability (R1CS)

Subset Sum

$$\{(\mathbf{M}, \mathbf{y}) : \exists\, \mathbf{x} \in \{0,1\}^m,\ \mathbf{M} \cdot \mathbf{x} = \mathbf{y}\}$$

- † Close connection to SIS (modular reduction, binariness $\rightarrow$ bounded-norm).
- † "Almost linear", i.e. $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$, while $\left(\mathbf{x} \in \{0,1\}^m\right) \iff \left((\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}\right)$.

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary (High Level Idea)

† Verifier preprocesses $(\mathbf{M}, \mathbf{y})$ by computing their vSIS commitments.

† Prover vSIS-commits to the witness $\mathbf{x}$ and some auxiliary witness $\mathbf{x}'$.

† Using the commitments of $\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{x}'$, the verifier homomorphically derive vSIS commitments of polynomials where the constant terms encode

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x}.$$

† Prover proves that these committed polynomials have no constant terms, i.e.

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} = \mathbf{0} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}.$$

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary (High Level Idea)

- Verifier preprocesses $(\mathbf{M}, \mathbf{y})$ by computing their vSIS commitments.
- Prover vSIS-commits to the witness $\mathbf{x}$ and some auxiliary witness $\mathbf{x}'$.
- Using the commitments of $\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{x}'$, the verifier homomorphically derive vSIS commitments of polynomials where the constant terms encode

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x}.$$

- Prover proves that these committed polynomials have no constant terms, i.e.

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} = \mathbf{0} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}.$$

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary (High Level Idea)

- Verifier preprocesses $(\mathbf{M}, \mathbf{y})$ by computing their vSIS commitments.
- Prover vSIS-commits to the witness $\mathbf{x}$ and some auxiliary witness $\mathbf{x}'$.
- Using the commitments of $\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{x}'$, the verifier homomorphically derive vSIS commitments of polynomials where the constant terms encode

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x}.$$

- Prover proves that these committed polynomials have no constant terms, i.e.

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} = \mathbf{0} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}.$$

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary (High Level Idea)

- Verifier preprocesses $(\mathbf{M}, \mathbf{y})$ by computing their vSIS commitments.
- Prover vSIS-commits to the witness $\mathbf{x}$ and some auxiliary witness $\mathbf{x}'$.
- Using the commitments of $\mathbf{M}, \mathbf{y}, \mathbf{x}, \mathbf{x}'$, the verifier homomorphically derive vSIS commitments of polynomials where the constant terms encode

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x}.$$

- Prover proves that these committed polynomials have no constant terms, i.e.

$$\mathbf{M} \cdot \mathbf{x} - \mathbf{y} = \mathbf{0} \qquad \text{and} \qquad (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}.$$

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary

Let $\mathbf{h}, \mathbf{k}, \mathbf{l}$ be random vectors with $0 \ll \|\mathbf{h}\|, \|\mathbf{k}\| \ll \|\mathbf{l}\| \ll q$.

Prover commits to and proves well-formedness of the following

Witness and Auxiliaries	$\mathbf{x}$	$\mathbf{x}' = \mathbf{k} \circ \mathbf{x}$
Commitment	$(p_{\mathbf{x}}(v), p_{\mathbf{x}}(v^{-1}))$	$p_{\mathbf{x}'}(v^{-1})$

Prover proves that the following are commitments to short Laurent polynomials without constant term:

	Commitment	Constant term and implication
1.	$p_{\mathbf{h}^\top \cdot \mathbf{M}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - \mathbf{h}^\top \cdot \mathbf{y}$	$\mathbf{h}^\top \cdot (\mathbf{M} \cdot \mathbf{x} - \mathbf{y}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{M} \cdot \mathbf{x} = \mathbf{y}$
2.	$p_{\mathbf{x}'}(v^{-1}) \cdot p_{\mathbf{l}}(v) - p_{\mathbf{l} \circ \mathbf{k}}(v^{-1}) \cdot p_{\mathbf{x}}(v)$	$\mathbf{l}^\top \cdot (\mathbf{x}' - \mathbf{k} \circ \mathbf{x}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{x}' = \mathbf{k} \circ \mathbf{x}$
3.	$(p_{\mathbf{x}'}(v^{-1}) - p_{\mathbf{k}}(v^{-1})) \cdot p_{\mathbf{x}}(v)$	$\underbrace{\mathbf{k}^\top \cdot ((\mathbf{x} - \mathbf{1}) \circ \mathbf{x}) = 0}_{2. \Longrightarrow} \overset{\text{SIS}}{\Longrightarrow} (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}$

Preprocessed Sent by prover

Proving $\mathbf{M} \cdot \mathbf{x} = \mathbf{y}$ and $\mathbf{x}$ Binary

Let $\mathbf{h}, \mathbf{k}, \mathbf{l}$ be random vectors with $0 \ll \|\mathbf{h}\|, \|\mathbf{k}\| \ll \|\mathbf{l}\| \ll q$.

Prover commits to and proves well-formedness of the following

Witness and Auxiliaries	$\mathbf{x}$	$\mathbf{x}' = \mathbf{k} \circ \mathbf{x}$
Commitment	$(p_{\mathbf{x}}(v), p_{\mathbf{x}}(v^{-1}))$	$p_{\mathbf{x}'}(v^{-1})$

Prover proves that the following are commitments to short Laurent polynomials without constant term:

	Commitment	Constant term and implication
1.	$p_{\mathbf{h}^\mathsf{T} \cdot \mathbf{M}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - \mathbf{h}^\mathsf{T} \cdot \mathbf{y}$	$\mathbf{h}^\mathsf{T} \cdot (\mathbf{M} \cdot \mathbf{x} - \mathbf{y}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{M} \cdot \mathbf{x} = \mathbf{y}$
2.	$p_{\mathbf{x}'}(v^{-1}) \cdot p_{\mathbf{l}}(v) - p_{\mathbf{l} \circ \mathbf{k}}(v^{-1}) \cdot p_{\mathbf{x}}(v)$	$\mathbf{l}^\mathsf{T} \cdot (\mathbf{x}' - \mathbf{k} \circ \mathbf{x}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{x}' = \mathbf{k} \circ \mathbf{x}$
3.	$(p_{\mathbf{x}'}(v^{-1}) - p_{\mathbf{k}}(v^{-1})) \cdot p_{\mathbf{x}}(v)$	$\underbrace{\mathbf{k}^\mathsf{T} \cdot ((\mathbf{x} - \mathbf{1}) \circ \mathbf{x}) = 0}_{2. \Longrightarrow} \overset{\text{SIS}}{\Longrightarrow} (\mathbf{x} - \mathbf{1}) \circ \mathbf{x} = \mathbf{0}$

Preprocessed   Sent by prover

Rank-1 Constraint Satisfiability (R1CS)

$$\{(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{y}) : \exists\, \mathbf{x},\ (\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x}) \wedge \mathbf{D} \cdot \mathbf{x} = \mathbf{y}\}$$

† The boundary constraint

$$\mathbf{D} \cdot \mathbf{x} = \mathbf{y}$$

is easy to deal with. In next slide, we ignore it and focus on

$$(\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x}).$$

Proving $(\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x})$

Let $\mathbf{h}, \mathbf{l}$ be random vectors with $0 \ll \|\mathbf{h}\| \ll \|\mathbf{l}\| \ll q$.

Prover commits to and proves well-formedness of the following

Witness and Auxiliaries	$\mathbf{x}$	$\mathbf{a} = \mathbf{A} \cdot \mathbf{x}$	$\mathbf{b} = \mathbf{B} \cdot \mathbf{x}$	$\mathbf{c} = \mathbf{C} \cdot \mathbf{x}$	$\mathbf{a}' = \mathbf{h} \circ \mathbf{a}$
Commitment	$(p_{\mathbf{x}}(v), p_{\mathbf{x}}(v^{-1}))$	$p_{\mathbf{a}}(v)$	$p_{\mathbf{b}}(v)$	$p_{\mathbf{c}}(v)$	$p_{\mathbf{a}'}(v^{-1})$

Prover proves that the following are commitments to short Laurent polynomials without constant term:

	Commitment	Constant term and implication
1.	$p_{\mathbf{h}^\top \cdot \mathbf{A}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{a}}(v)$	$\mathbf{h}^\top \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{a}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{A} \cdot \mathbf{x} = \mathbf{a}$
2.	$p_{\mathbf{h}^\top \cdot \mathbf{B}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{b}}(v)$	$\mathbf{h}^\top \cdot (\mathbf{B} \cdot \mathbf{x} - \mathbf{b}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{B} \cdot \mathbf{x} = \mathbf{b}$
3.	$p_{\mathbf{h}^\top \cdot \mathbf{C}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{c}}(v)$	$\mathbf{h}^\top \cdot (\mathbf{C} \cdot \mathbf{x} - \mathbf{c}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{C} \cdot \mathbf{x} = \mathbf{c}$
4.	$p_{\mathbf{a}'}(v^{-1}) \cdot p_{\mathbf{l}}(v) - p_{\mathbf{l} \circ \mathbf{h}}(v^{-1}) \cdot p_{\mathbf{a}}(v)$	$\mathbf{l}^\top \cdot (\mathbf{a}' - \mathbf{h} \circ \mathbf{a}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{a}' = \mathbf{h} \circ \mathbf{a}$
5.	$p_{\mathbf{a}'}(v^{-1}) \cdot p_{\mathbf{b}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{c}}(v)$	$\underbrace{\mathbf{h}^\top \cdot (\mathbf{a} \circ \mathbf{b} - \mathbf{c}) = 0}_{4. \Longrightarrow} \overset{\text{SIS}}{\Longrightarrow} \mathbf{a} \circ \mathbf{b} = \mathbf{c}$

Preprocessed Sent by prover

Proving $(\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x})$

Let $\mathbf{h}, \mathbf{l}$ be random vectors with $0 \ll \|\mathbf{h}\| \ll \|\mathbf{l}\| \ll q$.

Prover commits to and proves well-formedness of the following

Witness and Auxiliaries	$\mathbf{x}$	$\mathbf{a} = \mathbf{A} \cdot \mathbf{x}$	$\mathbf{b} = \mathbf{B} \cdot \mathbf{x}$	$\mathbf{c} = \mathbf{C} \cdot \mathbf{x}$	$\mathbf{a}' = \mathbf{h} \circ \mathbf{a}$
Commitment	$(p_{\mathbf{x}}(v), p_{\mathbf{x}}(v^{-1}))$	$p_{\mathbf{a}}(v)$	$p_{\mathbf{b}}(v)$	$p_{\mathbf{c}}(v)$	$p_{\mathbf{a}'}(v^{-1})$

Prover proves that the following are commitments to short Laurent polynomials without constant term:

	Commitment	Constant term and implication
1.	$p_{\mathbf{h}^\mathsf{T} \cdot \mathbf{A}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{a}}(v)$	$\mathbf{h}^\mathsf{T} \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{a}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{A} \cdot \mathbf{x} = \mathbf{a}$
2.	$p_{\mathbf{h}^\mathsf{T} \cdot \mathbf{B}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{b}}(v)$	$\mathbf{h}^\mathsf{T} \cdot (\mathbf{B} \cdot \mathbf{x} - \mathbf{b}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{B} \cdot \mathbf{x} = \mathbf{b}$
3.	$p_{\mathbf{h}^\mathsf{T} \cdot \mathbf{C}}(v^{-1}) \cdot p_{\mathbf{x}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{c}}(v)$	$\mathbf{h}^\mathsf{T} \cdot (\mathbf{C} \cdot \mathbf{x} - \mathbf{c}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{C} \cdot \mathbf{x} = \mathbf{c}$
4.	$p_{\mathbf{a}'}(v^{-1}) \cdot p_{\mathbf{l}}(v) - p_{\mathbf{l} \circ \mathbf{h}}(v^{-1}) \cdot p_{\mathbf{a}}(v)$	$\mathbf{l}^\mathsf{T} \cdot (\mathbf{a}' - \mathbf{h} \circ \mathbf{a}) = 0 \overset{\text{SIS}}{\Longrightarrow} \mathbf{a}' = \mathbf{h} \circ \mathbf{a}$
5.	$p_{\mathbf{a}'}(v^{-1}) \cdot p_{\mathbf{b}}(v) - p_{\mathbf{h}}(v^{-1}) \cdot p_{\mathbf{c}}(v)$	$\underbrace{\mathbf{h}^\mathsf{T} \cdot (\mathbf{a} \circ \mathbf{b} - \mathbf{c}) = 0}_{4.\ \Longrightarrow} \overset{\text{SIS}}{\Longrightarrow} \mathbf{a} \circ \mathbf{b} = \mathbf{c}$

Preprocessed   Sent by prover

Conclusion

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

Russell W. F. Lai
Aalto University, Finland
russell.lai@aalto.fi
russell-lai.hk

References I

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 549–579. DOI: 10.1007/978-3-030-84245-1_19.

[ACLMT22] Martin R. Albrecht et al. "Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable - (Extended Abstract)". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 102–132. DOI: 10.1007/978-3-031-15979-4_4.

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 519–548. DOI: 10.1007/978-3-030-84245-1_18.

[BCIOP13] Nir Bitansky et al. "Succinct Non-interactive Arguments via Linear Interactive Proofs". In: *TCC 2013*. Ed. by Amit Sahai. Vol. 7785. LNCS. Springer, Heidelberg, Mar. 2013, pp. 315–333. DOI: 10.1007/978-3-642-36594-2_18.

References II

[BISW17] Dan Boneh et al. “Lattice-Based SNARGs and Their Application to More Efficient Obfuscation”. In: *EUROCRYPT 2017, Part III*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10212. LNCS. Springer, Heidelberg, 2017, pp. 247–277. DOI: 10.1007/978-3-319-56617-7_9.

[BISW18] Dan Boneh et al. “Quasi-Optimal SNARGs via Linear Multi-Prover Interactive Proofs”. In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Heidelberg, 2018, pp. 222–255. DOI: 10.1007/978-3-319-78372-7_8.

[BLNS20] Jonathan Bootle et al. “A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge”. In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 441–469. DOI: 10.1007/978-3-030-56880-1_16.

[BS22] Ward Beullens and Gregor Seiler. *LaBRADOR: Compact Proofs for R1CS from Module-SIS*. Cryptology ePrint Archive, Report 2022/1341. https://eprint.iacr.org/2022/1341. 2022.

References III

[GMNO18] Rosario Gennaro et al. "Lattice-Based zk-SNARKs from Square Span Programs". In: *ACM CCS 2018*. Ed. by David Lie et al. ACM Press, Oct. 2018, pp. 556–573. DOI: 10.1145/3243734.3243845.

How hard is vanishing SIS?

$$\text{kRISIS} \leq \text{vSIS} \leq \text{kRISIS}^{\text{Knowledge-kRISIS}}$$

kRISIS $\leq$ vSIS (Solve vSIS $\implies$ Solve kRISIS):

- † Given kRISIS instance $\left(\mathbf{A}, \mathbf{T}, \mathbf{v}, (\mathbf{u_g})_{\mathbf{g}\in\mathcal{G}}, g^*\right)$.
- † Run vSIS solver on $\left(\mathcal{G} \cup \{\mathbf{g}^*\}, \mathbf{v}\right)$ to obtain $\mathbf{p} = \left(p_\mathbf{g}\right)_{\mathbf{g}\in\mathcal{G}}$ such that

$$\sum_{\mathbf{g}\in\mathcal{G}} p_\mathbf{g} \cdot \mathbf{g}(\mathbf{v}) + p_{\mathbf{g}^*} \cdot \mathbf{g}^*(\mathbf{v}) = \mathbf{0} \bmod q \qquad \text{and} \qquad \|\mathbf{p}\| \approx 0.$$

- † Output $\mathbf{u}^* = \sum_{\mathbf{g}\in\mathcal{G}} p_\mathbf{g} \cdot \mathbf{u_g}$ and $s^* = -p_{\mathbf{g}^*}$.
- † Clearly,

$$\mathbf{A} \cdot \mathbf{u}^* = \mathbf{T} \cdot \mathbf{g}^*(\mathbf{v}) \cdot s^* \bmod q \qquad \text{and} \qquad \|\mathbf{u}\| \approx 0.$$

How hard is vanishing SIS?

$$\text{kRISIS} \leq \text{vSIS} \leq \text{kRISIS}^{\text{Knowledge-kRISIS}}$$

vSIS $\leq$ kRISIS$^{\text{Knowledge-kRISIS}}$ (Assume knowledge-kRISIS. Solve kRISIS $\implies$ Solve vSIS):

- † Given vSIS instance $(\mathcal{G} \cup \{g^*\}, \mathbf{v})$.
- † Sample $(\mathbf{A}, \mathbf{T}, \mathbf{v}, (\mathbf{u_g})_{\mathbf{g} \in \mathcal{G}})$.
- † Run kRISIS solver on $(\mathbf{A}, \mathbf{T}, \mathbf{v}, (\mathbf{u_g})_{\mathbf{g} \in \mathcal{G}})$ to obtain $(\mathbf{u}^*, s^*)$ such that

$$\mathbf{A} \cdot \mathbf{u}^* = \mathbf{T} \cdot \mathbf{g}^*(\mathbf{v}) \cdot s^* \bmod q \qquad \text{and} \qquad \|\mathbf{u}\| \approx 0.$$

- † Run the knowledge-kRISIS extractor on the above algorithm to extract a vector $\mathbf{p}$ satisfying

$$\sum_{\mathbf{g} \in \mathcal{G}} p_{\mathbf{g}} \cdot \mathbf{g}(\mathbf{v}) = s^* \cdot \mathbf{g}^*(\mathbf{v}) \bmod q \qquad \text{and} \qquad \|\mathbf{p}\| \approx 0.$$

- † Let $\mathbf{p}^* = (p_{\mathbf{g}})_{\mathbf{g} \in \mathcal{G} \cup \{g^*\}}$ where $p_{\mathbf{g}^*} := -s^*$.
- † Output $\mathbf{p}^*$.
- † Clearly, $\sum_{\mathbf{g} \in \mathcal{G} \cup \{g^*\}} p_{\mathbf{g}} \cdot \mathbf{g}(\mathbf{v}) = \mathbf{0} \bmod q$ and $\|\mathbf{p}^*\| \approx 0$.

Connections to NTRU and IdealSVP

- NTRU: Given $h = f \cdot g^{-1} \bmod q$ where $\|(f, g)\| \approx 0$, find f', g' such that

$$f' + g' \cdot h = 0 \bmod q \qquad \text{and} \qquad \|(f', g')\| \approx 0.$$

  Can be see as univariate degree-1 vSIS with special instance distribution.
- Assuming decision NTRU, NTRU $\leq$ vSIS. $(*)$
- Assuming decision NTRU, worst-to-average reduction for vSIS. $(*)$
- IdealSVP $\leq$ NTRU $\stackrel{\text{generalise}}{\rightarrow}$ IdealSVP $\leq$ vSIS. $(*)$

$(*)$: For very restrictive parameters.

Trivial (Non-)Attacks

- Solve vSIS as standard SIS
- Hope that $v^i = 0 \bmod q$ for some small i, then $p(X) = X^i$ is a trivial solution.
  - Ruled out by sampling $v \leftarrow\$ \mathcal{R}_q^\times$.
- Hope that $v^i = c \bmod q$ for some $c \approx 0$ for some small i, then $p(X) = X^i - c$ is a trivial solution.
  - Number of elements in $\mathcal{R}_q$ of norm at most β is $(2\beta + 1)^{\deg(\mathcal{R})}$.
  - Let q be such that $\left(\frac{2\beta+1}{q}\right)^{\deg(\mathcal{R})} = \mathsf{negl}(\lambda)$.
  - Heuristically, think of the "multiplication-by-v" map $a \mapsto a \cdot v \bmod q$ as a random permutation over $\mathcal{R}_q^\times$.
  - The probability of hitting an element of norm at most β is negligible.

Divide-and-Conquer Attack

Idea 1

- Split an n-point vSIS problem into f n/f-point vSIS problems.
- Split $V = \{v_1, \ldots, v_n\}$ into $V_1, \ldots, V_f$ where $|V_i| = n/f$.
- For each $i \in [f]$, find short polynomial $p_i \in \mathcal{R}[X]$ vanishing at V_i.
- Output $p = \prod_{i=1}^{f} p_i$.

Idea 2

- Split a 1-point vSIS problem over $\mathcal{R}$ into $\deg(\mathcal{R})$ 1-point vSIS problems over $\mathbb{Z}$.
- Suppose $\langle q\mathcal{R} \rangle$ splits into f (not necessarily prime) ideals.
- Represent v in CRT basis by $(v_1, \ldots, v_f) \in \mathbb{Z}_q^f$.
- For each $i \in [f]$, find short polynomial $p_i \in \mathbb{Z}[X]$ vanishing at v_i.
- Output $p = \prod_{i=1}^{f} p_i$.

Divide-and-Conquer Attack

Non-Devastation

- † Norm of solution grows exponentially in f, the number of sub-problems.
- † Setting $q = \text{poly}(\lambda) \implies$ Can only afford $f = O(1)$.