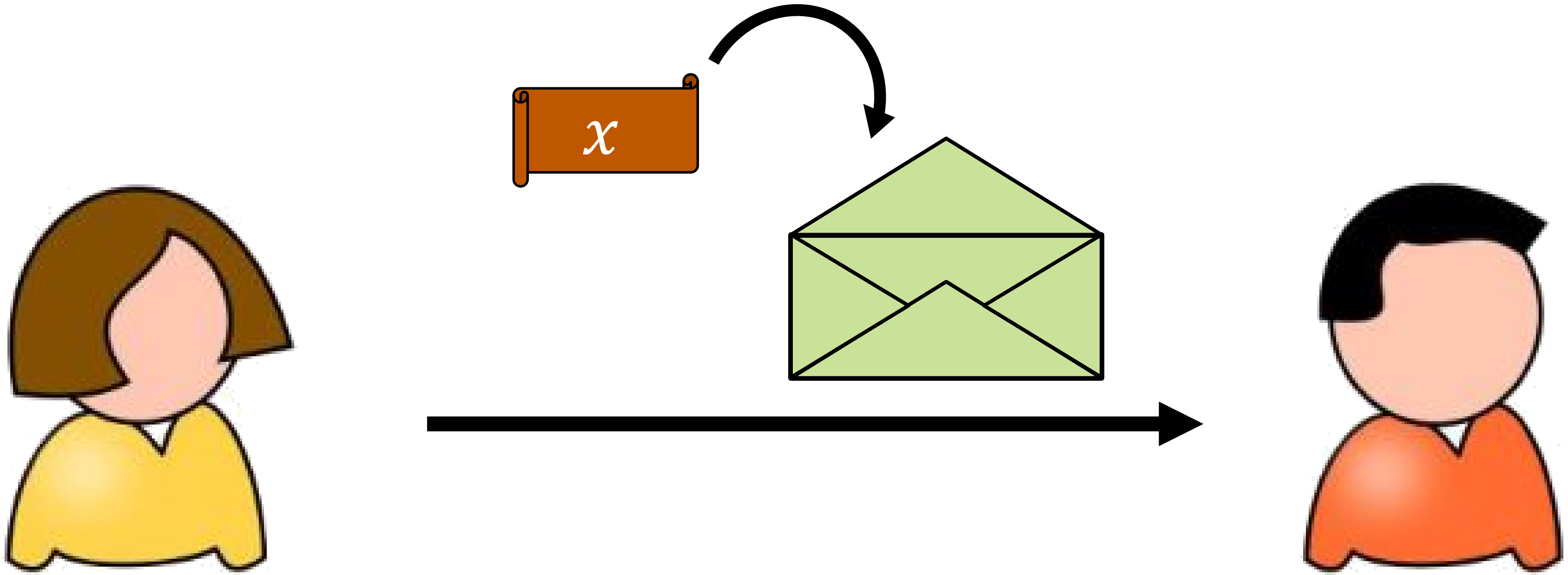
# Succinct Vector, Polynomial, and Functional Commitments from Lattices

Hoeteck Wee and David Wu

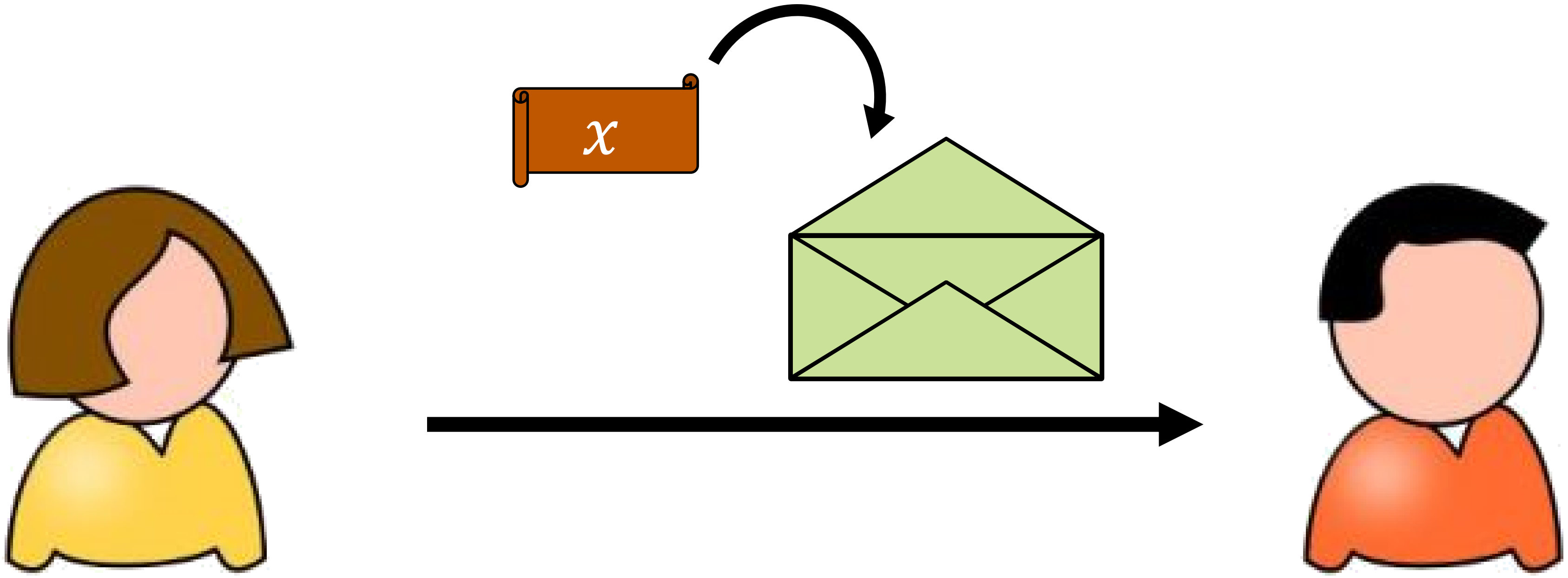
March 2023

# Commitment Schemes



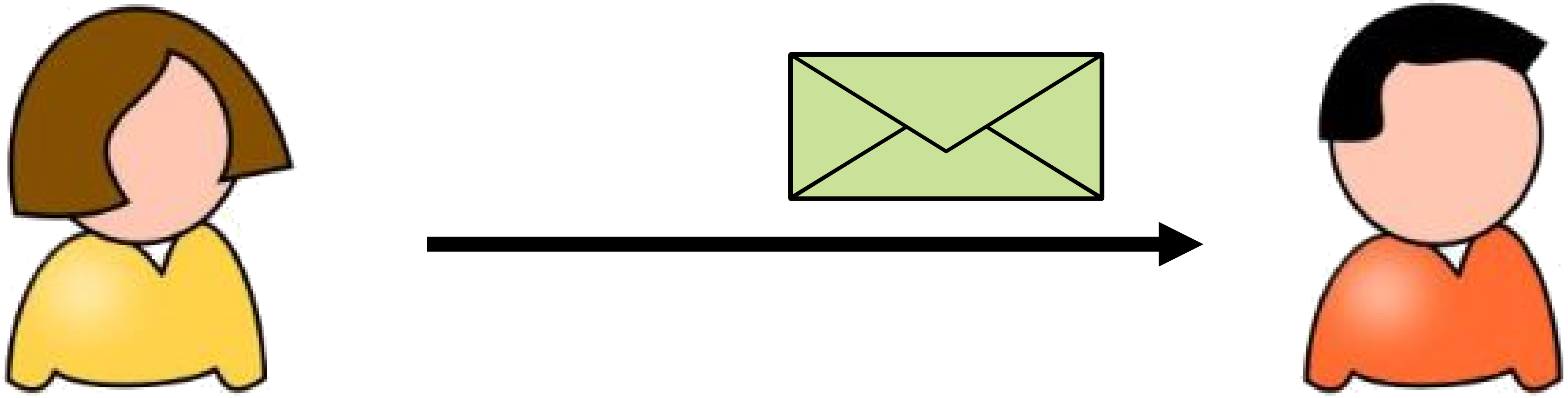

*cryptographic analog of a sealed envelope*

# Commitment Schemes




*cryptographic analog of a sealed envelope*

# Commitment Schemes



*cryptographic analog of a sealed envelope*

# Commitment Schemes


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

Takes a common reference string and commits to a message

Outputs commitment $\sigma$ and commitment state st

# Commitment Schemes


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

$\text{Open}(\text{st}) \rightarrow \pi$

Takes the commitment state and outputs an opening $\pi$

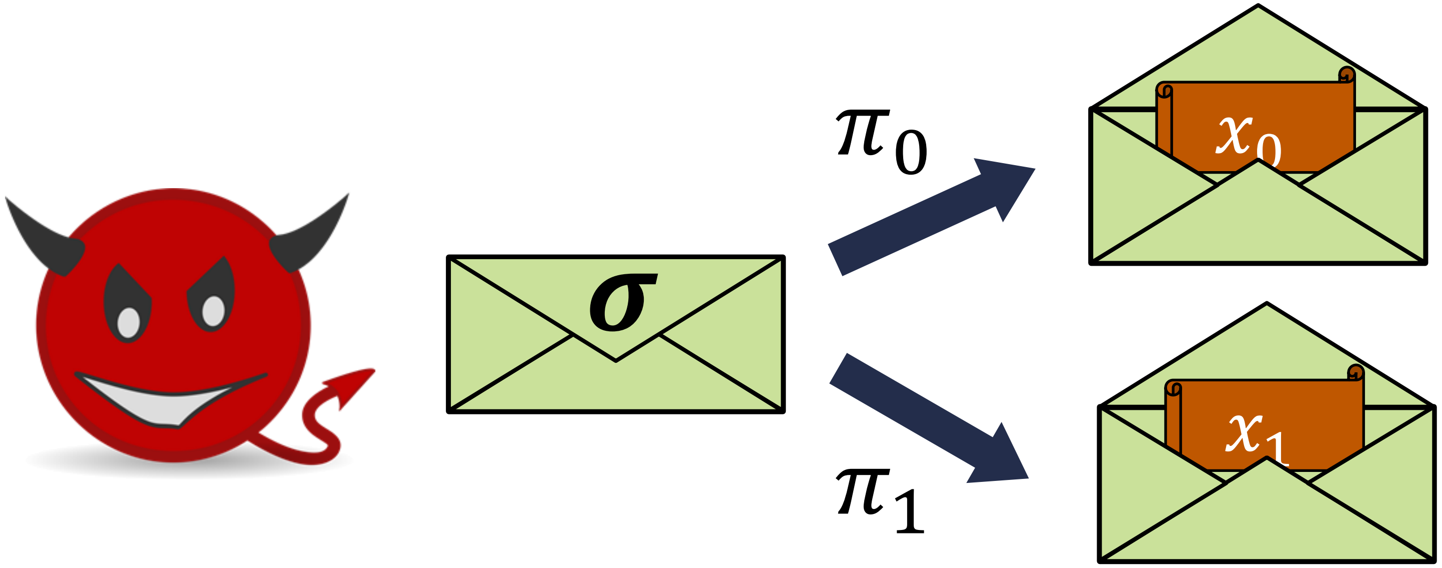
$\text{Verify}(\text{crs}, \sigma, x, \pi) \rightarrow 0/1$

Checks whether $\pi$ is valid opening of $\sigma$ to $x$

***Alternatively:*** *Could define* Commit *to output* $(\sigma, \pi)$ *and remove* Open

# Commitment Schemes


**Binding:** efficient adversary cannot open $\sigma$ to two different values

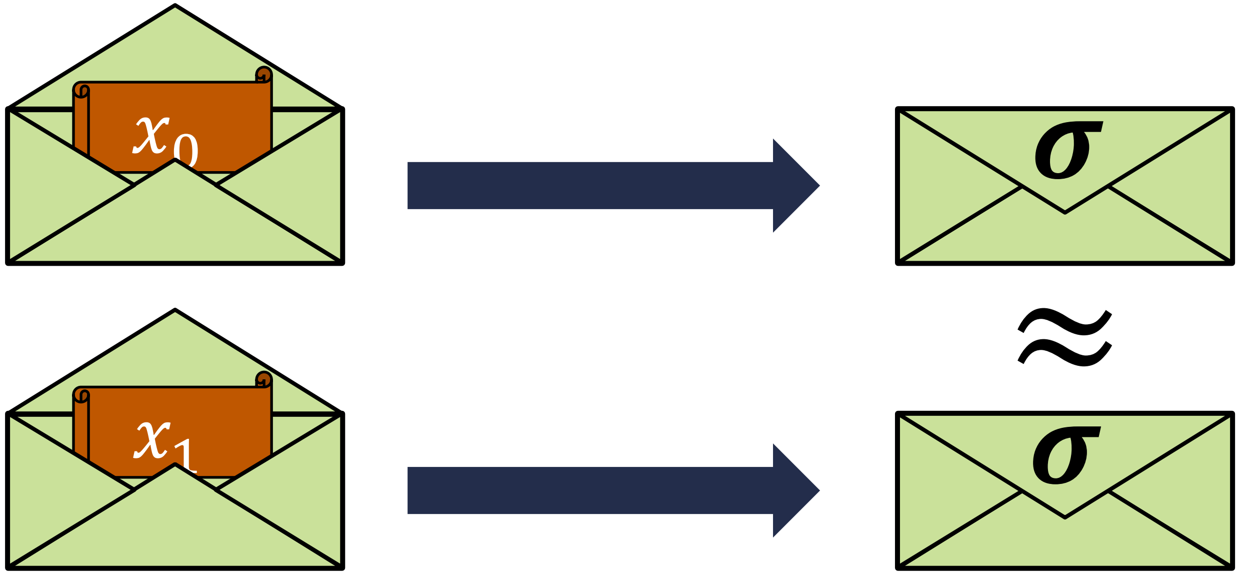



$$\mathrm{Verify}(\mathrm{crs}, \sigma, x_0, \pi_0) = 1$$

$$\mathrm{Verify}(\mathrm{crs}, \sigma, x_1, \pi_1) = 1$$

# Commitment Schemes


**Hiding:** the commitment $\sigma$ hides the input $x$




$$\mathrm{Commit}(\mathrm{crs}, x_0) \approx \mathrm{Commit}(\mathrm{crs}, x_1)$$

# This Talk: Succinct Functional Commitments


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

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

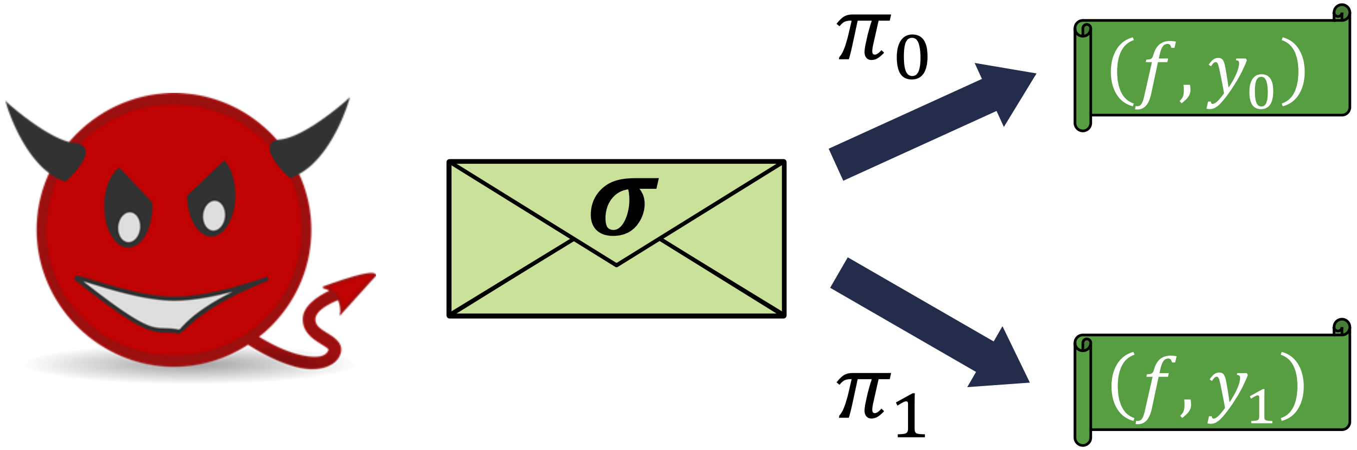
Takes the commitment state and a function $f$ and outputs an opening $\pi$

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

Checks whether $\pi$ is valid opening of $\sigma$ to value $y$ with respect to $f$

# This Talk: Succinct Functional Commitments


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

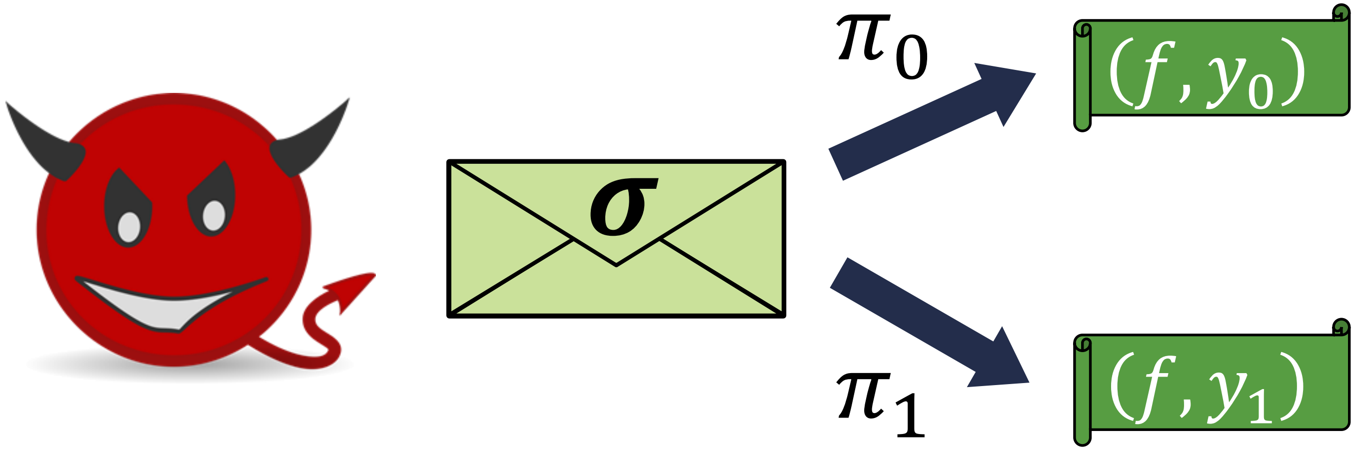



$$\text{Verify}(\text{crs}, \sigma, (f, y_0), \pi_0) = 1$$

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

# This Talk: Succinct Functional Commitments


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




**Note:** successful opening of $\sigma$ to $y$ with respect to $f$ does *not* mean there exists $x$ where $y = f(x)$

# This Talk: Succinct Functional Commitments


**Hiding:** commitment $\sigma$ and opening $\pi$ only reveal $f(x)$

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

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

# Special Cases of Functional Commitments

**Vector commitments:**

$[x_1, x_2, \dots, x_n]$ $\xrightarrow{\mathrm{ind}_i(x_1, \dots, x_n) \coloneqq x_i}$ $x_i$

*commit to a vector, open at an index*

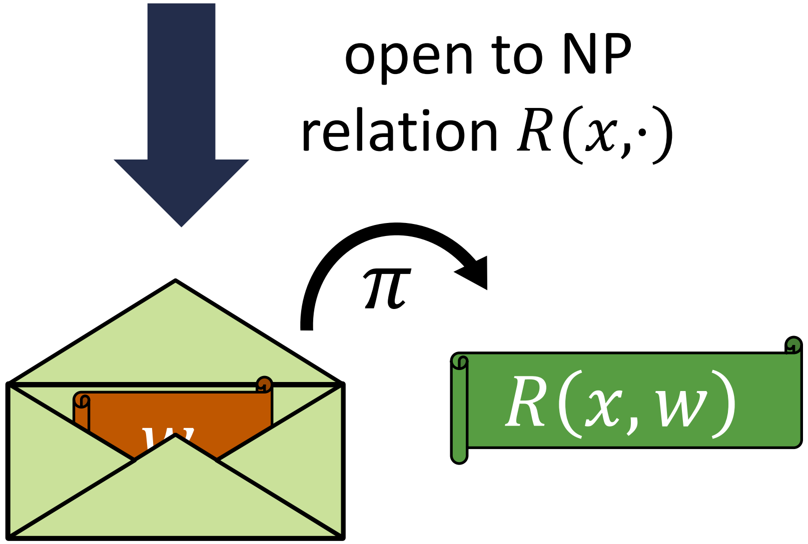
**Polynomial commitments:**

$[\alpha_0, \alpha_1, \dots, \alpha_d]$ $\xrightarrow{f_x(\alpha_0, \dots, \alpha_d) \coloneqq \alpha_0 + \alpha_1 x + \cdots + \alpha_d x^d}$ $y$

*commit to a polynomial, open to the evaluation at x*

# Connection to Succinct Arguments

**Goal:** prove that $x \in \mathcal{L}$ (where $\mathcal{L}$ is an NP language)


Proof consists of commitment $\sigma$ and opening $\pi$

Succinctness means that $|\sigma|, |\pi| = \text{poly}(\lambda, \log w)$

Soundness relies on **stronger** version of binding

**More generally:** can view functional commitments as providing <u>succinct</u> proofs on committed data

# Succinct Functional Commitments

*(not an exhaustive list!)*

| Scheme | Function Class | Assumption |
|---|---|---|
| [Mer87] | vector commitment | collision-resistant hash functions |
| [LY10, CF13, LM19, GRWZ20] | vector commitment | $q$-type pairing assumptions |
| [CF13, LM19] | vector commitment | groups of unknown order |
| [PPS21] | vector commitment | short integer solutions (SIS) |
| [KZG10, Lee20] | polynomial commitment | $q$-type pairing assumptions |
| [BFS19] | polynomial commitment | groups of unknown order |
| [LRY16] | Boolean circuits | collision-resistant hash functions + SNARKs |
| [LRY16] | linear functions | $q$-type pairing assumptions |
| [ACLMT22] | constant-degree polynomials | $k$-$R$-ISIS assumption (falsifiable) |
| **This work** | **vector commitment** | **short integer solutions (SIS)** |

*supports private openings, commitments to large values, linearly-homomorphic*

# Succinct Functional Commitments

*(not an exhaustive list!)*

| Scheme | Function Class | Assumption |
|---|---|---|
| [Mer87] | vector commitment | collision-resistant hash functions |
| [LY10, CF13, LM19, GRWZ20] | vector commitment | $q$-type pairing assumptions |
| [CF13, LM19] | vector commitment | groups of unknown order |
| [PPS21] | vector commitment | short integer solutions (SIS) |
| [KZG10, Lee20] | polynomial commitment | $q$-type pairing assumptions |
| [BFS19] | polynomial commitment | groups of unknown order |
| [LRY16] | Boolean circuits | collision-resistant hash functions + SNARKs |
| [LRY16] | linear functions | $q$-type pairing assumptions |
| [ACLMT22] | constant-degree polynomials | $k$-$R$-ISIS assumption (falsifiable) |
| **This work** | **vector commitment** | **short integer solutions (SIS)** |
| **This work** | **Boolean circuits** | $\mathbf{BASIS_{struct}}$ **assumption (falsifiable)** |

# Framework for Lattice Commitments

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

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

matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

target vectors $\boldsymbol{t}_1, \dots, \boldsymbol{t}_\ell \in \mathbb{Z}_q^{n}$

*auxiliary data:* cross-terms $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j) \in \mathbb{Z}_q^{m}$ where $i \neq j$


short (i.e., low-norm) vector satisfying $\boldsymbol{A}_i \boldsymbol{u}_{ij} = \boldsymbol{t}_j$

# Framework for Lattice Commitments

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

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

matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

target vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_\ell \in \mathbb{Z}_q^n$

*auxiliary data:* cross-terms $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j) \in \mathbb{Z}_q^m$ where $i \neq j$


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

$$\boldsymbol{c} = \sum_{i \in [\ell]} x_i \boldsymbol{t}_i$$

*linear combination of target vectors*

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

short $\boldsymbol{v}_i$ such that $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + y \cdot \boldsymbol{t}_i$

Honest opening:

*Correct as long as $\boldsymbol{x}$ is **short***

$$\boldsymbol{v}_i = \sum_{j \neq i} x_j \boldsymbol{u}_{ij}$$

$$\boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i = \sum_{j \neq i} x_j \boldsymbol{A}_i \boldsymbol{u}_{ij} + x_i \boldsymbol{t}_i = \sum_{j \in [\ell]} x_j \boldsymbol{t}_j = \boldsymbol{c}$$

# Framework for Lattice Commitments

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

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

matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

target vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_\ell \in \mathbb{Z}_q^n$

*auxiliary data:* cross-terms $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j) \in \mathbb{Z}_q^m$ where $i \neq j$


[PPS21]: $\boldsymbol{A}_i \leftarrow \mathbb{Z}_q^{n \times m}$ and $\boldsymbol{t}_i \leftarrow \mathbb{Z}_q^n$ are independent and uniform

*suffices for vector commitments (from SIS)*

[ACLMT21]: $\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$ and $\boldsymbol{t}_i = \boldsymbol{W}_i \boldsymbol{u}_i$ where $\boldsymbol{W}_i \leftarrow \mathbb{Z}_q^{n \times n}, \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{u}_i \leftarrow \mathbb{Z}_q^n$

(one candidate adaptation to the integer case)

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

# Our Approach

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

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

*for a short $\boldsymbol{v}_i$*

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

$$\begin{bmatrix} \boldsymbol{A}_1 & & & -\boldsymbol{I}_n \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{I}_n \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \boldsymbol{c} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

$\boldsymbol{I}_n$ denotes the identity matrix

# Our Approach

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

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

$$\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

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

*"powers of two matrix"*

$$\boldsymbol{G} = \begin{bmatrix} 1 & 2 & \cdots & 2^{\lfloor \log q \rfloor} & & & & \\ & & & & \ddots & & & \\ & & & & 1 & 2 & \cdots & 2^{\lfloor \log q \rfloor} \end{bmatrix}$$

# Our Approach

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

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

$$\begin{bmatrix} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

**Common reference string:**

- matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$
- target vectors $\boldsymbol{t}_1, \dots, \boldsymbol{t}_\ell \in \mathbb{Z}_q^n$
- *auxiliary data:* cross-terms $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j)$

# Our Approach

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

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

$$\underbrace{\begin{bmatrix} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{bmatrix}}_{\boldsymbol{B}_\ell} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

**Common reference string:**

matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

target vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_\ell \in \mathbb{Z}_q^{n}$

*auxiliary data:* ~~cross-terms $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j)$~~

trapdoor for $\boldsymbol{B}_\ell$

Trapdoor for $\boldsymbol{B}_\ell$ can be used to sample short solutions $\boldsymbol{x}$ to the linear system $\boldsymbol{B}_\ell \boldsymbol{x} = \boldsymbol{y}$ (for arbitrary $\boldsymbol{y}$)

# Our Approach

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

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

$$\underbrace{\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]}_{\boldsymbol{B}_\ell} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

Committing to an input $\boldsymbol{x}$:

Use trapdoor for $\boldsymbol{B}_\ell$ to **jointly** sample a solution $\boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell, \hat{\boldsymbol{c}}$

$\boldsymbol{c} = \boldsymbol{G}\hat{\boldsymbol{c}}$ is the commitment and $\boldsymbol{v}_1, \dots \boldsymbol{v}_\ell$ are the openings

Supports commitments to arbitrary (i.e., large) values over $\mathbb{Z}_q$

# Our Approach

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

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

$$\underbrace{\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]}_{\boldsymbol{B}_\ell} \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{t}_1 \\ \vdots \\ -x_\ell \boldsymbol{t}_\ell \end{bmatrix}$$

Committing to an input $\boldsymbol{x}$:

Use trapdoor for $\boldsymbol{B}_\ell$ to **jointly** sample a solution $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell, \hat{\boldsymbol{c}}$

$\boldsymbol{c} = \boldsymbol{G}\hat{\boldsymbol{c}}$ is the commitment and $\boldsymbol{v}_1, \ldots \boldsymbol{v}_\ell$ are the openings

Supports statistically private openings
(commitment + opening *hides* unopened positions)

# Proving Security

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

Suppose adversary can break binding

outputs $\boldsymbol{c}, (\boldsymbol{v}_i, x_i), (\boldsymbol{v}'_i, x'_i)$ such that

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \\ &= \boldsymbol{A}_i \boldsymbol{v}'_i + x'_i \boldsymbol{t}_i \end{aligned}$$

given matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell$

target vectors $\boldsymbol{t}_1, \dots, \boldsymbol{t}_\ell$

trapdoor for $\boldsymbol{B}_\ell$

set $\boldsymbol{A}_i \leftarrow \mathbb{Z}_q^{n \times m}$

set $\boldsymbol{t}_i = \boldsymbol{e}_1 = [1,0,\dots,0]^{\mathrm{T}}$

**Short integer solutions (SIS)**

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

$$\boldsymbol{A}_i(\boldsymbol{v}_i - \boldsymbol{v}'_i) = (x_i - x'_i)\boldsymbol{e}_1$$

$\boldsymbol{v}_i - \boldsymbol{v}'_i$ is a SIS solution for $\boldsymbol{A}_i$ without the first row

# Basis-Augmented SIS (BASIS) Assumption

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

Adversary that breaks binding can solve SIS with respect to $\boldsymbol{A}_i$

*(technically* $\boldsymbol{A}_i$ *without the first row – which is equivalent to SIS with dimension* $n-1$*)*

# Basis-Augmented SIS (BASIS) Assumption

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

**Verification invariant:** $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

*for a short* $\boldsymbol{v}_i$

Adversary that breaks binding can solve SIS with respect to $\boldsymbol{A}_i$

Basis-augmented SIS (BASIS) assumption:

*SIS is hard with respect to* $\boldsymbol{A}_i$

*given a trapdoor (a basis) for the matrix*

$$\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

Can simulate CRS from BASIS challenge:

matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$

trapdoor for $\boldsymbol{B}_\ell$

# Basis-Augmented SIS (BASIS) Assumption

*SIS is hard with respect to $\boldsymbol{A}_i$ given a trapdoor (a basis) for the matrix*

$$\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

When $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$ 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 $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

*auxiliary data: trapdoor for* $\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$

To commit to a vector $\boldsymbol{x} \in \mathbb{Z}_q^\ell$: sample solution $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell, \hat{\boldsymbol{c}})$

$$\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_\ell \\ \hat{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{e}_1 \\ \vdots \\ -x_\ell \boldsymbol{e}_\ell \end{bmatrix}$$

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

Openings are $\boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell$

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

Commitments and openings statistically **hide** unopened components

**Linearly homomorphic:**
$\boldsymbol{c} + \boldsymbol{c}'$ is a commitment to $\boldsymbol{x} + \boldsymbol{x}'$ with openings $\boldsymbol{v}_i + \boldsymbol{v}_i'$

# Functional Commitments for Circuits

**Setting:** commit to an input $\boldsymbol{x} \in \{0,1\}^\ell$, open to $f(\boldsymbol{x})$

*($f$ can be an arbitrary Boolean circuit)*

---

Will need some basic lattice machinery for homomorphic computation

[GSW13, BGGHNSVV14]

Let $\boldsymbol{A} = [\boldsymbol{A}_1 | \cdots | \boldsymbol{A}_\ell] \in \mathbb{Z}_q^{\ell n \times m}$ be an arbitrary matrix

Input-independent evaluation: $\boldsymbol{A}, f \mapsto \boldsymbol{A}_f \in \mathbb{Z}_q^{n \times m}$

Input-dependent evaluation: $\boldsymbol{A}, f, \boldsymbol{x} \mapsto \boldsymbol{H}_{\boldsymbol{A},f,\boldsymbol{x}} \in \mathbb{Z}_q^{\ell n \times m}$

Homomorphic evaluation:

$$[\boldsymbol{A}_1 - x_1 \boldsymbol{G} \mid \cdots \mid \boldsymbol{A}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\boldsymbol{A},f,\boldsymbol{x}} = \boldsymbol{A}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

*$\boldsymbol{A}_i - x_i \boldsymbol{G}$: "encoding" of $x_i$ with respect to $\boldsymbol{A}$*

# Functional Commitments using Structured $A_i$

Instead of using random $\boldsymbol{A}_i$, consider *structured* $\boldsymbol{A}_i$ (like in [ACLMT22])

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$$

$$\boldsymbol{W}_1, \dots, \boldsymbol{W}_\ell \leftarrow \mathbb{Z}_q^{n \times n} \quad \textit{(invertible)}$$

$$\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$$

Common reference string still consists of trapdoor for $\boldsymbol{B}_\ell$ (with the structured $\boldsymbol{A}_i$)

$$\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

# Functional Commitments using Structured $A_i$

Instead of using random $\boldsymbol{A}_i$, consider *structured* $\boldsymbol{A}_i$ (like in [ACLMT22])

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$$

$$\boldsymbol{W}_1, \dots, \boldsymbol{W}_\ell \leftarrow \mathbb{Z}_q^{n \times n} \quad \textit{(invertible)}$$

$$\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$$

$$\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

---

To commit to an input $\boldsymbol{x} \in \{0,1\}^\ell$:

Use trapdoor for $\boldsymbol{B}_\ell$ to **jointly** sample $\boldsymbol{V}_1, \dots, \boldsymbol{V}_\ell, \widehat{\boldsymbol{C}}$ that satisfy

$$\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_\ell \\ \widehat{\boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{W}_1 \boldsymbol{G} \\ \vdots \\ -x_\ell \boldsymbol{W}_\ell \boldsymbol{G} \end{bmatrix}$$

# Functional Commitments using Structured $A_i$

**Commitment relation:**

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

**Homomorphic evaluation:**

$\boldsymbol{B} = [\boldsymbol{B_1} \mid \cdots \mid \boldsymbol{B}_\ell]$

input-independent evaluation: $\boldsymbol{B}, f \mapsto \boldsymbol{B}_f$

input-dependent evaluation: $\boldsymbol{B}, f, \boldsymbol{x} \mapsto H_{\boldsymbol{B},f,\boldsymbol{x}}$

$$[\boldsymbol{B}_1 - x_1 \boldsymbol{G} \mid \cdots \mid \boldsymbol{B}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

for all $i \in [\ell]$

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

recall $\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$

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

recall $\boldsymbol{W}_i$ is invertible

$$\boldsymbol{A}\boldsymbol{V}_i - \boldsymbol{W}_i^{-1} \boldsymbol{G}\widehat{\boldsymbol{C}} = -x_i \boldsymbol{G}$$

rearranging

$$\boldsymbol{W}_i^{-1} \boldsymbol{G}\widehat{\boldsymbol{C}} - x_i \boldsymbol{G} = \boldsymbol{A}\boldsymbol{V}_i$$

# Functional Commitments using Structured $A_i$

**Commitment relation:**

$$\begin{bmatrix} A_1 & & & -G \\ & \ddots & & \vdots \\ & & A_\ell & -G \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ \vdots \\ V_\ell \\ \widehat{C} \end{bmatrix} = \begin{bmatrix} -x_1 W_1 G \\ \vdots \\ -x_\ell W_\ell G \end{bmatrix}$$

**Homomorphic evaluation:**

$B = [B_1 \mid \cdots \mid B_\ell]$

input-independent evaluation: $B, f \mapsto B_f$

input-dependent evaluation: $B, f, x \mapsto H_{B,f,x}$

$$[B_1 - x_1 G \mid \cdots \mid B_\ell - x_\ell G] \cdot H_{B,f,x} = B_f - f(x) \cdot G$$

function only of the commitment $C = G\widehat{C}$ $\quad \widetilde{C}_i = W_i^{-1} G \widehat{C}$

for all $i \in [\ell]$

$$A_i V_i - G\widehat{C} = -x_i W_i G$$

recall $A_i = W_i A$

$$W_i A V_i - G\widehat{C} = -x_i W_i G$$

recall $W_i$ is invertible

$$A V_i - W_i^{-1} G\widehat{C} = -x_i G$$

rearranging

$$W_i^{-1} G\widehat{C} - x_i G = A V_i$$

$$\widetilde{C}_i - x_i G = A V_i$$

# Functional Commitments using Structured $A_i$

**Commitment relation:**

$$\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_\ell \\ \widehat{\boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{W}_1 \boldsymbol{G} \\ \vdots \\ -x_\ell \boldsymbol{W}_\ell \boldsymbol{G} \end{bmatrix}$$

**Homomorphic evaluation:**

$\boldsymbol{B} = [\boldsymbol{B}_1 \mid \cdots \mid \boldsymbol{B}_\ell]$

input-independent evaluation: $\boldsymbol{B}, f \mapsto \boldsymbol{B}_f$

input-dependent evaluation: $\boldsymbol{B}, f, \boldsymbol{x} \mapsto H_{\boldsymbol{B},f,\boldsymbol{x}}$

$$[\boldsymbol{B}_1 - x_1 \boldsymbol{G} \mid \cdots \mid \boldsymbol{B}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

function only of the commitment $\boldsymbol{C} = \boldsymbol{G}\widehat{\boldsymbol{C}}$

$$\widetilde{\boldsymbol{C}}_i = \boldsymbol{W}_i^{-1} \boldsymbol{G} \widehat{\boldsymbol{C}}$$

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

$\widetilde{\boldsymbol{C}}_i$ is a GSW encryption of $x_i$ with randomness $\boldsymbol{V}_i$

Can also be viewed as a homomorphic commitment to $x_i$ with opening $\boldsymbol{V}_i$ [GVW15]

**[GVW15]:** independent $\boldsymbol{V}_i$ is sampled for each input bit, so commitments $\boldsymbol{C}_i$ are independent

- long commitment, security from SIS

**This work:** publish a trapdoor that allows deriving $\boldsymbol{C}_i$ (and associated $\boldsymbol{V}_i$) from a **single** commitment $\widehat{\boldsymbol{C}}$

- short commitment, stronger assumption

# Functional Commitments using Structured $A_i$

**Commitment relation:**

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

**Homomorphic evaluation:**

$\boldsymbol{B} = [\boldsymbol{B}_1 \mid \cdots \mid \boldsymbol{B}_\ell]$

input-independent evaluation: $\boldsymbol{B}, f \mapsto \boldsymbol{B}_f$

input-dependent evaluation: $\boldsymbol{B}, f, \boldsymbol{x} \mapsto H_{\boldsymbol{B},f,\boldsymbol{x}}$

$$[\boldsymbol{B}_1 - x_1\boldsymbol{G} \mid \cdots \mid \boldsymbol{B}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

$$\widetilde{\boldsymbol{C}}_i - x_i \boldsymbol{G} = \boldsymbol{A}\boldsymbol{V}_i \qquad \text{for all } i \in [\ell]$$

$$[\widetilde{\boldsymbol{C}}_1 - x_1\boldsymbol{G} \mid \cdots \mid \widetilde{\boldsymbol{C}}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\widetilde{\boldsymbol{C}},f,\boldsymbol{x}} = \boxed{\widetilde{\boldsymbol{C}}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}}$$

Can be computed by the verifier from commitment $\boldsymbol{C}$ and function $f$

function only of the commitment $\boldsymbol{C} = \boldsymbol{G}\widehat{\boldsymbol{C}}$

$$\boxed{\widetilde{\boldsymbol{C}}_i = \boldsymbol{W}_i^{-1} \boldsymbol{G} \widehat{\boldsymbol{C}}}$$

*Opening to function $f$ proceeds exactly as in [GVW15]*

# Functional Commitments using Structured $A_i$

**Commitment relation:**

$$\begin{bmatrix} A_1 & & & -G \\ & \ddots & & \vdots \\ & & A_\ell & -G \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ \vdots \\ V_\ell \\ \widehat{C} \end{bmatrix} = \begin{bmatrix} -x_1 W_1 G \\ \vdots \\ -x_\ell W_\ell G \end{bmatrix}$$

**Homomorphic evaluation:**

$B = [B_1 \mid \cdots \mid B_\ell]$

input-independent evaluation: $B, f \mapsto B_f$

input-dependent evaluation: $B, f, x \mapsto H_{B,f,x}$

$$[B_1 - x_1 G \mid \cdots \mid B_\ell - x_\ell G] \cdot H_{B,f,x} = B_f - f(x) \cdot G$$

function only of the commitment $C = G\widehat{C}$ $\quad \boxed{\widetilde{C}_i = W_i^{-1} G \widehat{C}}$

$\widetilde{C}_i - x_i G = A V_i$ for all $i \in [\ell]$

$$[\widetilde{C}_1 - x_1 G | \cdots | \widetilde{C}_\ell - x_\ell G] \cdot H_{\widetilde{C},f,x} = \boxed{\widetilde{C}_f - f(x) \cdot G}$$

Can be computed by the verifier from commitment $C$ and function $f$

$$[\widetilde{C}_1 - x_1 G | \cdots | \widetilde{C}_\ell - x_\ell G] = A[V_1 \mid \cdots \mid V_\ell] = A\widetilde{V}$$

Then

$$A\widetilde{V} H_{\widetilde{C},f,x} = \widetilde{C}_f - f(x) \cdot G$$

*Opening to function $f$ proceeds exactly as in [GVW15]*

# Functional Commitments using Structured $A_i$

**Commitment relation:**

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

**Homomorphic evaluation:**

$\boldsymbol{B} = [\boldsymbol{B}_1 \mid \cdots \mid \boldsymbol{B}_\ell]$

input-independent evaluation: $\boldsymbol{B}, f \mapsto \boldsymbol{B}_f$

input-dependent evaluation: $\boldsymbol{B}, f, \boldsymbol{x} \mapsto H_{\boldsymbol{B},f,\boldsymbol{x}}$

$$[\boldsymbol{B}_1 - x_1\boldsymbol{G} \mid \cdots \mid \boldsymbol{B}_\ell - x_\ell \boldsymbol{G}] \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

function only of the commitment $\boldsymbol{C} = \boldsymbol{G}\widehat{\boldsymbol{C}}$

$$\widetilde{\boldsymbol{C}}_i = \boldsymbol{W}_i^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}}$$

$\widetilde{\boldsymbol{C}}_i - x_i\boldsymbol{G} = \boldsymbol{A}\boldsymbol{V}_i$ for all $i \in [\ell]$

$$[\widetilde{\boldsymbol{C}}_1 - x_1\boldsymbol{G} \mid \cdots \mid \widetilde{\boldsymbol{C}}_\ell - x_\ell\boldsymbol{G}] \cdot \boldsymbol{H}_{\widetilde{\boldsymbol{C}},f,\boldsymbol{x}} = \widetilde{\boldsymbol{C}}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

Can be computed by the verifier from commitment $\boldsymbol{C}$ and function $f$

$$[\widetilde{\boldsymbol{C}}_1 - x_1\boldsymbol{G} \mid \cdots \mid \widetilde{\boldsymbol{C}}_\ell - x_\ell\boldsymbol{G}] = \boldsymbol{A}[\boldsymbol{V}_1 \mid \cdots \mid \boldsymbol{V}_\ell] = \boldsymbol{A}\widetilde{\boldsymbol{V}}$$

Then

$$\boldsymbol{A}\widetilde{\boldsymbol{V}}\boldsymbol{H}_{\widetilde{\boldsymbol{C}},f,\boldsymbol{x}} = \widetilde{\boldsymbol{C}}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

Define opening to be $\boldsymbol{V}_{f,f(x)} = \widetilde{\boldsymbol{V}}\boldsymbol{H}_{\widetilde{\boldsymbol{C}},f,\boldsymbol{x}}$

*Opening to function $f$ proceeds exactly as in [GVW15]*

# Functional Commitments using Structured $A_i$

**Commitment relation:**

$$\begin{bmatrix} A_1 & & & -G \\ & \ddots & & \vdots \\ & & A_\ell & -G \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ \vdots \\ V_\ell \\ \hat{C} \end{bmatrix} = \begin{bmatrix} -x_1 W_1 G \\ \vdots \\ -x_\ell W_\ell G \end{bmatrix}$$

**Homomorphic evaluation:**

$B = [B_1 \mid \cdots \mid B_\ell]$

input-independent evaluation: $B, f \mapsto B_f$

input-dependent evaluation: $B, f, x \mapsto H_{B,f,x}$

$[B_1 - x_1 G \mid \cdots \mid B_\ell - x_\ell G] \cdot H_{B,f,x} = B_f - f(x) \cdot G$

function only of the commitment $C = G\hat{C}$ $\quad \boxed{\tilde{C}_i = W_i^{-1} G \hat{C}}$

$\tilde{C}_i - x_i G = A V_i$ for all $i \in [\ell]$

$[\tilde{C}_1 - x_1 G \mid \cdots \mid \tilde{C}_\ell - x_\ell G] \cdot H_{\tilde{C},f,x} = \boxed{\tilde{C}_f - f(x) \cdot G}$

Can be computed by the verifier from commitment $C$ and function $f$

$[\tilde{C}_1 - x_1 G \mid \cdots \mid \tilde{C}_\ell - x_\ell G] = A[V_1 \mid \cdots \mid V_\ell]$

$= A\tilde{V}$

Then

$A\boxed{\tilde{V} H_{\tilde{C},f,x}} = \tilde{C}_f - f(x) \cdot G$

Define opening to be $V_{f,f(x)} = \tilde{V} H_{\tilde{C},f,x}$

Verification relation (for opening $C$ to $(f, z)$):

$$A V_{f,z} = \tilde{C}_f - z \cdot G$$

# Functional Commitments from Lattices

Security follows from BASIS assumption with a **structured** matrix:

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

$$\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

where $\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$ where $\boldsymbol{W}_i \leftarrow \mathbb{Z}_q^{n \times n}$ and $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$

Falsifiable assumption but does not appear to reduce to standard SIS

$\ell = 1$ case does follow from plain SIS

**Open problem:** Understanding security or attacks when $\ell > 1$

# Functional Commitments from Lattices

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

matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$ where $\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$

*auxiliary data: trapdoor for* $\boldsymbol{B}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$

To commit to a vector $\boldsymbol{x} \in \{0,1\}^\ell$: sample $(\boldsymbol{V}_1, \dots, \boldsymbol{V}_\ell, \widehat{\boldsymbol{C}})$

$$\left[\begin{array}{ccc|c} \boldsymbol{A}_1 & & & -\boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_\ell \\ \widehat{\boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} -x_1 \boldsymbol{W}_1 \boldsymbol{G} \\ \vdots \\ -x_\ell \boldsymbol{W}_\ell \boldsymbol{G} \end{bmatrix}$$

Commitment is $\boldsymbol{C} = \boldsymbol{G}\widehat{\boldsymbol{C}}$

Openings for function $f$ is $[\boldsymbol{V}_1 \mid \cdots \mid \boldsymbol{V}_\ell] \cdot \boldsymbol{H}_{\widetilde{\boldsymbol{C}}, f, \boldsymbol{x}}$

Scheme supports functions computable by Boolean circuits of (bounded) depth $d$

$$|\text{crs}| = \ell^2 \cdot \text{poly}(\lambda, d, \log \ell)$$

$$|\boldsymbol{C}| = \text{poly}(\lambda, d, \log \ell)$$

$$|\boldsymbol{V}_{f, f(\boldsymbol{x})}| = \text{poly}(\lambda, d, \log \ell)$$

Verification **time** scales with $|f|$ (i.e., size of circuit computing $f$)

# Fast Verification with Preprocessing

$$\widetilde{\boldsymbol{C}}_i = \boldsymbol{W}_i^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{W}_i^{-1}\boldsymbol{C}$$

To verify opening $\boldsymbol{V}$ to $(f, z)$, verifier computes the following:

- Homomorphic evaluation: $\widetilde{\boldsymbol{C}}_1, \dots, \widetilde{\boldsymbol{C}}_\ell, f \mapsto \widetilde{\boldsymbol{C}}_f$
- Verification relation: $\boldsymbol{AV} = \widetilde{\boldsymbol{C}}_f - z \cdot \boldsymbol{G}$

Computing $\widetilde{\boldsymbol{C}}_f$ corresponds to homomorphic computation on $\widetilde{\boldsymbol{C}}_1, \dots, \widetilde{\boldsymbol{C}}_\ell$

Suppose $f$ is a linear function:

$$f(x_1, \dots, x_\ell) = \sum_{i \in [\ell]} \alpha_i x_i$$

Then $\widetilde{\boldsymbol{C}}_f = \sum_{i\in[\ell]} \alpha_i \widetilde{\boldsymbol{C}}_i = \sum_{i\in[\ell]} \alpha_i \boldsymbol{W}_i^{-1}\boldsymbol{C} = \underbrace{\left(\sum_{i\in[\ell]} \alpha_i \boldsymbol{W}_i^{-1}\right)}_{\text{Can be precomputed}}\boldsymbol{C}$

Can be **precomputed**

For linear functions, if $f$ is known in advance, verification runs in time $\operatorname{poly}(\lambda, \log \ell)$

# Fast Verification with Preprocessing

$$\widetilde{\boldsymbol{C}}_i = \boldsymbol{W}_i^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{W}_i^{-1}\boldsymbol{C}$$

To verify opening $\boldsymbol{V}$ to $(f, z)$, verifier computes the following:

- Homomorphic evaluation: $\widetilde{\boldsymbol{C}}_1, \dots, \widetilde{\boldsymbol{C}}_\ell, f \mapsto \widetilde{\boldsymbol{C}}_f$
- Verification relation: $\boldsymbol{AV} = \widetilde{\boldsymbol{C}}_f - z \cdot \boldsymbol{G}$

Computing $\widetilde{\boldsymbol{C}}_f$ corresponds to homomorphic computation on $\widetilde{\boldsymbol{C}}_1, \dots, \widetilde{\boldsymbol{C}}_\ell$

Suppose $f$ is a linear function:

$$f(x_1, \dots, x_\ell) = \sum \alpha_i x_i$$

Can be **precomputed**

Captures polynomial commitments as a special case (polynomial evaluation can be described by a linear function)

For linear functions, if $f$ is known in advance, verification runs in time $\mathrm{poly}(\lambda, \log \ell)$

# Summary

New methodology for constructing lattice-based commitments:

1. Write down the main verification relation ($\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i$)
2. Publish a trapdoor for the linear system by the verification relation

Security analysis relies on basis-augmented SIS assumptions:

*SIS with respect to $A$ is hard given a trapdoor for a* ***related*** *matrix $B$*

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

“Structured” variant of BASIS assumption implies functional commitments

- Yields linear and polynomial commitments with fast preprocessed verification
- Structure also enables **aggregating** openings

*[see paper for details]*

# Open Questions

Analyzing BASIS family of assumptions (new reductions to SIS or attacks)

Functional commitments for **circuits** that supports fast preprocessed verification

- [ACLMT22]: fast preprocessed verification for constant-degree polynomials

Describe and analyze knowledge variants of the assumption or the constructions

Reducing CRS size: can we obtain functional commitments with *linear*-size CRS?

**Thank you!**

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