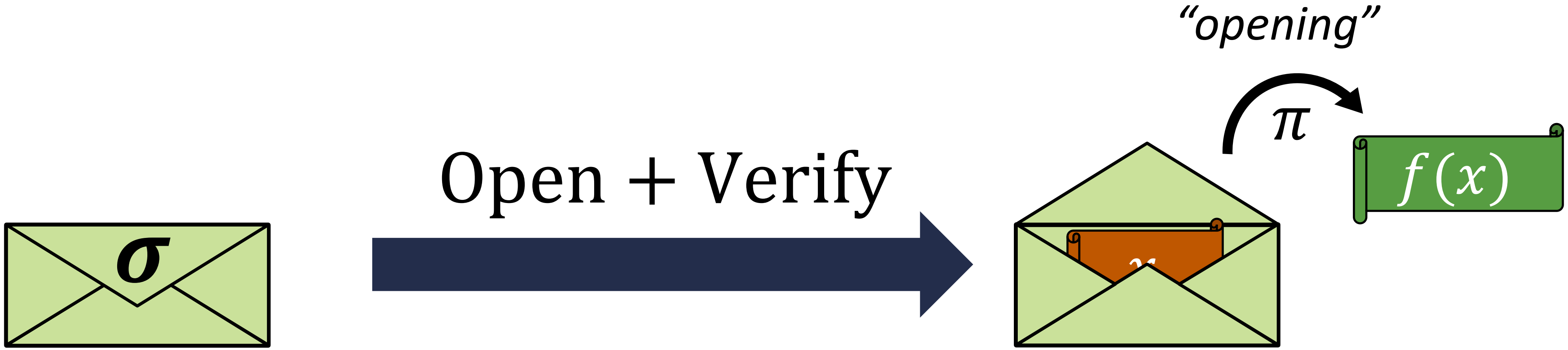
Lattice-Based Functional Commitments: Fast Verification and Cryptanalysis

Hoeteck Wee and David Wu

December 2023

Functional Commitments



Functional Commitments

$$\mathrm{Commit}(\mathrm{crs}, x) \rightarrow (\sigma, \mathrm{st})$$

Takes a common reference string and commits to an input x

Outputs commitment σ and commitment state st

Functional Commitments

$\mathrm{Commit}(\mathrm{crs}, x) \rightarrow (\sigma, \mathrm{st})$

$\mathrm{Open}(\mathrm{st}, f) \rightarrow \pi$

Takes the commitment state and a function f and outputs an opening π

$\mathrm{Verify}(\mathrm{crs}, \sigma, (f, y), \pi) \rightarrow 0/1$

Checks whether π is valid opening of σ to value y with respect to f

Functional Commitments

Can also consider the dual notion where user commits to the function f and opens at an input x to the value $f(x)$

$$\text{Commit}(\text{crs}, f) \to (\sigma, \text{st})$$

$$\text{Open}(\text{st}, x) \to \pi$$

Takes the commitment state and an input x and outputs an opening π

$$\text{Verify}(\text{crs}, \sigma, (x, y), \pi) \to 0/1$$

Checks whether π is valid opening of σ to value y at input x

Functional Commitments

$$\mathsf{Commit}(\mathsf{crs}, f) \to (\sigma, \mathsf{st})$$

$$\mathsf{Open}(\mathsf{st}, x) \to \pi$$

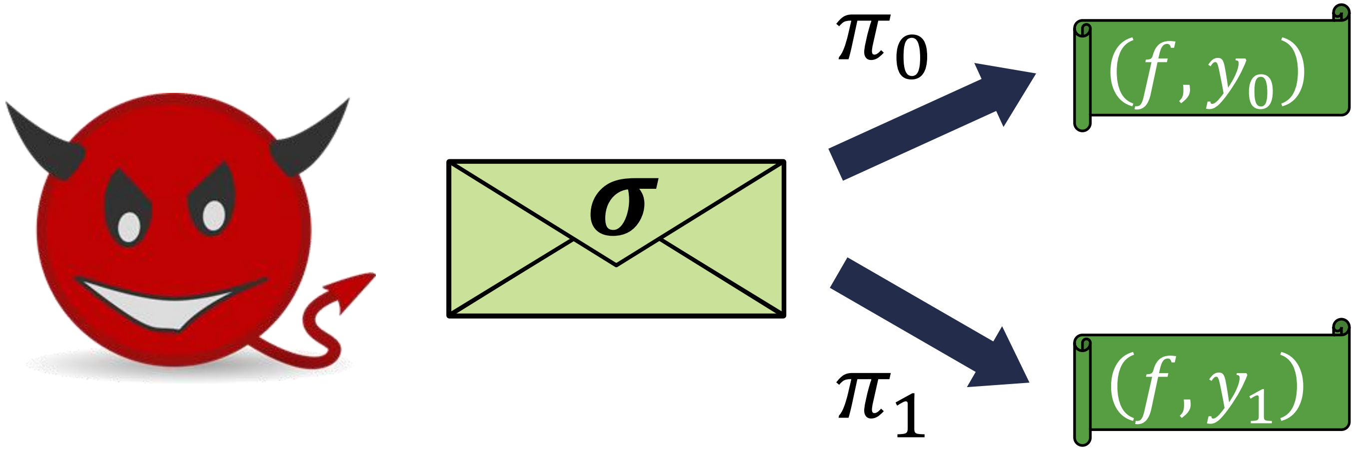
Can also consider the dual notion where user commits to the function f and opens at an input x to the value $f(x)$

Takes the commitment state and an input x and outputs an opening π

This talk: will just focus on the first notion (commit to x, open to f)

Functional Commitments

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



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

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

Functional Commitments

Succinctness: commitments and openings should be short

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

Will consider relaxation where $|\sigma|$ and $|\pi|$ can grow with **depth** of the circuit computing f

Fast verification: can preprocess f into a short verification key vk_f so that "online" verification runs in time $\mathrm{poly}(\lambda, \log|x|, d)$ where d is the depth of f

Functional Commitments

Succinctness: commitments and openings should be short

- **Short commi**
- **Short openin**

Note: having short commitments + openings does not imply fast verification (e.g., verification procedure in [WW23] basically evaluates f on the commitment)

Fast verification: can preprocess f into a short verification key vk_f so that "online" verification runs in time $\mathrm{poly}(\lambda, \log|x|, d)$ where d is the depth of f

Lattice-Based Functional Commitments

Scheme	Function Class	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	FV	BB	Assumption
[KLVW23]	Boolean circuits	1	1	1	✓	✗	LWE
[BCFL23]	width-w, depth-d circuits	w^5	1	1	✓	✓	twin-k-M-ISIS
[WW23]	depth-d circuits	ℓ^2	1	1	✗	✓	$\mathrm{BASIS}_{\mathrm{struct}}$
[ACLMT22]	degree-d polynomials	ℓ^{2d}	1	1	✓	✓	k-R-ISIS
[BCFL23]*	degree-d polynomials	ℓ^{5d}	1	1	✓	✓	twin-k-R-ISIS
This work	degree-d polynomials	ℓ^{d+1}	1	1	✓	✓	$O(\ell^d)$-succinct SIS

- ℓ is the input length
- **FV:** scheme supports fast verification
- **BB:** scheme only makes black-box use of cryptography

*can decrease CRS size at the cost of longer openings

Comparisons ignore all $\mathrm{poly}(\lambda, d, \log \ell)$ terms

This talk: only consider **lattice-based** functional commitment schemes

Lattice-Based Functional Commitments

Scheme	Function Class	$\|\text{crs}\|$	$\|\sigma\|$	$\|\pi\|$	FV	BB	Assumption
[KLVW23]	Boolean circuits	1	1	1	✓	✗	LWE
[BCFL23]	width-w, depth-d circuits	w^5	1	1	✓	✓	twin-k-M-ISIS
[WW23]	depth-d circuits	ℓ^2	1	1	✗	✓	$\text{BASIS}_{\text{struct}}$
[ACLMT22]	degree-d polynomials	ℓ^{2d}	1	1	✓	✓	k-R-ISIS
[BCFL23]*	degree-d polynomials	ℓ^{5d}	1	1	✓	✓	twin-k-R-ISIS
This work	degree-d polynomials	ℓ^{d+1}	1	1	✓	✓	$O(\ell^d)$-succinct SIS

Concurrent works:
- [FLV23]: polynomial commitment with linear-size CRS from k-R-ISIS assumption
- [CLM23]: functional commitment for quadratic functions with linear linear-size CRS from vanishing SIS

Lattice-Based Functional Commitments

Scheme	Function Class	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	FV	BB	Assumption
[KLVW23]	Boolean circuits	1	1	1	✓	✗	LWE
[BCFL23]	width-w, depth-d circuits	w^5	1	1	✓	✓	twin-k-M-ISIS
[WW23]	depth-d circuits	ℓ^2	1	1	✗	✓	$\mathrm{BASIS}_{\mathrm{struct}}$
[ACLMT22]	degree-d polynomials	ℓ^{2d}	1	1	✓	✓	k-R-ISIS
[BCFL23]*	degree-d polynomials	ℓ^{5d}	1	1	✓	✓	twin-k-R-ISIS
This work	degree-d polynomials	ℓ^{d+1}	1	1	✓	✓	$O(\ell^d)$-succinct SIS

functional commitments

Scheme	Function Class	$\lvert\mathrm{crs}\rvert$	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	FV	BB	Assumption
[KLVW23]	Boolean circuits	1	1	1	✓	✗	LWE
[dCP23]	depth-d circuits	ℓ	1	ℓ	✗	✓	SIS
This work	depth-d circuits	ℓ^2	1	1	✓	✓	ℓ-succinct SIS

dual functional commitments

This talk: only consider **lattice-based** functional commitment schemes

This Work

Functional commitments with fast verification (and black-box use of cryptography)

- Functional commitment for degree-d polynomials with $O(\ell^{d+1})$-size CRS
  Previously: $O(\ell^{2d})$-size CRS

This talk

- Dual functional commitment for (bounded-depth) Boolean circuits
  First construction to support fast verification (without non-black-box use of cryptography)

Cryptanalysis of **knowledge** versions of the new lattice assumptions

- Construct oblivious sampler that (heuristically) falsifies the knowledge k-R-ISIS assumption in [ACLMT22]
- Approach breaks extractability of several lattice-based functional commitments (our construction and the [ACLMT22] extractable commitment for linear functions)

This talk

Attacks do not break standard binding security of the commitment nor does it (currently) give an attack on the SNARK candidates based on knowledge k-R-ISIS [ACLMT22, CLM23, FLV23] – but does break the underlying knowledge assumption for these SNARK candidates

Starting Point: the Wee-Wu Functional Commitment

[WW23]

Common reference string (CRS)

$\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ $\quad$ $\boldsymbol{W}_1 \in \mathbb{Z}_q^{n \times m}$ $\quad \cdots \quad$ $\boldsymbol{W}_\ell \in \mathbb{Z}_q^{n \times m}$ $\quad$ trapdoor for matrix related to $\boldsymbol{A}, \boldsymbol{W}_1, \dots, \boldsymbol{W}_\ell$

Commitment relation (for all $i \in [\ell]$)

$$\boldsymbol{W}_i \boldsymbol{C} = \boxed{x_i} \boldsymbol{G} - \boldsymbol{A} \boldsymbol{V}_i$$

commitment $\quad$ **gadget matrix** $\quad$ **opening**

(matrix with short entries)

Trapdoor in CRS allow for joint sampling of $(\boldsymbol{C}, \boldsymbol{V}_1, \dots, \boldsymbol{V}_\ell)$

Structure does not support fast verification for polynomials of degree $d > 1$

commitment to ℓ-dimensional vectors $\boldsymbol{x} \in \{0,1\}^\ell$

Our Approach: A “Chaining” Structure

Common reference string (CRS)

$$\begin{array}{cccc} \boldsymbol{A} & \boldsymbol{W}_1 & \cdots & \boldsymbol{W}_\ell \\ & \boldsymbol{W}_{1,1} & \cdots & \boldsymbol{W}_{1,\ell} \\ & \vdots & & \vdots \\ & \boldsymbol{W}_{\ell,1} & \cdots & \boldsymbol{W}_{\ell,\ell} \end{array}$$

trapdoor for matrix related to $\boldsymbol{A}, \boldsymbol{W}_i, \boldsymbol{W}_{ij}$

More structure in the CRS

[WW23] relation: $\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$

This work: $\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$

$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$

Will also assume require that $\boldsymbol{C}$ be a short matrix

Our Approach: A “Chaining” Structure

$$\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^{\ell}$, we construct an opening to $x_i x_j$ as follows:

$$\boldsymbol{W}_{ij} \boldsymbol{C}^2$$

function of commitment
and public parameters

Our Approach: A "Chaining" Structure

$$\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boxed{\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^\ell$, we construct an opening to $x_i x_j$ as follows:

$$\underbrace{\boldsymbol{W}_{ij} \boldsymbol{C}^2}_{\text{function of commitment and public parameters}} = \left(x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}\right)\boldsymbol{C}$$

Our Approach: A "Chaining" Structure

$$\boldsymbol{W}_i\boldsymbol{C} = x_i\boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij}\boldsymbol{C} = x_i\boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^\ell$, we construct an opening to $x_i x_j$ as follows:

$$\boldsymbol{W}_{ij}\boldsymbol{C}^2 = \left(x_i\boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}\right)\boldsymbol{C}$$

function of commitment
and public parameters

$$= x_i\boldsymbol{W}_j\boldsymbol{C} - \boldsymbol{A}\boldsymbol{V}_{ij}\boldsymbol{C}$$

Our Approach: A "Chaining" Structure

$$\boxed{\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i}$$

$$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^\ell$, we construct an opening to $x_i x_j$ as follows:

$$\begin{aligned} \boldsymbol{W}_{ij}\boldsymbol{C}^2 &= (x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij})\boldsymbol{C} \\ &= x_i \boldsymbol{W}_j \boldsymbol{C} - \boldsymbol{A}\boldsymbol{V}_{ij}\boldsymbol{C} \\ &= x_i x_j \boldsymbol{G} - \boldsymbol{A}(\boldsymbol{V}_{ij}\boldsymbol{C} + x_i \boldsymbol{V}_j) \end{aligned}$$

function of commitment and public parameters

opening for $x_i x_j$
(short if $\boldsymbol{C}, \boldsymbol{V}_i, \boldsymbol{V}_j, x_i$ short)

Our Approach: A "Chaining" Structure

$$\boldsymbol{W}_i\boldsymbol{C} = x_i\boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij}\boldsymbol{C} = x_i\boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^\ell$, we construct an opening to $x_i x_j$ as follows:

$$\underbrace{\boldsymbol{W}_{ij}\boldsymbol{C}^2}_{\text{function of commitment and public parameters}} = \left(x_i\boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}\right)\boldsymbol{C}$$

$$= x_i\boldsymbol{W}_j\boldsymbol{C} - \boldsymbol{A}\boldsymbol{V}_{ij}\boldsymbol{C}$$

$$= x_i x_j\boldsymbol{G} - \boldsymbol{A}\underbrace{\left(\boldsymbol{V}_{ij}\boldsymbol{C} + x_i\boldsymbol{V}_j\right)}_{\text{opening for } x_i x_j \text{ (short if } \boldsymbol{C}, \boldsymbol{V}_i, \boldsymbol{V}_j, x_i \text{ short)}}$$

Verification procedure: compute $\boldsymbol{W}_{ij}\boldsymbol{C}^2$ and check above relation

Our Approach: A “Chaining” Structure

$$\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Given commitment $\boldsymbol{C}$ to $\boldsymbol{x} \in \{0,1\}^\ell$, we construct an opening to $x_i x_j$ as follows:

$\boldsymbol{W}_{ij}\boldsymbol{C}^2$

function of commitment and public parameters

$$= \left(x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}\right)\boldsymbol{C}$$

$$= x_i \boldsymbol{W}_j \boldsymbol{C} - \boldsymbol{A}\boldsymbol{V}_{ij}\boldsymbol{C}$$

$$= x_i x_j \boldsymbol{G} - \boldsymbol{A}\left(\boldsymbol{V}_{ij}\boldsymbol{C} \quad \right)$$

Online verification just computes $\boldsymbol{W}_f \boldsymbol{C}^2$, which is independent of input length ℓ

Verification procedure: compute $\boldsymbol{W}_{ij}\boldsymbol{C}^2$ … e

Can precompute $\boldsymbol{W}_f = \sum_{i,j} \gamma_{ij} \boldsymbol{W}_{ij}$

To open to $f(\boldsymbol{x}) = \sum_{i,j} \gamma_{ij} x_i x_j$, verifier computes $\sum_{i,j} \gamma_{ij} \boldsymbol{W}_{ij} \boldsymbol{C}^2$

How to Construct C, V_i, V_{ij}?

$$\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Approach: sample trapdoor for following matrix

$$\begin{bmatrix} \boldsymbol{A} & & & & & & \boldsymbol{W}_1 \\ & \ddots & & & & & \vdots \\ & & \boldsymbol{A} & & & & \boldsymbol{W}_\ell \\ & & & \boldsymbol{A} & & & \boldsymbol{W}_{11} \\ & & & & \ddots & & \vdots \\ & & & & & \boldsymbol{A} & \boldsymbol{W}_{\ell\ell} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_\ell \\ \boldsymbol{V}_{11} \\ \vdots \\ \boldsymbol{V}_{\ell\ell} \\ \boldsymbol{C} \end{bmatrix} = \begin{bmatrix} x_1 \boldsymbol{G} \\ \vdots \\ x_\ell \boldsymbol{G} \\ x_1 \boldsymbol{W}_1 \\ \vdots \\ x_\ell \boldsymbol{W}_\ell \end{bmatrix}$$

Size of full trapdoor: $O(\ell^4)$

can use the trapdoor to sample $\boldsymbol{C}, \boldsymbol{V}_i, \boldsymbol{V}_{ij}$ that satisfies relation for any $\boldsymbol{x}$

How to Construct $\boldsymbol{C}, \boldsymbol{V}_i, \boldsymbol{V}_{ij}$?

$$\boldsymbol{W}_i \boldsymbol{C} = x_i \boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_i$$

$$\boldsymbol{W}_{ij} \boldsymbol{C} = x_i \boldsymbol{W}_j - \boldsymbol{A}\boldsymbol{V}_{ij}$$

Approach: sample trapdoor for following matrix

$$\begin{bmatrix} \boldsymbol{A} & & & & & & \boldsymbol{W}_1 \\ & \ddots & & & & & \vdots \\ & & \boldsymbol{A} & & & & \boldsymbol{W}_\ell \\ & & & \boldsymbol{A} & & & \boldsymbol{W}_{11} \\ & & & & \ddots & & \vdots \\ & & & & & \boldsymbol{A} & \boldsymbol{W}_{\ell\ell} \end{bmatrix}$$

Size of full trapdoor: $O(\ell^4)$

Opening relations are linear:
if $\boldsymbol{C}_1$ is a commitment to x_1 and $\boldsymbol{C}_2$ is a commitment to x_2, then $\boldsymbol{C}_1 + \boldsymbol{C}_2$ is a commitment to $x_1 + x_2$

Instead of publishing full trapdoor, publish commitments $\boldsymbol{C}$ and openings $V_1, \ldots, V_\ell, V_{11}, \ldots, V_{\ell\ell}$ to ℓ basis vectors

Shorter CRS: leverage homomorphism

Size of CRS: $O(\ell^3)$

Evaluation Binding

ℓ**-succinct SIS** [Wee23]**:** SIS is hard with respect to $\boldsymbol{A}$ even given the trapdoor for the matrix

$$\begin{bmatrix} \boldsymbol{A} & & & & & & \boldsymbol{W}_1 \\ & \ddots & & & & & \vdots \\ & & \boldsymbol{A} & & & & \boldsymbol{W}_\ell \\ & & & \boldsymbol{A} & & & \boldsymbol{W}_{11} \\ & & & & \ddots & & \vdots \\ & & & & & \boldsymbol{A} & \boldsymbol{W}_{\ell\ell} \end{bmatrix}$$

The $\boldsymbol{W}_i$'s and $\boldsymbol{W}_{ij}$'s are **uniform random**

Assumption has less structure than BASIS assumption from [WW23] and k-R-ISIS assumption from [ACLMT22]

Trapdoor for above matrix suffices to simulate CRS

Can show that adversary that breaks evaluation binding solves SIS with respect to $\boldsymbol{A}$

[see paper for details]

Conclusion: functional commitment for degree-d polynomials with fast verification and $O(\ell^{d+1})$-size CRS from $O(\ell^d)$-succinct SIS

Evaluation Binding

ℓ-succinct SIS [Wee23]**:** SIS is hard with respect to $\boldsymbol{A}$ even given the trapdoor for the matrix

$$\begin{bmatrix} \boldsymbol{A} & & & & & & \boldsymbol{W}_1 \\ & \ddots & & & & & \vdots \\ & & \boldsymbol{A} & & & & \boldsymbol{W}_\ell \\ & & & \boldsymbol{A} & & & \boldsymbol{W}_{11} \\ & & & & \ddots & & \vdots \\ & & & & & \boldsymbol{A} & \boldsymbol{W}_{\ell\ell} \end{bmatrix}$$

The $\boldsymbol{W}_i$'s and $\boldsymbol{W}_{ij}$'s are **uniform random**

Assumption has less structure than BASIS assumption from [WW23] and k-R-ISIS assumption from [ACLMT22]

Trapdoor for above matrix suffices to simulate CRS

Can show that adversary that breaks evaluation binding solv

Previous (black-box) lattice-based constructions with fast verification: $O(\ell^{2d})$-size CRS

Conclusion: functional commitment for degree-d polynomials with fast verification and $O(\ell^{d+1})$-size CRS from $O(\ell^d)$-succinct SIS

Cryptanalysis of Lattice-Based Knowledge Assumptions

Cryptanalysis of Lattice-Based Knowledge Assumptions

Typical lattice-based knowledge assumption (to get extractable commitment / SNARK):

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

the only way an adversary can produce a short vector $\boldsymbol{v}$ such that $\boldsymbol{A}\boldsymbol{v}$ is in the image of $\boldsymbol{D}$ (i.e., $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{D}\boldsymbol{c}$) is by setting $\boldsymbol{v} = \boldsymbol{Z}\boldsymbol{x}$

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

Obliviously Sampling a Solution

Typical lattice-based knowledge assumption (to get extractable commitment / SNARK):

This work: algorithm to obliviously sample a solution $\boldsymbol{Av} = \boldsymbol{Dc}$ without knowledge of a linear combination $\boldsymbol{v} = \boldsymbol{Zx}$

Rewrite $\boldsymbol{AZ} = \boldsymbol{DT}$ as

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

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

Obliviously Sampling a Solution

This work: algorithm to obliviously sample a solution $\boldsymbol{Av} = \boldsymbol{Dc}$ without knowledge of a linear combination $\boldsymbol{v} = \boldsymbol{Zx}$

Rewrite $\boldsymbol{AZ} = \boldsymbol{DT}$ as

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

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

Oblivious sampler (Babai rounding):

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

Correctness: $[\boldsymbol{A} \mid \boldsymbol{DG}] \cdot \boldsymbol{y}^* = [\boldsymbol{A} \mid \boldsymbol{DG}] \cdot \boldsymbol{B}^*(\boldsymbol{z} - \lfloor \boldsymbol{z} \rceil) = \boldsymbol{0} \bmod q$ and $\boldsymbol{y}^*$ is short

Obliviously Sampling a Solution

This work: algorithm to obliviously sample a solution $\boldsymbol{Av} = \boldsymbol{Dc}$ without knowledge of a linear combination $\boldsymbol{v} = \boldsymbol{Zx}$

Rewrite $\boldsymbol{AZ} = \boldsymbol{DT}$ as

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

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

Oblivious sampler (Babai roun

1. Take a long integer solut
2. Assuming $\boldsymbol{B}^*$ is full-rank
3. Set $\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{B}^*\lfloor \boldsymbol{z} \rceil = \boldsymbol{B}$

This solution is obtained by "rounding" off a <u>long</u> solution

Question: Can we explain such solutions as taking a <u>short</u> linear combination of $\boldsymbol{Z}$ (i.e., what the knowledge assumption asserts)

Correctness: $[\boldsymbol{A} \mid \boldsymbol{DG}] \cdot \boldsymbol{y}^* = [\boldsymbol{A} \mid \boldsymbol{DG}] \cdot \boldsymbol{B}^*(\boldsymbol{z} - \lfloor \boldsymbol{z} \rceil) = \boldsymbol{0} \bmod q$ and $\boldsymbol{y}^*$ is short

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

1

$$\boldsymbol{A}\boldsymbol{v} = \boldsymbol{D}\boldsymbol{c}$$

2

$$[\boldsymbol{A} \mid \boldsymbol{D}\boldsymbol{G}] \begin{bmatrix} \boldsymbol{v} \\ -\boldsymbol{G}^{-1}(\boldsymbol{c}) \end{bmatrix} = \boldsymbol{0}$$

3

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

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 our functional commitment scheme for quadratic functions (i.e., obliviously sample a commitment $\boldsymbol{c}$ and openings to $x_1^2 = 0, x_1 x_2 = 1$)
- 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 k-R-ISIS assumption from [ACLMT22]
  Implementation by Martin Albrecht
- Breaks extractability of our
  obliviously sample a commi
- Breaks extractability of the
  [ACLMT22] assuming hardn
  for oblivious sampler implies algorithm for inhomogeneous SIS)

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

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

This Work

Functional commitments with fast verification (and black-box use of cryptography)

- Functional commitment for degree-d polynomials with $O(\ell^{d+1})$-size CRS
  Previously: $O(\ell^{2d})$-size CRS
- Dual functional commitment for (bounded-depth) Boolean circuits
  First construction to support fast verification (without non-black-box use of cryptography)

[see paper for details]

Cryptanalysis of **knowledge** versions of the new lattice assumptions

- Construct oblivious sampler that (heuristically) falsifies the knowledge k-R-ISIS assumption in [ACLMT22]
- Approach breaks extractability of several lattice-based functional commitments (our construction and the [ACLMT22] extractable commitment for linear functions)

Open Questions

(Black-box) functional commitments with fast verification from standard SIS?

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 our attacks

Thank you!