Orbweaver

Succinct Linear Functional Commitments from Lattices

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

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

Form used in [Gro10, LRY16, AC20]
Evaluation verification equation

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

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

\[
\begin{align*}
\langle f, x \rangle & \equiv \left( \sum_{i=1}^{n} f_i \cdot Y^{-i} \right) \cdot \left( \sum_{i=1}^{n} x_i \cdot Y^i \right) + \sum_{i = -n + 1, i \neq 0}^{n-1} b_i \cdot Y^i \mod q \\
\end{align*}
\]

Form used in [Gro10,LRY16,AC20], translated to lattice setting using techniques from [ACLMT22]
Ring Vandermonde SIS (R-V-SIS) commitment

\[ c := \sum_{i=1}^{n} x_i \cdot v^i \mod q , \text{ where } v \leftarrow R_q \text{ is public} \]
Ring Vandermonde SIS (R-V-SIS) commitment

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

- Ajtai’s R-SIS commitment, with a Vandermonde key
Ring Vandermonde SIS (R-V-SIS) commitment

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

- 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

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

\[
\begin{align*}
&c_f \quad \cdot \quad c_x \\
\text{(preprocessed)}
\end{align*}
\]

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

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

\[ c_f \cdot c_x \equiv y \]

(preprocessed)

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

Generate short preimages $u_i$ for $i \in \{-n + 1, \ldots, n - 1\}\setminus\{0\}$ such that

$$\langle a, u_i \rangle \equiv v^i \mod q$$

Using a trapdoor public SIS matrix $a$ [MP12]
Computing the proof

- Given $\langle a, u_i \rangle \equiv v^i \mod q$ except for $i = 0$

- Where $b_i$ is the sum of cross terms corresponding to the coefficient of $v^i$ compute

$$
\pi := \sum_{i = -n + 1, i \neq 0}^{n-1} b_i \cdot u_i \mod q
$$
Computing the proof

• Given \( \langle a, u_i \rangle \equiv v^i \mod q \) except for \( i = 0 \)

• Where \( b_i \) is the sum of cross terms corresponding to the coefficient of \( v^i \) compute

\[
\pi := \sum_{\substack{i = -n + 1, \\
i \neq 0}}^{n-1} b_i \cdot u_i \mod q
\]

• Then

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

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

- Then

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

- $\mathbf{f}, \mathbf{x}$ short $\implies b_i$ short, $u_i$ short $\implies \pi$ short
Evaluation binding

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

\[
c_f \cdot c_x \equiv y + \langle a, \pi \rangle \mod q
\]
Evaluation binding

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

\[
c_f \cdot c_x \equiv y + \langle a, \pi \rangle \mod q
\]

\[
\langle a, \pi - \pi' \rangle \equiv y' - y \mod q
\]
Evaluation binding

\[ \langle a, \pi - \pi' \rangle \equiv y' - y \mod q \]

- 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

\[ \langle a, \pi - \pi' \rangle \equiv y' - y \mod q \]

- 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 \( v \leftarrow R_q \) all \( v^i \mod q \) will be long whp, as will the random targets
Multiple outputs

Can prove $\langle f_i, x \rangle = y_i$ for $i \in [t]$ with a single evaluation proof:

$$\langle a, \pi \rangle \equiv c \cdot \sum_{i=1}^{t} h_i \cdot c k_f_i - \sum_{i=1}^{t} h_i \cdot y_i \mod q$$
Multiple outputs

Can prove $\langle f_i, x \rangle = y_i$ for $i \in [t]$ with a single evaluation proof:

$$\langle a, \pi \rangle \equiv c \cdot \sum_{i=1}^{t} h_i \cdot c^k_f - \sum_{i=1}^{t} h_i \cdot y_i \mod q$$

**Key observation:** the prover submits a separate knowledge proof $\pi'$ for $c$ from which we extract $x$. It’s thus unnecessary to extract the hypothetical $\pi_i$ s.t.

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

Using extracted $x$ we get

$$\langle a, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot (\langle f_i, x \rangle - y_i) - \sum_{i=1}^{t} h_i \cdot \sum_{j=-n+1}^{n-1} b_{i,j} \cdot Y^i \mod q$$
Multiple outputs

\[ \langle a, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot (\langle f_i, x \rangle - y_i) - \sum_{i=1}^{t} h_i \cdot \sum_{j=-n+1, j \neq 0}^{n-1} b_{i,j} \cdot Y^i \mod q \]

\[ p(h_1, \ldots, h_t) \]

- 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
Multiple outputs

\[ \langle a, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot (\langle f_i, x \rangle - y_i) - \sum_{i=1}^{t} h_i \cdot \sum_{j=-n+1, j \neq 0}^{n-1} b_{i,j} \cdot Y^i \mod q \]

\[ p(h_1, \ldots, h_t) \]

- 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

\[
\langle a, \pi \rangle \equiv \sum_{i=1}^{t} h_i \cdot (\langle f_i, x \rangle - y_i) - \sum_{i=1}^{t} h_i \cdot \sum_{j=-n+1, j \neq 0}^{n-1} b_{i,j} \cdot Y^i \mod q
\]

\[
p(h_1, \ldots, h_t)
\]

- 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]
- Better to perform ternary decomposition on \( f, x \) and batch verification
Proof and SRS sizes for $\mathbb{Z}_2^{32}$

<table>
<thead>
<tr>
<th>log2(x)</th>
<th>18</th>
<th>22</th>
<th>26</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>c</td>
<td>$ (B)</td>
<td>293</td>
<td>347</td>
</tr>
<tr>
<td>total proof size (KiB)</td>
<td>845</td>
<td>1,081</td>
<td>1,315</td>
<td>1,571</td>
</tr>
<tr>
<td>verifier key (MiB)</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>prover key (GiB)</td>
<td>0.3</td>
<td>6</td>
<td>111</td>
<td>2,070</td>
</tr>
</tbody>
</table>

- These are maximum proof sizes. When $f$ or $x$ are sparse or have entries much smaller than the norm bound this is reflected by the proof size.
Proof and SRS sizes for $\mathbb{Z}_{2^{32}}$

<table>
<thead>
<tr>
<th>log2(x)</th>
<th>18</th>
<th>22</th>
<th>26</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>c</td>
<td>(B)$</td>
<td>293</td>
<td>347</td>
</tr>
<tr>
<td>total proof size (KiB)</td>
<td>845</td>
<td>1,081</td>
<td>1,315</td>
<td>1,571</td>
</tr>
<tr>
<td>verifier key (MiB)</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>prover key (GiB)</td>
<td>0.3</td>
<td>6</td>
<td>111</td>
<td>2,070</td>
</tr>
</tbody>
</table>

- These are maximum proof sizes. When $f$ or $x$ 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 $\mathbb{Z}_{2^{32}}$

<table>
<thead>
<tr>
<th>log2(x)</th>
<th>18</th>
<th>22</th>
<th>26</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c</td>
<td>(B)</td>
<td>293</td>
<td>347</td>
</tr>
<tr>
<td>total proof size (KiB)</td>
<td>845</td>
<td>1,081</td>
<td>1,315</td>
<td>1,571</td>
</tr>
<tr>
<td>verifier key (MiB)</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>prover key (GiB)</td>
<td>0.3</td>
<td>6</td>
<td>111</td>
<td>2,070</td>
</tr>
</tbody>
</table>

- These are maximum proof sizes. When $f$ or $x$ 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\textsuperscript{1}, Russell W. F. Lai\textsuperscript{2}, Giulio Malavolta\textsuperscript{3}

\textsuperscript{1}AIT Austrian Institute of Technology, Austria
\textsuperscript{2}Aalto University, Finland
\textsuperscript{3}Max Planck Institute for Security and Privacy, Germany

CRYPTO, Santa Barbara, CA, U.S., 2023
# Lattice-based Succinct Arguments

<table>
<thead>
<tr>
<th>Approach</th>
<th>Publicly verifiable</th>
<th>Sublinear-verifier (preprocessing)</th>
<th>Linear-prover</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCP/IOP + linear-only enc. [BCIOP13; BISW17; BISW18; GMNO18]</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Linearisation + folding [BLNS20; AL21; ACK21; BS22]</td>
<td>✓</td>
<td>✗  $\tilde{O}_\lambda(</td>
<td>\text{stmt}</td>
</tr>
<tr>
<td>Direct [ACLMT22]</td>
<td>✓</td>
<td>✓</td>
<td>✗  $\tilde{O}_\lambda(</td>
</tr>
<tr>
<td>This work (and [BCS23])</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
# Lattice-based Succinct Arguments

<table>
<thead>
<tr>
<th>Approach</th>
<th>Publicly verifiable</th>
<th>Sublinear-verifier (preprocessing)</th>
<th>Linear-prover</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCP/IOP + linear-only enc. [BCIOP13; BISW17; BISW18; GMNO18]</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Linearisation + folding [BLNS20; AL21; ACK21; BS22]</td>
<td>✓</td>
<td>✗ 𝑂̃(</td>
<td>stmt</td>
</tr>
<tr>
<td>Direct [ACLMT22]</td>
<td>✓</td>
<td>✓</td>
<td>✗ 𝑂̃(</td>
</tr>
<tr>
<td>This work (and [BCS23])</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
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(|stmt|)) commitment key
  ‡ (Almost) additively and multiplicatively homomorphic
  ‡ Admit $\tilde{O}(|stmt|)$-prover polylog(|stmt|)-verifier arguments for commitment openings
† New lattice-based succinct arguments for NP $\iff$ 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 \textit{multiplicatively} homomorphic
   ‡ Admit \(\tilde{O}(|\text{stmt}|)\)-prover polylog\(|\text{stmt}|\)-verifier arguments for commitment openings

† New lattice-based succinct arguments for NP \(\leftrightarrow\) 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 \textit{multiplicatively} homomorphic
  ‡ Admit \( \tilde{O}(|\text{stmt}|) \)-prover polylog\(|\text{stmt}|\)-verifier arguments for commitment openings

† New lattice-based succinct arguments for NP \iff\ 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
1. Preliminaries

Short Integer Solution (SIS) Assumption

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

$$A \cdot u = 0 \mod q$$

and

$$0 < \|u\| \approx 0.$$

† Shorthand: If $u$ is a short non-zero vector satisfying $A \cdot u = v \mod q$, write

$$u \in A^{-1}(v).$$
## Vanishing SIS as SIS Generalisations

<table>
<thead>
<tr>
<th>SIS</th>
<th>Find short solution to linear equations</th>
</tr>
</thead>
</table>
|     | \[
<table>
<thead>
<tr>
<th></th>
<th>{A \cdot u = 0 \mod q \quad \text{and} \quad 0 &lt; |u| \approx 0.}</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIS (Alternative Interpretation)</td>
<td>Find linear function with short coefficients which vanishes at all given points</td>
</tr>
<tr>
<td>Vanishing SIS (vSIS)</td>
<td>Find polynomial (from some class) with short coefficients which vanishes at all given points</td>
</tr>
</tbody>
</table>
### Vanishing SIS as SIS Generalisations

<table>
<thead>
<tr>
<th>SIS</th>
<th>Find short solution to linear equations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ \mathbf{A} \cdot \mathbf{u} = 0 \mod q \quad \text{and} \quad 0 &lt; | \mathbf{u} | \approx 0. ]</td>
</tr>
</tbody>
</table>

| 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

\[ A \cdot u = 0 \mod q \]

and

\[ 0 < ||u|| \approx 0. \]

**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

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

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

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

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

$$\begin{bmatrix} v & v^2 & \ldots & v^m \end{bmatrix} \cdot p = 0 \mod q \quad \text{and} \quad 0 < \|p\| \approx 0.$$
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 $p$:
  \[
  \text{com}(p) = p(v) \mod q.
  \]
† Binding: If $p(v) = p'(v) \mod q$, then we break vSIS, i.e.
  \[
  (p - p')(v) = 0 \mod q \quad \|p - p'\| \leq \|p\| + \|p'\| \approx 0.
  \]
† (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):
  \[
  p(v) + p'(v) = (p + p')(v) \mod q \quad \|p + p'\| \leq \|p\| + \|p'\| \approx 0
  \]
  \[
  p(v) \cdot p'(v) = (p \cdot p')(v) \mod q \quad \|p \cdot p'\| \leq \|p\| \cdot \|p'\| \approx 0.
  \]
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 $p$:
  \[
  \text{com}(p) = p(v) \mod q.
  \]

† Binding: If $p(v) = p'(v) \mod q$, then we break vSIS, i.e.
  \[
  (p - p')(v) = 0 \mod q \quad \|p - p'\| \leq \|p\| + \|p'\| \approx 0.
  \]

† (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):
  \[
  \begin{align*}
  p(v) + p'(v) &= (p + p')(v) \mod q \quad \|p + p'\| \leq \|p\| + \|p'\| \approx 0 \\
  p(v) \cdot p'(v) &= (p \cdot p')(v) \mod q \quad \|p \cdot p'\| \leq \|p\| \cdot \|p'\| \approx 0.
  \end{align*}
  \]
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 $p$:
$$\text{com}(p) = p(v) \mod q.$$

† Binding: If $p(v) = p'(v) \mod q$, then we break vSIS, i.e.
$$ (p - p')(v) = 0 \mod q \quad \quad \|p - p'\| \leq \|p\| + \|p'\| \approx 0. $$

† (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):
$$ p(v) + p'(v) = (p + p')(v) \mod q \quad \quad \|p + p'\| \leq \|p\| + \|p'\| \approx 0 $$
$$ p(v) \cdot p'(v) = (p \cdot p')(v) \mod q \quad \quad \|p \cdot p'\| \approx \|p\| \cdot \|p'\| \approx 0.$$
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 \( p \):
  \[ \text{com}(p) = p(v) \mod q. \]

† Binding: If \( p(v) = p'(v) \mod q \), then we break vSIS, i.e.
  \[ (p - p')(v) = 0 \mod q \quad \|p - p'\| \leq \|p\| + \|p'\| \approx 0. \]

† (Almost) additively and multiplicatively homomorphic (w.r.t. polynomial addition and multiplications):
  \[ p(v) + p'(v) = (p + p')(v) \mod q \quad \|p + p'\| \leq \|p\| + \|p'\| \approx 0 \]
  \[ p(v) \cdot p'(v) = (p \cdot p')(v) \mod q \quad \|p \cdot p'\| \approx \|p\| \cdot \|p'\| \approx 0. \]
Encoding Vectors as (Laurent) Polynomials

\[ a := (a_1, \ldots, a_m) \in \mathcal{R}^m \quad \bar{p}_a(X) := p_a(X^{-1}) := a_1X^{-1} + a_2X^{-2} + \ldots + a_mX^{-m} \]

\[ b := (b_1, \ldots, b_m) \in \mathcal{R}^m \quad p_b(X) := b_1X + b_2X^2 + \ldots + b_mX^m \]

Note that

\[ \bar{p}_a(X) \cdot p_b(X) = \hat{p}_{a \ast b}(X) \implies \hat{p}_{a \ast b} \text{ has } O(m) \text{ terms (lots of collisions!)} \]

where

† \( a \ast b := \left( \sum_{j-i=k} a_i \cdot b_j \right)_k^{i=-m} \) “convolution”, and

† constant term is given by \( \langle a, b \rangle \).
Encoding Vectors as (Laurent) Polynomials

\[ \mathbf{a} := (a_1, \ldots, a_m) \in \mathbb{R}^m \quad \overline{p}_\mathbf{a}(X) := p_{\mathbf{a}}(X^{-1}) := a_1 X^{-1} + a_2 X^{-2} + \ldots + a_m X^{-m} \]

\[ \mathbf{b} := (b_1, \ldots, b_m) \in \mathbb{R}^m \quad p_{\mathbf{b}}(X) := b_1 X + b_2 X^2 + \ldots + b_m X^m \]

Note that
\[ \overline{p}_\mathbf{a}(X) \cdot p_{\mathbf{b}}(X) = \hat{p}_{\mathbf{a}*\mathbf{b}}(X) \implies \hat{p}_{\mathbf{a}*\mathbf{b}} \text{ has } O(m) \text{ terms (lots of collisions!)} \]

where
\[ \uparrow \mathbf{a} \ast \mathbf{b} := \left( \sum_{j-i=k} a_i \cdot b_j \right)_{k=-m}^m \text{ "convolution", and} \]
\[ \uparrow \text{ constant term is given by } \langle \mathbf{a}, \mathbf{b} \rangle. \]
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 $c_x := \rho_x(v)$.
† $\mathbf{x}$ is committed also in dual vSIS commitment as $\bar{c}_x := \bar{\rho}_x(v)$,
† $1$ is committed in dual vSIS commitment as $\bar{c}_1 := \bar{\rho}_1(v)$.

Observe that

\[
\sum_{i} x_i \cdot v^{i} \cdot \left( \sum_{j} x_j \cdot v^{-j} - \sum_{j} 1 \cdot v^{-j} \right) = \sum_{i} x_i \cdot (x_i - 1) + \text{mixed terms}
\]

\[
\hat{\rho}_x^{*(1-x)}(v) \quad \begin{pmatrix} c_x \vline \bar{c}_x \vline \bar{c}_1 \end{pmatrix}
\]
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 $c_x := \rho_x(v)$.
† $x$ is committed also in dual vSIS commitment as $\bar{c}_x := \bar{\rho}_x(v)$,
† $1$ is committed in dual vSIS commitment as $\bar{c}_1 := \bar{\rho}_1(v)$.

Observe that

$$
\sum_i x_i \cdot v^i \cdot \left( \sum_j x_j \cdot v^{-j} - \sum_j 1 \cdot v^{-j} \right) = \sum_i x_i \cdot (x_i - 1) + \text{mixed terms}
$$
2. Vanishing SIS Assumptions and Commitments

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 $c_x := p_x(v)$.
† $x$ is committed also in dual vSIS commitment as $\bar{c}_x := \bar{p}_x(v)$,
† $1$ is committed in dual vSIS commitment as $\bar{c}_1 := \bar{p}_1(v)$.

Observe that

$$\sum_i x_i \cdot v^i \cdot \left( \sum_j x_j \cdot v^{-j} - \sum_j 1 \cdot v^{-j} \right) = \sum_i x_i \cdot (x_i - 1) + \text{mixed terms}$$
2. Vanishing SIS Assumptions and Commitments

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 $c_x := p_x(v)$.
- $\mathbf{h} \circ \mathbf{x}$ 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 $\bar{c}_h := \bar{p}_h(v)$.

Observe that

$$
\sum_i x_i \cdot v^i \cdot \left( \sum_j h_j \cdot x_j \cdot v^{-j} - \sum_j h_j \cdot v^{-j} \right) = \sum_i h_i \cdot x_i \cdot (x_i - 1) + \text{mixed terms}
$$

$$
\hat{p}_{x \circ h}(x-1)(v)
$$
To prove that a vSIS commitment is committing to a (Laurent) polynomial without constant term:

\[
\begin{bmatrix}
    v & v^2 & \ldots & v^m \\
    v^{-1} & v^{-2} & \ldots & v^{-m}
\end{bmatrix} \cdot x = \begin{bmatrix}
    c_x \\
    \bar{c}_x
\end{bmatrix} \mod q \land \|x\| \approx 0,
\]

and

\[
\begin{bmatrix}
    v^{-m} & \ldots & v^{-1} & v^1 & \ldots & v^m
\end{bmatrix} \cdot w = \frac{c_x \cdot (\bar{c}_x - \bar{c}_1)}{\hat{c}} \mod q \land \|w\| \approx 0,
\]

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:

$$\begin{bmatrix} v & v^2 & \cdots & v^m \\ v^{-1} & v^{-2} & \cdots & v^{-m} \end{bmatrix} \cdot x = \begin{bmatrix} c_x \\ \bar{c}_x \end{bmatrix} \mod q \land \|x\| \approx 0,$$

and

$$\begin{bmatrix} v^{-m} & \cdots & v^{-1} & v^1 & \cdots & v^m \end{bmatrix} \cdot w = c_x \cdot (\bar{c}_x - \bar{c}_1) \mod q \land \|w\| \approx 0,$$

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 \(G\).

Assumption: If a PPT (quantum) algorithm \(A\), which on input \((A, T, v, (u_g)_{g \in G})\)
where \(A \in \mathcal{R}_{q}^{n \times m}\), \(T \in (\mathcal{R}_{q}^{\times})^{n \times t}\), \(v \in \mathcal{R}_{q}^{\times}\), and \(u_g \in A^{-1}(T \cdot g(v))\),
can find \((u, c)\) where \(u \in A^{-1}(T \cdot c)\),
then it must “know” short linear combination \(x\) such that
\[ c = \sum_{g \in G} g(v) \cdot x_g \mod q. \]
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

\[
(A, T, \nu, (u_g)_{g \in \mathcal{G}})
\]

where \(A \in \mathbb{R}^{n \times m}_q\), \(T \in (\mathbb{R}^{\times}_q)^{n \times t}\), \(\nu \in \mathbb{R}^{\times}_q\), and \(u_g \in A^{-1}(T \cdot g(\nu))\),

can find \((u, c)\) where

\[
u \in A^{-1}(T \cdot c),
\]

then it must “know” short linear combination \(x\) such that

\[
c = \sum_{g \in \mathcal{G}} g(\nu) \cdot x_g \mod q.
\]
Using Knowledge-kRISIS

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

\[
\begin{align*}
    w_0 &= 0 \\
    \hat{c} &= \hat{p}_w(\nu) \\
    ||w|| &\approx 0.
\end{align*}
\]

† Public parameters: kRISIS instance $(A, t, \nu, (u_i)_{i \in \pm [m]})$ where

\[
    u_i \in A^{-1}(t \cdot \nu^i).
\]

† Prover: Output $u = \sum_{i \in \pm [m]} u_i \cdot w_i$.

† Verifier: Check that $A \cdot u = t \cdot \hat{c} \mod q$ and $||u|| \approx 0$.

† 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

<table>
<thead>
<tr>
<th>Want to prove $\hat{c}$ and $w \in \mathcal{R}^{2m+1}$ satisfies:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0 = 0$</td>
</tr>
</tbody>
</table>

† Public parameters: kRISIS instance $(A, t, v, (u_i)_{i \in \pm[m]})$ where

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

† Prover: Output $u = \sum_{i \in \pm[m]} u_i \cdot w_i$. 

† Verifier: Check that $A \cdot u = t \cdot \hat{c} \mod q$ and $\|u\| \approx 0$. 

† 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.
3. Succinct Arguments for vSIS Commitment Openings

**Using Knowledge-kRISIS**

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

\[
\begin{align*}
    w_0 &= 0 \\
    \hat{c} &= \hat{\rho}_w(v) \\
    \|\mathbf{w}\| &\approx 0.
\end{align*}
\]

† Public parameters: kRISIS instance \( (\mathbf{A}, t, v, (u_i)_{i \in \pm [m]}) \) where

\[
    u_i \in \mathbf{A}^{-1}(t \cdot v^i).
\]

† Prover: Output \( \mathbf{u} = \sum_{i \in \pm [m]} u_i \cdot w_i \).

† Verifier: Check that \( \mathbf{A} \cdot \mathbf{u} = t \cdot \hat{c} \mod q \) 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:

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

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 $\mathcal{V}(\mathbf{M}, \mathbf{y})$

\[
\begin{align*}
\mathbf{y}_{12} &:= \mathbf{M}_1 \cdot \mathbf{x}_2 \\
\mathbf{y}_{21} &:= \mathbf{M}_2 \cdot \mathbf{x}_1 \\
\hat{\mathbf{x}}_c &:= c \cdot \mathbf{x}_1 + \mathbf{x}_2 \\
\hat{\mathbf{y}}_c &:= \mathbf{y}_{12} + \mathbf{y} \cdot c + \mathbf{y}_{21} \cdot c^2 \mod q \\
\hat{\mathbf{M}}_c &:= \mathbf{M}_1 + c \cdot \mathbf{M}_2
\end{align*}
\]

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
3. Succinct Arguments for vSIS Commitment Openings

**Lattice-based Bulletproofs**

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

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

where $m = 2^\ell$, $M = [M_1 \mid M_2]$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

**Prover $P((M, y), x)$**

- $y_{12} := M_1 \cdot x_2$
- $y_{21} := M_2 \cdot x_1$
- $\hat{x}_c := c \cdot x_1 + x_2$

**Verifier $V(M, y)$**

- $c \leftarrow$ $C$
- $\hat{M}_c := M_1 + c \cdot M_2$
- $\hat{y}_c := y_{12} + y \cdot c + y_{21} \cdot c^2 \mod q$

**Return**

$$\begin{cases} \hat{M}_c \cdot \hat{x}_c = \hat{y}_c \\ \|\hat{x}_c\| \approx 0 \end{cases}$$

Just another SIS relation but with only $m/2$ columns $\implies$ Recursion
### Lattice-based Bulletproofs

After $\ell$-fold recursive composition:

**Prover** $\mathcal{P}((\mathbf{M}, \mathbf{y}), \mathbf{x})$

\[
\begin{align*}
\mathbf{y}^{(1)}_{12}, \mathbf{y}^{(1)}_{21} \\
\vdots \\
\mathbf{y}^{(\ell)}_{12}, \mathbf{y}^{(\ell)}_{21} \\
\end{align*}
\]

\[
\begin{align*}
c_1 \\
\vdots \\
c_\ell \\
\end{align*}
\]

\[
\hat{\mathbf{x}}_{c_1, \ldots, c_\ell}
\]

**Verifier** $\mathcal{V}(\mathbf{M}, \mathbf{y})$

\[
\begin{align*}
(\hat{\mathbf{M}}_{c_1}, \hat{\mathbf{y}}_{c_1}) := \ldots \\
\vdots \\
(\hat{\mathbf{M}}_{c_1, \ldots, c_\ell}, \hat{\mathbf{y}}_{c_1, \ldots, c_\ell}) := \ldots \\
\end{align*}
\]

\[
\begin{align*}
\hat{\mathbf{M}}_{c_1, \ldots, c_\ell} \cdot \hat{\mathbf{x}}_{c_1, \ldots, c_\ell} = \hat{\mathbf{y}}_{c_1, \ldots, c_\ell} \\
\|\hat{\mathbf{x}}_{c_1, \ldots, c_\ell}\| \approx 0
\end{align*}
\]

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

After $\ell$-fold recursive composition:

Prover $P((M, y), x)$

$y^{(1)}_{12}, y^{(1)}_{21}$

$\leftarrow c_1$

$\vdots$

$y^{(\ell)}_{12}, y^{(\ell)}_{21}$

$\leftarrow c_\mu$

$\hat{x}_{c_1, \ldots, c_\ell}$

Verifier $V(M, y)$

$(\hat{M}_{c_1}, \hat{y}_{c_1}) := \ldots$

$\vdots$

$(\hat{M}_{c_1, \ldots, c_\ell}, \hat{y}_{c_1, \ldots, c_\ell}) := \ldots$

return $\left\{ \hat{M}_{c_1, \ldots, c_\ell} \cdot \hat{x}_{c_1, \ldots, c_\ell} = \hat{y}_{c_1, \ldots, c_\ell}, \|\hat{x}_{c_1, \ldots, c_\ell}\| \approx 0 \right\}$

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

Core Idea

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

Example for $\ell = 3$

\[
M = \begin{pmatrix} v & v^2 & v^3 & v^4 & v^5 & v^6 & v^7 & v^8 \end{pmatrix}
\]
\[
\hat{M}_{c_1} = \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} + \begin{pmatrix} v^5 & v^6 & v^7 & v^8 \end{pmatrix} \cdot c_1
\]
\[
= \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} \cdot (1 + v^4 \cdot c_1)
\]
\[
\hat{M}_{c_1,c_2} = \begin{pmatrix} v & v^2 \end{pmatrix} \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2)
\]
\[
\hat{M}_{c_1,c_2,c_3} = v \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \cdot (1 + v \cdot c_3)
\]
\[
= v \cdot \prod_{i=1}^{3} (1 + v^{2^{3-i}} \cdot c_i)
\]
Structured Folding for vSIS

Core Idea

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

Example for \( \ell = 3 \)

\[
\begin{align*}
M &= \begin{pmatrix} v & v^2 & v^3 & v^4 & v^5 & v^6 & v^7 & v^8 \end{pmatrix} \\
\hat{M}_{c_1} &= \left( \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} + \begin{pmatrix} v^5 & v^6 & v^7 & v^8 \end{pmatrix} \right) \cdot c_1 \\
&= \left( \begin{pmatrix} v & v^2 & v^3 & v^4 \end{pmatrix} \right) \cdot (1 + v^4 \cdot c_1) \\
\hat{M}_{c_1,c_2} &= \left( v \cdot v^2 \right) \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \\
\hat{M}_{c_1,c_2,c_3} &= v \cdot (1 + v^4 \cdot c_1) \cdot (1 + v^2 \cdot c_2) \cdot (1 + v \cdot c_3) \\
&= v \cdot \prod_{i=1}^{3} (1 + v^{2^{3-i}} \cdot c_i)
\end{align*}
\]
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