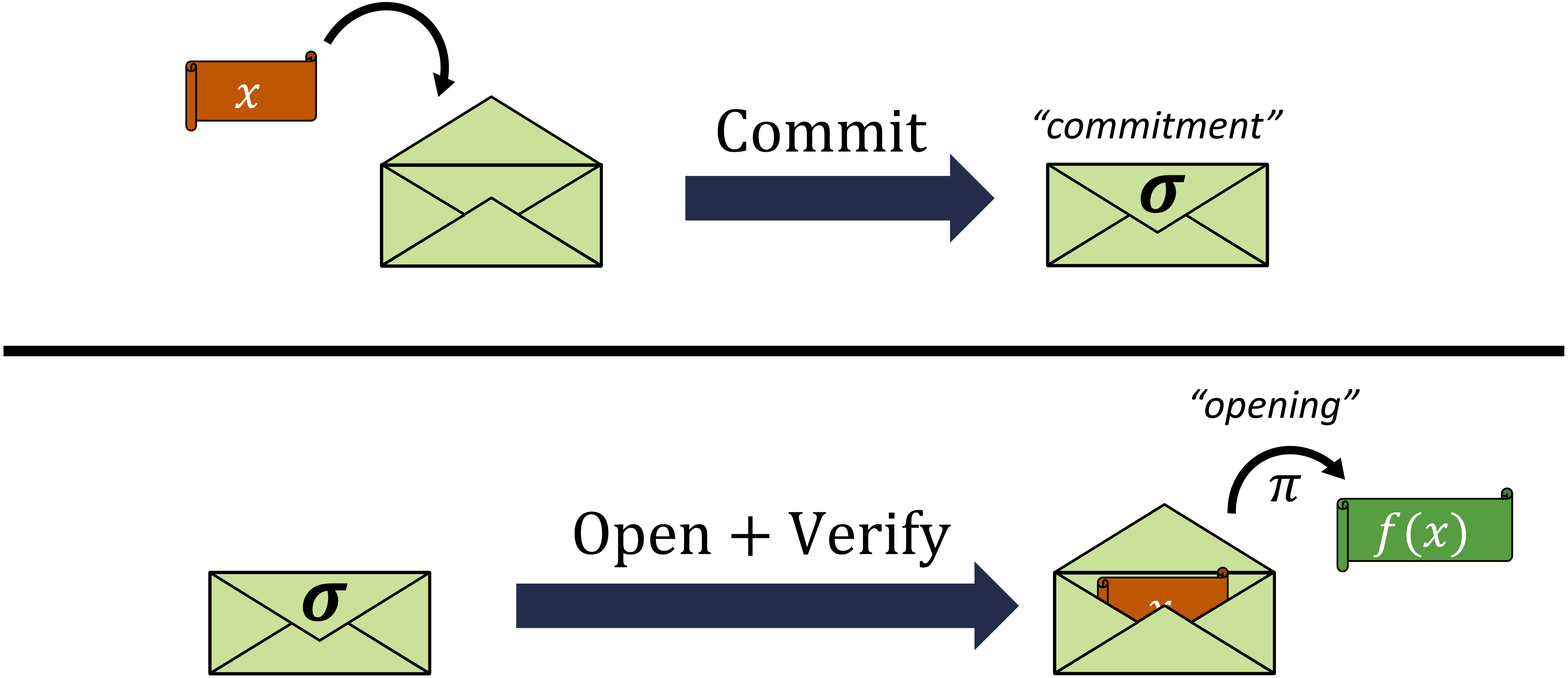
Succinct Vector, Polynomial, and Functional Commitments from Lattices

Hoeteck Wee and David Wu

April 2023

Functional Commitments



Functional Commitments

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

Takes a common reference string and commits to a message

Outputs commitment σ and commitment state st

Functional Commitments

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

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

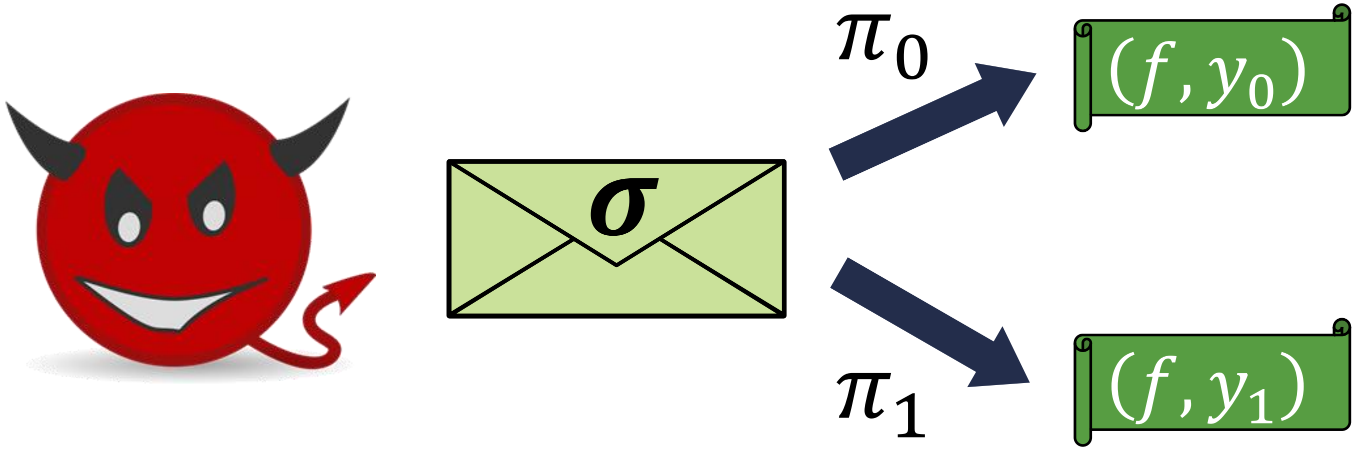
Takes the commitment state and a function f and outputs an opening π

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

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

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)**

$\mathrm{BASIS_{struct}}$ *assumption less structured than* [ACLMT22]

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)
		$\mathsf{BASIS}_{\mathsf{struct}}$ 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 $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell \in \mathbb{Z}_q^{n \times m}$

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

auxiliary data: short preimages $\boldsymbol{u}_{ij}$ where $\boldsymbol{A}_i \boldsymbol{u}_{ij} = \boldsymbol{t}_j$ for $i \neq j$

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: short preimages $\boldsymbol{u}_{ij}$ where $\boldsymbol{A}_i \boldsymbol{u}_{ij} = \boldsymbol{t}_j$ for $i \neq j$

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: short preimages $\boldsymbol{u}_{ij}$ where $\boldsymbol{A}_i \boldsymbol{u}_{ij} = \boldsymbol{t}_j$ for $i \neq j$

[PPS21]: $\boldsymbol{A}_i$ and $\boldsymbol{t}_i$ are **random**

suffices for vector commitments (from SIS)

[ACLMT22]: $\boldsymbol{A}_i$ and $\boldsymbol{t}_i$ are **structured**

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

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

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

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

Trapdoor for $\boldsymbol{B}_\ell$ can be used to sample <u>short</u> 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)

Computational Binding

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 $n - 1$)

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

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 trapdoor (a basis) for the matrix

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

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 $\boldsymbol{i} = \mathbf{1}$:

- 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 $\boldsymbol{B}_\ell$

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

Implication: vector commitment that supports committing to *large* values and private openings based on SIS

Previously: could only commit to *small* values and without hiding

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

$$\begin{gathered} \boldsymbol{C}_1 = \boldsymbol{A}\boldsymbol{V}_1 + x_1 \boldsymbol{G} \\ \vdots \\ \boldsymbol{C}_\ell = \boldsymbol{A}\boldsymbol{V}_\ell + x_\ell \boldsymbol{G} \end{gathered} \xrightarrow{\text{homomorphic evaluation}} \boldsymbol{C}_f = \boldsymbol{A}\boldsymbol{V}_f + f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

[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 <u>independent</u>

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

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)

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

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

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 $\boldsymbol{W}_1, \dots, \boldsymbol{W}_\ell$)

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

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

Homomorphic computation + opening verification now proceed as in [GVW15]

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 $\ell > 1$

Extensions

Our functional commitment:

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

Fast verification: for linear functions (captures polynomial commitments), can preprocess and support fast verification

Aggregation: can aggregate openings to $f_1, \dots, f_T$ into single opening

[see paper for details]

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

Open Questions

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

Describe and analyze knowledge variants of the assumption or the constructions

Reducing CRS size: functional commitments with *linear*-size CRS?

Constructing lattice-based *subvector* commitments

Thank you!

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