Lattice-Based Functional Commitments: Constructions and Cryptanalysis

David Wu
May 2024

based on joint works with Hoeteck Wee
Functional Commitments

\[ \sigma \]

Commit

"commitment"

Open + Verify

"opening"

\[ \pi \]

\[ f(x) \]
Functional Commitments

Commit\((crs, x) \rightarrow (\sigma, st)\)

Takes a common reference string and commits to an input \(x\)

Outputs commitment \(\sigma\) and commitment state \(st\)
Functional Commitments

- **Commit**($crs, x$) $\rightarrow$ ($\sigma, st$)
- **Open**($st, f$) $\rightarrow$ $\pi$
  
  Takes the commitment state and a function $f$ and outputs an opening $\pi$

- **Verify**($crs, \sigma, (f, y), \pi$) $\rightarrow$ 0/1
  
  Checks whether $\pi$ is valid opening of $\sigma$ to value $y$ with respect to $f$
Functional Commitments

**Binding**: efficient adversary cannot open $\sigma$ to two different values with respect to the same $f$

- $\sigma$
- $f(x)$
- $\pi$
- Verify$(\text{crs}, \sigma, (f, y_0), \pi_0) = 1$
- Verify$(\text{crs}, \sigma, (f, y_1), \pi_1) = 1$
**Functional Commitments**

\[ \sigma \xrightarrow{\text{Open + Verify}} \pi \]

**Succinctness:** commitments and openings should be short

- **Short commitment:** \(|\sigma| = \text{poly}(\lambda, \log |x|)\)
- **Short opening:** \(|\pi| = \text{poly}(\lambda, \log|x|, |f(x)|)\)

Will consider relaxation where \(|\sigma|\) and \(|\pi|\) can grow with **depth** of the circuit computing \(f\)
Special Cases of Functional Commitments

Vector commitments:

\[ [x_1, x_2, \ldots, x_n] \]

\[ \text{ind}_i(x_1, \ldots, x_n) := x_i \]

commit to a vector, open at an index

Polynomial commitments:

\[ [\alpha_0, \alpha_1, \ldots, \alpha_d] \]

\[ f_x(\alpha_0, \ldots, \alpha_d) := \alpha_0 + \alpha_1 x + \cdots + \alpha_d x^d \]

commit to a polynomial, open to the evaluation at \( x \)
## Succinct Functional Commitments

*(not an exhaustive list!)*

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Function Class</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Mer87]</td>
<td>vector commitment</td>
<td>collision-resistant hash functions</td>
</tr>
<tr>
<td>[LY10, CF13, LM19, GRW20]</td>
<td>vector commitment</td>
<td>(q)-type pairing assumptions</td>
</tr>
<tr>
<td>[CF13, LM19, BBF19]</td>
<td>vector commitment</td>
<td>groups of unknown order</td>
</tr>
<tr>
<td>[PPS21]</td>
<td>vector commitment</td>
<td>short integer solutions (SIS)</td>
</tr>
<tr>
<td>[KZG10, Lee20]</td>
<td>polynomial commitment</td>
<td>(q)-type pairing assumptions</td>
</tr>
<tr>
<td>[BFS19, BHRRS21, BF23]</td>
<td>polynomial commitment</td>
<td>groups of unknown order</td>
</tr>
<tr>
<td>[LRY16]</td>
<td>linear functions</td>
<td>(q)-type pairing assumptions</td>
</tr>
<tr>
<td>[ACLMT22]</td>
<td>constant-degree polynomials</td>
<td>(k)-(R)-ISIS assumption (falsifiable)</td>
</tr>
<tr>
<td>[LRY16]</td>
<td>Boolean circuits</td>
<td>collision-resistant hash functions + SNARKs</td>
</tr>
<tr>
<td>[dCP23]</td>
<td>Boolean circuits</td>
<td>SIS (non-succinct openings in general)</td>
</tr>
<tr>
<td>[KLVW23]</td>
<td>Boolean circuits</td>
<td>LWE (via batch arguments)</td>
</tr>
<tr>
<td>[BCFL23]</td>
<td>Boolean circuits</td>
<td>twin (k)-(R)-ISIS (or (q)-type pairing assumption)</td>
</tr>
<tr>
<td>[WW23a, WW23b]</td>
<td>Boolean circuits</td>
<td>(\ell)-succinct SIS</td>
</tr>
<tr>
<td>[WW24]</td>
<td>Boolean circuits</td>
<td>(k)-Lin (pairings)</td>
</tr>
</tbody>
</table>
Framework for Lattice Commitments

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Common reference string (for inputs of length ℓ):

- matrices $A_1, \ldots, A_\ell \in \mathbb{Z}_q^{n \times m}$
- target vectors $t_1, \ldots, t_\ell \in \mathbb{Z}_q^n$

**auxiliary data**: cross-terms $u_{ij} \leftarrow A_i^{-1}(t_j) \in \mathbb{Z}_q^m$ where $i \neq j$

short (i.e., low-norm) vector satisfying $A_i u_{ij} = t_j$
Framework for Lattice Commitments

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Common reference string (for inputs of length $\ell$):

- matrices $A_1, \ldots, A_\ell \in \mathbb{Z}_q^{n \times m}$
- target vectors $t_1, \ldots, t_\ell \in \mathbb{Z}_q^n$

**auxiliary data**: cross-terms $u_{ij} \leftarrow A_i^{-1}(t_j) \in \mathbb{Z}_q^m$ where $i \neq j$

Commitment to $x \in \mathbb{Z}_q^\ell$:

$$c = \sum_{i \in [\ell]} x_i t_i$$

Linear combination of target vectors

Opening to value $y$ at index $i$:

- short $v_i$ such that $c = A_i v_i + y \cdot t_i$

Honest opening:

$$v_i = \sum_{j \neq i} x_j u_{ij}$$

$$A_i v_i + x_i t_i = \sum_{j \neq i} x_j A_i u_{ij} + x_i t_i = \sum_{j \in [\ell]} x_j t_j = c$$

Correct as long as $x$ is short
**Framework for Lattice Commitments**

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Common reference string (for inputs of length $\ell$):

- matrices $A_1, \ldots, A_\ell \in \mathbb{Z}_q^{n \times m}$
- target vectors $t_1, \ldots, t_\ell \in \mathbb{Z}_q^n$

**auxiliary data:** cross-terms $u_{ij} \leftarrow A_i^{-1}(t_j) \in \mathbb{Z}_q^m$ where $i \neq j$

[PPS21]: $A_i \leftarrow \mathbb{Z}_q^{n \times m}$ and $t_i \leftarrow \mathbb{Z}_q^n$ are independent and uniform 

**suffices for vector commitments (from SIS)**

[ACLMT21]: $A_i = W_i A$ and $t_i = W_i u_i$ where $W_i \leftarrow \mathbb{Z}_q^{n \times n}$, $A \leftarrow \mathbb{Z}_q^{n \times m}$, $u_i \leftarrow \mathbb{Z}_q^n$

(one candidate adaptation to the integer case)

*generalizes to functional commitments for constant-degree polynomials (from $k$-R-ISIS)*
Our Approach

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

**Verification invariant:** \( c = A_i v_i + x_i t_i \) \( \forall i \in [\ell] \)

for a short \( v_i \)

**Our approach:** rewrite \( \ell \) relations as a single linear system

\[
\begin{bmatrix}
A_1 \\
\vdots \\
A_\ell
\end{bmatrix}
\begin{bmatrix}
-I_n \\
\vdots \\
-I_n
\end{bmatrix}
\begin{bmatrix}
v_1 \\
v_{\ell}
\end{bmatrix}
= \begin{bmatrix}
-x_1 t_1 \\
\vdots \\
-x_\ell t_\ell
\end{bmatrix}
\]

\( I_n \) denotes the identity matrix
Our Approach

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

**Verification invariant:** \( c = A_i v_i + x_i t_i \quad \forall i \in [\ell] \)

*for a short* \( v_i \)

**Our approach:** rewrite \( \ell \) relations as a single linear system

\[
\begin{bmatrix}
A_1 \\
\vdots \\
A_\ell
\end{bmatrix}
\begin{bmatrix}
-v_1 \\
\vdots \\
-v_\ell
\end{bmatrix}
= \begin{bmatrix}
-x_1 t_1 \\
\vdots \\
-x_\ell t_\ell
\end{bmatrix}
\]

For security and functionality, it will be useful to write \( c = G \hat{c} \)

\[
G = \begin{bmatrix}
1 & 2 & \ldots & 2^{\lfloor \log q \rfloor} \\
\vdots & & & \vdots \\
& & & \\
1 & 2 & \ldots & 2^{\lfloor \log q \rfloor}
\end{bmatrix}
\]
Our Approach

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

**Verification invariant:**  \( c = A_i v_i + x_i t_i \) \( \forall i \in [\ell] \)

*for a short* \( v_i \)

**Our approach:** rewrite \( \ell \) relations as a single linear system

\[
\begin{bmatrix}
A_1 \\
\vdots \\
A_\ell
\end{bmatrix}
\begin{bmatrix}
-G \\
\vdots \\
-G
\end{bmatrix}
\begin{bmatrix}
v_1 \\
\vdots \\
v_\ell
\end{bmatrix}
= \begin{bmatrix}
-x_1 t_1 \\
\vdots \\
-x_\ell t_\ell
\end{bmatrix}
\]

**Common reference string:**
- matrices \( A_1, \ldots, A_\ell \in \mathbb{Z}_{q}^{n \times m} \)
- target vectors \( t_1, \ldots, t_\ell \in \mathbb{Z}_{q}^{m} \)

**auxiliary data:** cross-terms \( u_{ij} \leftarrow A_i^{-1}(t_j) \)
Our Approach

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Verification invariant: $c = A_i v_i + x_i t_i \quad \forall i \in [\ell]$

\[
\begin{bmatrix}
A_1 & -G & \vdots \\
& \ddots & \vdots \\
A_\ell & -G & \\
\end{bmatrix}
\begin{bmatrix}
v_1 \\
\vdots \\
v_\ell \\
\end{bmatrix}
= 
\begin{bmatrix}
x_1 t_1 \\
\vdots \\
x_\ell t_\ell \\
\end{bmatrix}
\]

Our approach: rewrite $\ell$ relations as a single linear system (and publish a trapdoor for it)

Common reference string:
- matrices $A_1, \ldots, A_\ell \in \mathbb{Z}_q^{n \times m}$
- target vectors $t_1, \ldots, t_\ell \in \mathbb{Z}_q^n$
- auxiliary data: cross terms $u_{ij} \leftarrow A_i^{-1}(t_j)$

 Trapdoor for $B_\ell$ can be used to sample short solutions $x$ to the linear system $B_\ell x = y$ (for arbitrary $y$)
Our Approach

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Verification invariant: \( c = A_i v_i + x_i t_i \) for all \( i \in [\ell] \)

for a short \( v_i \)

Our approach: rewrite \( \ell \) relations as a single linear system (and publish a trapdoor for it)

\[
\begin{bmatrix}
A_1 & \ldots & -G \\
\vdots & \ddots & \vdots \\
A_\ell & \ldots & -G
\end{bmatrix}
\begin{bmatrix}
v_1 \\
\vdots \\
v_\ell \\
\hat{c}
\end{bmatrix}
= 
\begin{bmatrix}
-x_1 t_1 \\
\vdots \\
-x_\ell t_\ell
\end{bmatrix}
\]

Committing to an input \( x \):

Use trapdoor for \( B_\ell \) to jointly sample a solution \( v_1, \ldots, v_\ell, \hat{c} \)

\( c = G \hat{c} \) is the commitment and \( v_1, \ldots, v_\ell \) are the openings
Proving Security

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Verification invariant: \( c = A_i v_i + x_i t_i \quad \forall i \in [\ell] \)
for a short \( v_i \)

Suppose adversary can break binding

outputs \( c, (v_i, x_i), (v'_i, x'_i) \) such that

\[
\begin{align*}
c &= A_i v_i + x_i t_i \\
&= A_i v'_i + x'_i t_i
\end{align*}
\]

- Short integer solutions (SIS)
  - given \( A \leftarrow \mathbb{Z}_{q \times m}^n \), hard to find short \( x \neq 0 \) such that \( Ax = 0 \)

- \( A_i (v_i - v'_i) = (x'_i - x_i) t_i \)
  - (short)
  - (non-zero)

Looks like an SIS solution...

How to choose \( A_i, t_i \)?
Proving Security

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

**Verification invariant:** \( c = A_i v_i + x_i t_i \quad \forall i \in \llbracket \ell \rrbracket \)

for a short \( v_i \)

Suppose adversary can break binding

outputs \( c, (v_i, x_i), (v'_i, x'_i) \) such that

\[
c = A_i v_i + x_i t_i = A_i v'_i + x'_i t_i
\]

set \( A_i \leftarrow \mathbb{Z}_q^{n \times m} \)

set \( t_i = e_1 = [1, 0, \ldots, 0]^T \)

(cannot set \( t_i = 0 \) as otherwise, it could be \( v_i = v'_i \))

Short integer solutions (SIS)

given \( A \leftarrow \mathbb{Z}_q^{n \times m} \), hard to find short \( x \neq 0 \) such that \( Ax = 0 \)

\[
A_i (v_i - v'_i) = (x'_i - x_i) e_1
\]

\( v_i - v'_i \) is a SIS solution for \( A_i \) without the first row
Proving Security

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

Verification invariant: \( c = A_i v_i + x_i t_i \quad \forall i \in [\ell] \)

\( \text{for a short } v_i \)

Adversary that breaks binding can solve SIS with respect to \( A_i \)

\( \text{(technically } A_i \text{ without the first row – which is equivalent to SIS with dimension } n - 1) \)

but... adversary also gets additional information beyond \( A_i \)

\[
B_\ell = \begin{bmatrix}
A_1 & \cdots & -G \\
\vdots & \ddots & \vdots \\
A_\ell & \cdots & -G \\
\end{bmatrix}
\]

Adversary sees \textbf{trapdoor} for \( B_\ell \)
Basis-Augmented SIS (BASIS) Assumption

Captures and generalizes other lattice-based functional commitments [PPS21, ACLMT22]

**Verification invariant:** \( c = A_i v_i + x_i t_i \quad \forall i \in [\ell] \)
for a short \( v_i \)

Adversary that breaks binding can solve SIS with respect to \( A_i \)

Basis-augmented SIS (BASIS) assumption:

\textit{SIS is hard with respect to} \( A_i \)
\textit{given a trapdoor (a basis) for the matrix}

\[
B_\ell = \begin{bmatrix} A_1 & \quad \quad \quad \quad & -G \\
\vdots & \quad & \vdots \\
A_\ell & \quad & -G \\
\end{bmatrix}
\]

Can simulate CRS from BASIS challenge:
Matrices \( A_1, \ldots, A_\ell \leftarrow \mathbb{Z}_q^{n \times m} \)
trapdoor for \( B_\ell \)
Basis-Augmented SIS (BASIS) Assumption

SIS is hard with respect to $A_i$ given a trapdoor (a basis) for the matrix

$$B_\ell = \begin{bmatrix} A_1 & -G \\ \vdots \\ A_\ell & -G \end{bmatrix}$$

When $A_1, \ldots, A_\ell \leftarrow \mathbb{Z}_{q}^{n \times m}$ are uniform and independent:

*hardness of SIS implies hardness of BASIS*

(follows from standard lattice trapdoor extension techniques)
Vector Commitments from SIS

Common reference string (for inputs of length $\ell$):

matrices $A_1, \ldots, A_\ell \in \mathbb{Z}_q^{n \times m}$

**auxiliary data:** trapdoor for $B_\ell = \begin{bmatrix} A_1 & -G \\ \vdots & \vdots \\ A_\ell & -G \end{bmatrix}$

To commit to a vector $x \in \mathbb{Z}_q^\ell$: sample solution $(v_1, \ldots, v_\ell, \hat{c})$

\[
\begin{bmatrix} A_1 \\ \vdots \\ A_\ell \end{bmatrix} \begin{bmatrix} -G \\ \vdots \\ -G \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_\ell \hat{c} \end{bmatrix} = \begin{bmatrix} -x_1 e_1 \\ \vdots \\ -x_\ell e_\ell \end{bmatrix}
\]

Commitment is $c = G\hat{c}$

Openings are $v_1, \ldots, v_\ell$

Can commit and open to arbitrary $\mathbb{Z}_q$ vectors

Commitments and openings statistically **hide** unopened components

**Linearly homomorphinc:** $c + c'$ is a commitment to $x + x'$ with openings $v_i + v'_i$
Extending to Functional Commitments

**Goal:** commit to $\mathbf{x} \in \{0,1\}^\ell$, open to function $f(\mathbf{x})$

Suppose $f(\mathbf{x}) = \sum_{i \in [\ell]} \alpha_i x_i$ is a **linear** function

**Verification invariant:** $c = A_i \mathbf{v}_i + x_i \mathbf{t}_i \quad \forall i \in [\ell]$

Can also view $c$ as commitment to vector $x_i \mathbf{t}_i$ with respect to $A_i$ and opening $\mathbf{v}_i$

Suppose $c_1, c_2$ are commitments to vectors $\mathbf{u}_1, \mathbf{u}_2$ with respect to the same $A$

$$
c_1 = A \mathbf{v}_1 + \mathbf{u}_1 \quad \text{and} \quad c_2 = A \mathbf{v}_2 + \mathbf{u}_2
$$

$$
c_1 + c_2 = A(\mathbf{v}_1 + \mathbf{v}_2) + (\mathbf{u}_1 + \mathbf{u}_2)
$$
Extending to Functional Commitments

\[ c_1 = A\nu_1 + x_1 t \]
\[ \vdots \]
\[ c_\ell = A\nu_\ell + x_\ell t \]

Desired correctness relation

\[ W_1 c = A\nu_1 + x_1 t \]
\[ \vdots \]
\[ W_\ell c = A\nu_\ell + x_\ell t \]

Cannot define commitment to be \( (c_1, \ldots, c_\ell) \) since this is long

Instead, suppose \( c_i = W_i c \) can be derived from a (single) \( c \)

\[ B_\ell \]
\[
\begin{bmatrix}
A & -W_1 \\
\vdots & \vdots \\
A & -W_\ell \\
\end{bmatrix}
\begin{bmatrix}
\nu_1 \\
\vdots \\
\nu_\ell \\
\end{bmatrix}
= 
\begin{bmatrix}
-x_1 t \\
\vdots \\
-x_\ell t \\
\end{bmatrix}
\]

Our approach: rewrite \( \ell \) relations as a single linear system (and publish a trapdoor for it)
Extending to Functional Commitments

To commit to $x \in \{0,1\}^\ell$, use trapdoor for $B_\ell$ to sample $c, v_1, \ldots, v_\ell$ where

$$W_1 c = Av_1 + x_1 t$$

$$\vdots$$

$$W_\ell c = Av_\ell + x_\ell t$$

Opening to value $y = f(x) = \sum_{i \in [\ell]} \alpha_i x_i$ is $v_f := \sum_{i \in [\ell]} \alpha_i v_i$

Verification relation

$$\sum_{i \in [\ell]} \alpha_i W_i c = Av_f + y \cdot t$$
Security follows from \( \ell \)-succinct SIS assumption [Wee24]:

\[
\text{SIS is hard with respect to } A \text{ given a trapdoor (a basis) for the matrix}
\]

\[
B_\ell = \begin{bmatrix}
A \\
\vdots \\
A
\end{bmatrix}
\begin{bmatrix}
W_1 \\
\vdots \\
W_\ell
\end{bmatrix}
\]

where \( A \leftarrow \mathbb{Z}_{q}^{n \times m} \) and \( W_i \leftarrow \mathbb{Z}_{q}^{n \times m} \)

Falsifiable assumption but does not appear to reduce to standard SIS

\( \ell = 1 \) case does follow from plain SIS (and when \( W_i \) is very wide)

**Open problem:** Understanding security or attacks when \( \ell > 1 \)
Security follows from $\ell$-succinct SIS assumption [Wee24]:

**SIS is hard with respect to $A$ given a trapdoor (a basis) for the matrix**

$$
B_\ell = \begin{bmatrix}
A & W_1 \\
\vdots & \\
A & W_\ell
\end{bmatrix}
$$

where $A \leftarrow \mathbb{Z}_{q}^{n \times m}$ and $W_i \leftarrow \mathbb{Z}_{q}^{n \times m}$

Equivalent formulation:

**SIS is hard with respect to $A$ given $A^{-1}(W_i R)$ along with $W_i, R$**

where $A \leftarrow \mathbb{Z}_{q}^{n \times m}$, $W_i \leftarrow \mathbb{Z}_{q}^{n \times m}$, and $R \leftarrow D_{\mathbb{Z}_{s}}^{m \times k}$ where $k \geq m(\ell + 1)$
Functional Commitments from Lattices

Linear functional commitments extends readily to support (bounded-depth) circuits

\[ W_1 c = A v_1 + x_1 t \]
\[ \vdots \]
\[ W_\ell c = A v_\ell + x_\ell t \]

Supports openings to linear functions

\[ W_1 C = A V_1 + x_1 G \]
\[ \vdots \]
\[ W_\ell C = A V_\ell + x_\ell G \]

Supports openings to Boolean circuits

In this setting, \((W_1 C, \ldots, W_\ell C)\) is a [GVW14] homomorphic commitment to \(x\) (can be opened to any function \(f(x)\) of bounded depth)

Can be sampled using same trapdoor for \(B_\ell\)
(security still reduces to \(\ell\)-succinct SIS)

[see paper for details]
Summary of Functional Commitments

New methodology for constructing lattice-based commitments:

1. Write down the main verification relation \( c = A_i v_i + x_i t_i \)
2. Publish a trapdoor for the linear system induced by the verification relation

Security analysis relies on new \( q \)-type variants of SIS:

\[ SIS \text{ with respect to } A \text{ is hard given a trapdoor for a related matrix } B \]

“Random” variant of the assumption implies vector commitments and reduces to SIS

“Structured” variant (\( \ell \)-succinct SIS) implies functional commitments for circuits

• Structure also enables aggregating openings

[see paper for details]
Cryptanalysis of Lattice-Based Knowledge Assumptions
**Extractable Functional Commitments**

**Binding:** efficient adversary cannot open $\sigma$ to two different values with respect to the same $f$

Scheme could be binding, but still allow an efficient adversary to construct (malformed) commitment $\sigma$ and opening to value 1 with respect to the **all-zeroes** function.

$$\text{Verify}(\text{crs}, \sigma, (f, y_0), \pi_0) = 1$$

$$\text{Verify}(\text{crs}, \sigma, (f, y_1), \pi_1) = 1$$
**Binding:** efficient adversary cannot open $\sigma$ to two different values with respect to the same $f$

$$\pi_0 \rightarrow (f, y_0) \quad \text{Verify(crs, } \sigma, (f, y_0), \pi_0) = 1$$

$$\pi_1 \rightarrow (f, y_1) \quad \text{Verify(crs, } \sigma, (f, y_1), \pi_1) = 1$$

**Extractability:** efficient adversary that opens $\sigma$ to $y$ with respect to $f$ must know an $x$ such that $f(x) = y$

efficient extractor $\pi$ $\rightarrow$ $x$ such that $y = f(x)$

**Note:** $f$ could have multiple outputs
Extractable Functional Commitments

**Binding:** efficient adversary cannot open $\sigma$ to two different values with respect to the same $f$

Verifying:
- $\text{Verify}(\text{crs}, \sigma, (f, y_0), \pi_0) = 1$
- $\text{Verify}(\text{crs}, \sigma, (f, y_1), \pi_1) = 1$

**Extractability:** efficient adversary that opens $\sigma$ to $y$ with respect to $f$ must know an $x$ such that $f(x) = y$

- Efficient extractor
- $x$ such that $y = f(x)$

Note: $f$ could have multiple outputs

Notion is equivalent to SNARKs, so will be challenging to build from a falsifiable assumption.
Typical lattice-based knowledge assumption (to get extractable commitments / SNARKs):

\[
\begin{align*}
\mathbf{A} & \quad \mathbf{D} & \quad \mathbf{Z} & \quad \mathbf{T} \\
\text{short} & & & \text{random}
\end{align*}
\]

given (tall) matrices \(\mathbf{A}, \mathbf{D}\) and short preimages \(\mathbf{Z}\) of a random target \(\mathbf{T}\)

if adversary can produce a short vector \(\mathbf{v}\) such that \(\mathbf{A}\mathbf{v}\) is in the image of \(\mathbf{D}\) (i.e., \(\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}\)), then there exists an extractor that outputs short \(\mathbf{x}\) where \(\mathbf{v} = \mathbf{Z}\mathbf{x}\)

Observe: \(\mathbf{A}\mathbf{v}\) for a random (short) \(\mathbf{v}\) is outside the image of \(\mathbf{D}\) (since \(\mathbf{D}\) is tall)
Cryptanalysis of Lattice-Based Knowledge Assumptions

Typical lattice-based knowledge assumption (to get extractable commitments / SNARKs):

\[ \mathbf{A} \mathbf{v} \] for a random (short) \( \mathbf{v} \) is outside the image of \( \mathbf{D} \) (since \( \mathbf{D} \) is tall)

Given (tall) matrices \( \mathbf{A}, \mathbf{D} \) and short preimages \( \mathbf{Z} \) of a random target \( \mathbf{T} \)

If adversary can produce a short vector \( \mathbf{v} \) such that \( \mathbf{A} \mathbf{v} \) is in the image of \( \mathbf{D} \) (i.e., \( \mathbf{A} \mathbf{v} = \mathbf{D} \mathbf{c} \)), then there exists an extractor that outputs short \( \mathbf{x} \) where \( \mathbf{v} = \mathbf{Z} \mathbf{x} \)

Observe: \( \mathbf{A} \mathbf{v} \) for a random (short) \( \mathbf{v} \) is outside the image of \( \mathbf{D} \) (since \( \mathbf{D} \) is tall)

For extractable functional commitments:
- \( \mathbf{Z} \) is in the CRS
- Commitment is \( \mathbf{c} = \mathbf{T} \mathbf{x} \)
- Opening is \( \mathbf{v} \) where \( \mathbf{A} \mathbf{v} = \mathbf{D} \mathbf{c} \)

Extractable since valid opening can be associated with an honestly-generated commitment
Obliviously Sampling a Solution

Typical lattice-based knowledge assumption (to get extractable commitments / SNARKs):

\[ \mathbf{A} \mathbf{v} = \mathbf{D} \mathbf{c} \]

Our work: algorithm to obliviously sample a solution \( \mathbf{A} \mathbf{v} = \mathbf{D} \mathbf{c} \) without knowledge of a linear combination \( \mathbf{v} = \mathbf{Z} \mathbf{x} \)

Rewrite \( \mathbf{A} \mathbf{Z} = \mathbf{D} \mathbf{T} \) as

\[ [\mathbf{A} | \mathbf{D} \mathbf{G}] \cdot \begin{bmatrix} \mathbf{Z} \\ -\mathbf{G}^{-1}(\mathbf{T}) \end{bmatrix} = 0 \]

If \( \mathbf{Z} \) and \( \mathbf{T} \) are wide enough, we (heuristically) obtain a basis for \( [\mathbf{A} | \mathbf{D} \mathbf{G}] \)

[ACLMT22]
Our work: algorithm to obliviously sample a solution $Av = Dc$ without knowledge of a linear combination $v = Zx$

Rewrite $AZ = DT$ as

$$[A \mid DG] \cdot \begin{bmatrix} Z \\ -G^{-1}(T) \end{bmatrix} = 0$$

If $Z$ and $T$ are wide enough, we (heuristically) obtain a basis for $[A \mid DG]$

Oblivious sampler (Babai rounding):

1. Take any (non-zero) integer solution $y$ where $[A \mid DG]y = 0 \mod q$
2. Assuming $B^*$ is full-rank over $\mathbb{Q}$, find $z$ such that $B^*z = y$ (over $\mathbb{Q}$)
3. Set $y^* = y - B^*[z] = B^*(z - \lfloor z \rfloor)$ and parse into $v, c$

Correctness: $[A \mid DG] \cdot y^* = [A \mid DG] \cdot B^*(z - \lfloor z \rfloor) = 0 \mod q$ and $y^*$ is short
This work: algorithm to obliviously sample a solution $Av = Dc$ without knowledge of a linear combination $v = Zx$

Rewrite $AZ = DT$ as

$$[A \mid DG] \cdot \begin{bmatrix} Z \\ -G^{-1}(T) \end{bmatrix} = 0$$

If $Z$ and $T$ are wide enough, we (heuristically) obtain a basis for $[A \mid DG]$

Oblivious sampler (Babai rounding):

1. Take any (non-zero) integer solution $y$ where $[A \mid DG]y = 0 \mod q$
2. Assuming $B^*$ is full-rank over $\mathbb{Q}$, find $z$ such that $B^*z = y$ (over $\mathbb{Q}$)
3. Set $y^* = y - B^*[z] = B^*z - z$

This solution is obtained by “rounding” off a long solution

Question: Can we explain such solutions as taking a short linear combination of $Z$ (i.e., what the knowledge assumption asserts)

Correctness: $[A \mid DG] \cdot y^* = [A \mid DG] \cdot B^*(z - [z]) = 0 \mod q$ and $y^*$ is short
1. Start with the key verification relation (i.e., knowledge of a short solution to a linear system)
2. Express verification relation as finding non-zero vector in the kernel of a lattice defined by the verification equation
3. Use components in the CRS to derive a basis for the related lattice

\[ A \nu = Dc \]

\[ \begin{bmatrix} A | DG \end{bmatrix} \begin{bmatrix} \nu \\ -G^{-1}(c) \end{bmatrix} = 0 \]

\[ [A | DG] \cdot \begin{bmatrix} \mathbb{Z} \\ -G^{-1}(T) \end{bmatrix} = 0 \]
Template for Analyzing Lattice-Based Knowledge Assumptions

1. Start with the key verification relation (i.e., knowledge of a short solution to a linear system)
2. Express verification relation as finding non-zero vector in the kernel of a lattice defined by the verification equation
3. Use components in the CRS to derive a basis for the related lattice

Implications:
- Oblivious sampler for integer variant of knowledge $k$-$R$-ISIS assumption from [ACLMT22]
  Implementation by Martin Albrecht: https://gist.github.com/malb/7c8b86520c675560be62eda98dab2a6f
- Breaks extractability of the (integer variant of the) linear functional commitment from [ACLMT22] assuming hardness of inhomogeneous SIS (i.e., existence of efficient extractor for oblivious sampler implies algorithm for inhomogeneous SIS)

Open question: Can we extend the attacks to break soundness of the SNARK?
Template for Analyzing Lattice-Based Knowledge Assumptions

1. Start with the key verification relation (i.e., knowledge of a short solution to a linear system)
2. Express verification relation as finding non-zero vector in the kernel of a lattice defined by the verification equation
3. Use components in the CRS to derive a basis for the related lattice

Implications:
- Oblivious sampler for integer variant of knowledge assumption from [ACLMT22]
- Breaks extractability of the linear functional commitment from [ACLMT22] assuming hardness of inhomogeneous SIS (i.e., existence of efficient extractor for oblivious sampler implies algorithm for inhomogeneous SIS)

Open question: Can we extend the attacks to break soundness of the SNARK?

The SNARK considers extractable commitment for quadratic functions while our current oblivious sampler only works for linear functions in the case of [ACLMT22]
Open Questions

Understanding the hardness of $\ell$-succinct SIS/LWE (hardness reductions or cryptanalysis)?

Martin Albrecht’s blog post: https://malb.io/sis-with-hints.html

New applications of $\ell$-succinct SIS/LWE?

Broadcast encryption, succinct ABE, succinct laconic function evaluation [Wee24]

Cryptanalysis of lattice-based SNARKs based on knowledge $k$-$R$-ISIS [ACLMT22, CLM23, FLV23]

Our oblivious sampler (heuristically) falsifies the assumption, but does not break existing constructions

Formulation of new lattice-based knowledge assumptions that avoids attacks

Thank you!

https://eprint.iacr.org/2022/1515
https://eprint.iacr.org/2024/028