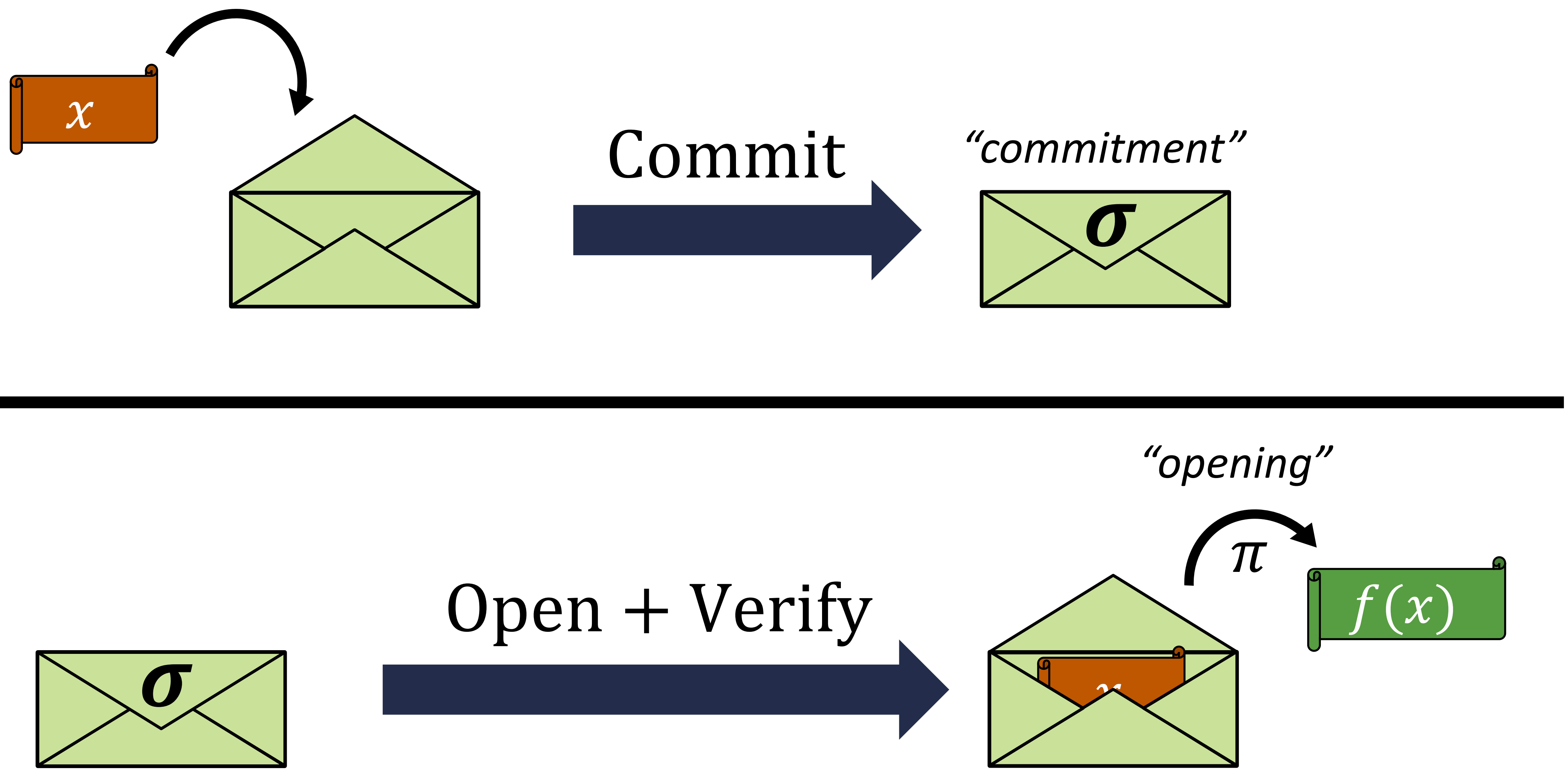
Succinct Vector, Polynomial, and Functional Commitments from Lattices

Hoeteck Wee and David Wu

May 2023

Functional Commitments



Functional Commitments

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

Takes a common reference string and commits to a message

Outputs commitment σ and commitment state st

Focus exclusively on non-interactive schemes

Functional Commitments

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

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

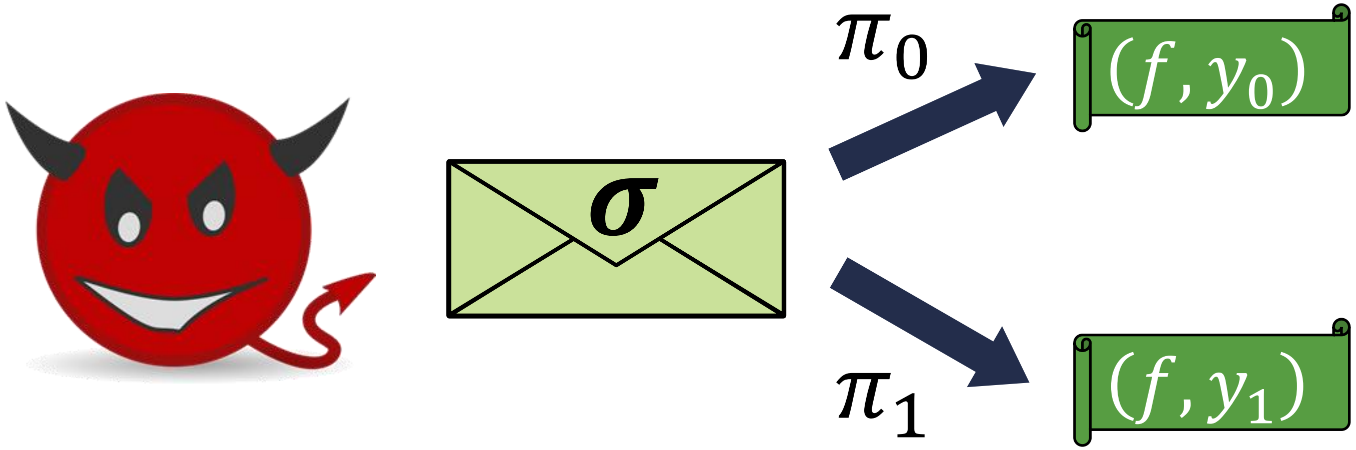
Takes the commitment state and a function f and outputs an opening π

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

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

Functional Commitments

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

Hiding: commitment σ and opening π 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: vector commitments, polynomial commitments

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

Functional Commitment Constructions

(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, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
[BFS19, BHRRS21, BF23]	polynomial commitment	groups of unknown order
[LRY16]	Boolean circuits	collision-resistant hash functions + SNARKs

non-falsifiable, non-black box

Functional Commitment Constructions

(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, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
[BFS19, BHRRS21, BF23]	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

Functional Commitment Constructions

(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, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
[BFS19, BHRRS21, BF23]	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)**

$\mathrm{BASIS}_{\mathrm{struct}}$ *assumption less structured than* [ACLMT22] (no short preimages of powers)

Functional Commitment Constructions

(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, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
[BFS19, BHRRS21, BF23]	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)**

Concurrent works [BCFL22, dCP23]: lattice-based constructions of functional commitments for Boolean circuits

Functional Commitment Constructions

(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, BBF19]	vector commitment	groups of unknown order
[PPS21]	vector commitment	short integer solutions (SIS)
[KZG10, Lee20]	polynomial commitment	q-type pairing assumptions
		groups of unknown order
		collision-resistant hash functions + SNARKs
		q-type pairing assumptions
		k-R-ISIS assumption (falsifiable)
		short integer solutions (SIS)
		$\mathbf{BASIS_{struct}}$ **assumption (falsifiable)**

[BCFL22]: short openings and supports *fast* verification with preprocessing; based on (falsifiable) twin-k-M-ISIS assumption

[dCP23]: transparent setup from SIS, long openings, selectively-secure (without complexity leveraging)

Concurrent works [BCFL22, dCP23]: lattice-based constructions of functional commitments for Boolean circuits

Framework for Lattice Commitments

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

Common reference string (for inputs of length ℓ):

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 ℓ):

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 $i \neq j$

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

$$\boldsymbol{c} = \sum_{j \in [\ell]} x_j \boldsymbol{t}_j$$

linear combination of target vectors

Opening to value y at index i:

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

Honest opening:

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

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

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

Framework for Lattice Commitments

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

Common reference string (for inputs of length ℓ):

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 $i \neq j$

[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)

<u>generalizes</u> 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: $\boldsymbol{c} = \boldsymbol{A}_i \boldsymbol{v}_i + x_i \boldsymbol{t}_i \qquad \forall i \in [\ell]$

for a short $\boldsymbol{v}_i$

Our approach: rewrite ℓ relations as a single linear system

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

$\boldsymbol{I}_n$ 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 ℓ relations as a single linear system

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

"powers of two matrix"

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

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

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 ℓ relations as a single linear system

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

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 ℓ relations as a single linear system

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

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 $\boldsymbol{u}_{ij} \leftarrow \boldsymbol{A}_i^{-1}(\boldsymbol{t}_j)$~~

(random) 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 ℓ 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 ℓ 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 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 $\boldsymbol{v}_i$

Our scheme

Suppose adversary can break binding

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

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

$x_i \neq x_i' \in \mathbb{Z}_q$
can be large!

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

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

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

Goal: reduce to **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 $\boldsymbol{v}_i$

Our scheme

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 $\boldsymbol{v}_i$

Our scheme

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 random trapdoor (a random 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)

$$\boldsymbol{B}_\ell = \left[\begin{array}{cccc|c} \boldsymbol{A}_1 & & & & -\boldsymbol{G} \\ & \boldsymbol{A}_2 & & & -\boldsymbol{G} \\ & & \ddots & & \vdots \\ & & & \boldsymbol{A}_\ell & -\boldsymbol{G} \end{array}\right]$$

Sketch for $\boldsymbol{i = 1}$:

- Sample $\boldsymbol{A}_2, \dots, \boldsymbol{A}_\ell$ with trapdoors
- Use trapdoors for $\boldsymbol{A}_2, \dots, \boldsymbol{A}_\ell$ and $\boldsymbol{G}$ to trapdoor for $\boldsymbol{B}_\ell$

Vector Commitments from SIS

Common reference string (for inputs of length ℓ):

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)

Starting point: lattice-based homomorphic commitments [GSW13, BGGHNSVV14, GVW15]

Let $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ be an arbitrary matrix

$\boldsymbol{C}_f$ is a function of $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell, f$
$\boldsymbol{V}_f$ is a function of $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_\ell, f, \boldsymbol{x}$

$$\boldsymbol{C}_1 = \boldsymbol{A}\boldsymbol{V}_1 + x_1\boldsymbol{G}$$
$$\vdots$$
$$\boldsymbol{C}_\ell = \boldsymbol{A}\boldsymbol{V}_\ell + x_\ell\boldsymbol{G}$$

homomorphic evaluation

$$\boldsymbol{C}_f = \boldsymbol{A}\boldsymbol{V}_f + f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

[GVW15]: $\boldsymbol{C}_i$ is a commitment to x_i with (short) opening $\boldsymbol{V}_i$

$\boldsymbol{C}_f$ is a commitment to $f(\boldsymbol{x})$ with (short) opening $\boldsymbol{V}_f$

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)

Starting point: lattice-based homomorphic commitments [GSW13, BGGHNSVV14, GVW15]

Let $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ be an arbitrary matrix

$$\boldsymbol{C}_1 = \boldsymbol{A}\boldsymbol{V}_1 + x_1 \boldsymbol{G}$$
$$\vdots$$
$$\boldsymbol{C}_\ell = \boldsymbol{A}\boldsymbol{V}_\ell + x_\ell \boldsymbol{G}$$

[GVW15]: $\boldsymbol{C}_i$ is a commitment to x_i with (short) opening $\boldsymbol{V}_i$

[GVW15]: long commitments (linear in $|\boldsymbol{x}|$)
$\boldsymbol{C}_1, \dots, \boldsymbol{C}_\ell$ are independent

Our approach: compress $\boldsymbol{C}_1, \dots, \boldsymbol{C}_\ell$ into a single $\widehat{\boldsymbol{C}}$

We will define $\boldsymbol{C}_i = \boldsymbol{W}_i^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}}$ where $\boldsymbol{W}_i \in \mathbb{Z}_q^{n \times n}$ is part of the common reference string

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)

$$\begin{matrix} \boldsymbol{C}_1 = \boldsymbol{A}\boldsymbol{V}_1 + x_1\boldsymbol{G} \\ \vdots \\ \boldsymbol{C}_\ell = \boldsymbol{A}\boldsymbol{V}_\ell + x_\ell\boldsymbol{G} \end{matrix} \Rightarrow \begin{matrix} \boldsymbol{W}_1^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{A}\boldsymbol{V}_1 + x_1\boldsymbol{G} \\ \vdots \\ \boldsymbol{W}_\ell^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{A}\boldsymbol{V}_\ell + x_\ell\boldsymbol{G} \end{matrix} \Rightarrow \begin{matrix} \boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{W}_1\boldsymbol{A}\boldsymbol{V}_1 + x_1\boldsymbol{W}_1\boldsymbol{G} \\ \vdots \\ \boldsymbol{G}\widehat{\boldsymbol{C}} = \boldsymbol{W}_\ell\boldsymbol{A}\boldsymbol{V}_\ell + x_\ell\boldsymbol{W}_\ell\boldsymbol{G} \end{matrix}$$

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

Target is now a *matrix*

Our approach: commitment is $\widehat{\boldsymbol{C}}$ and set $\boldsymbol{C}_i = \boldsymbol{W}_i^{-1}\boldsymbol{G}\widehat{\boldsymbol{C}}$

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)

As in the case of vector commitments, we can publish a trapdoor for $\boldsymbol{B}_\ell$ in the CRS (along with the matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_\ell$)

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

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

Our approach: commitment is $\widehat{\boldsymbol{C}}$ and set $\boldsymbol{C}_i = \boldsymbol{W}_i^{-1} \boldsymbol{G} \widehat{\boldsymbol{C}}$

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)

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

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

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

To compute an opening with respect to f:

$\boldsymbol{V}_1, \dots, \boldsymbol{V}_\ell, f \mapsto \boldsymbol{V}_f$ as in [GVW15]

To check an opening $\boldsymbol{V}_f$ to z with respect to f:

derive commitments $\boldsymbol{C}_i \leftarrow \boldsymbol{W}_i^{-1} \boldsymbol{G} \widehat{\boldsymbol{C}}$

compute $\boldsymbol{C}_1, \dots, \boldsymbol{C}_\ell, f \mapsto \boldsymbol{C}_f$ as in [GVW15]

check $\boldsymbol{C}_f = \boldsymbol{A}\boldsymbol{V}_f + z \cdot \boldsymbol{G}$

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 ℓ):

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

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

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

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

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: $\boldsymbol{A}\boldsymbol{V} = \widetilde{\boldsymbol{C}}_f - z \cdot \boldsymbol{G}$

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 we can write $\widetilde{\boldsymbol{C}}_f = \overbrace{\left(\sum_{i \in [\ell]} \alpha_i \boldsymbol{W}_i^{-1}\right)}^{\boldsymbol{W}_f} \boldsymbol{C}$

$\boldsymbol{W}_f$ is a fixed matrix that depends only on f and can be computed in the *offline phase*

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

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: $\boldsymbol{A}\boldsymbol{V} = \widetilde{\boldsymbol{C}}_f - z \cdot \boldsymbol{G}$

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:

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

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 $\mathrm{poly}(\lambda, \log \ell)$

Comparison to Concurrent Work

Consider a bivariate function $F(x, y)$

commit to input x

open at y to the value $F(x, y)$

F is computable by a circuit of depth d and width w

Scheme	$\lvert\mathrm{crs}\rvert$	$\lvert\mathrm{com}\rvert$	$\lvert\mathrm{open}\rvert$	Assumption	Fast Verification	Transparent	Adaptive Security
[dCP23]	$\lvert y\rvert$	1	$\lvert y\rvert$	SIS	✗	✓	✗
[BCFL22]	w^5	1	1	twin-k-M-ISIS	✓	✗	✓
This work	$\lvert x\rvert^2$	1	1	$\mathrm{BASIS}_{\mathrm{struct}}$	✗	✗	✓

All comparisons ignoring $\mathrm{poly}(\lambda, d)$ *factors*

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 $\boldsymbol{A}$ is hard given a trapdoor for a **related** matrix $\boldsymbol{B}$*

"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)

Analyze knowledge variants of the assumption

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

Solved in [CLM23] for the case of constant-degree polynomials!

Direct construction of lattice-based *subvector* commitments

Construction in our paper does not satisfy consistency

Thank you!

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