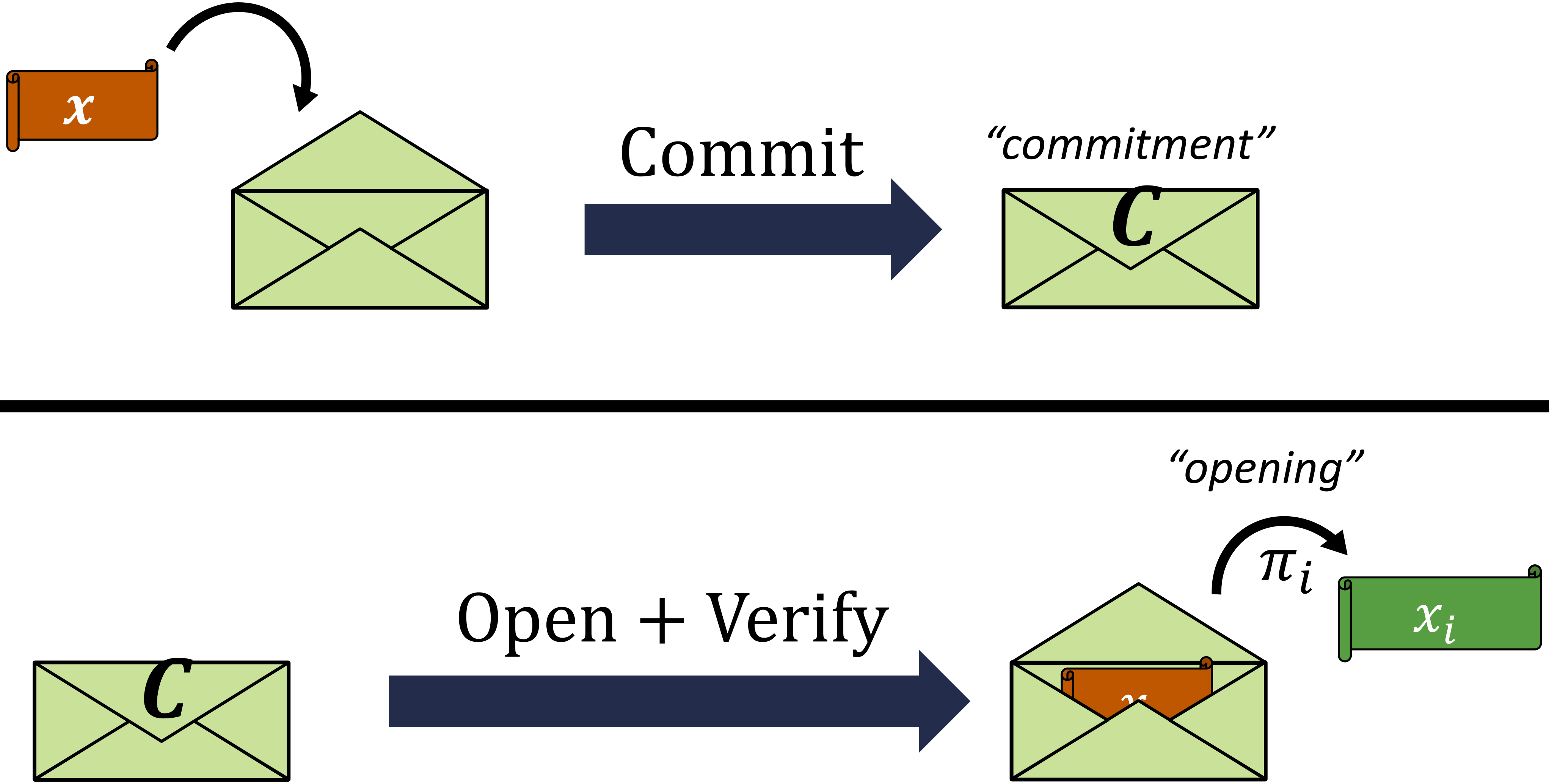
Updatable, Aggregatable, Succinct Mercurial Vector Commitment from Lattice

Hongxiao Wang, Siu-Ming Yiu, Yanmin Zhao, Zoe L. Jiang

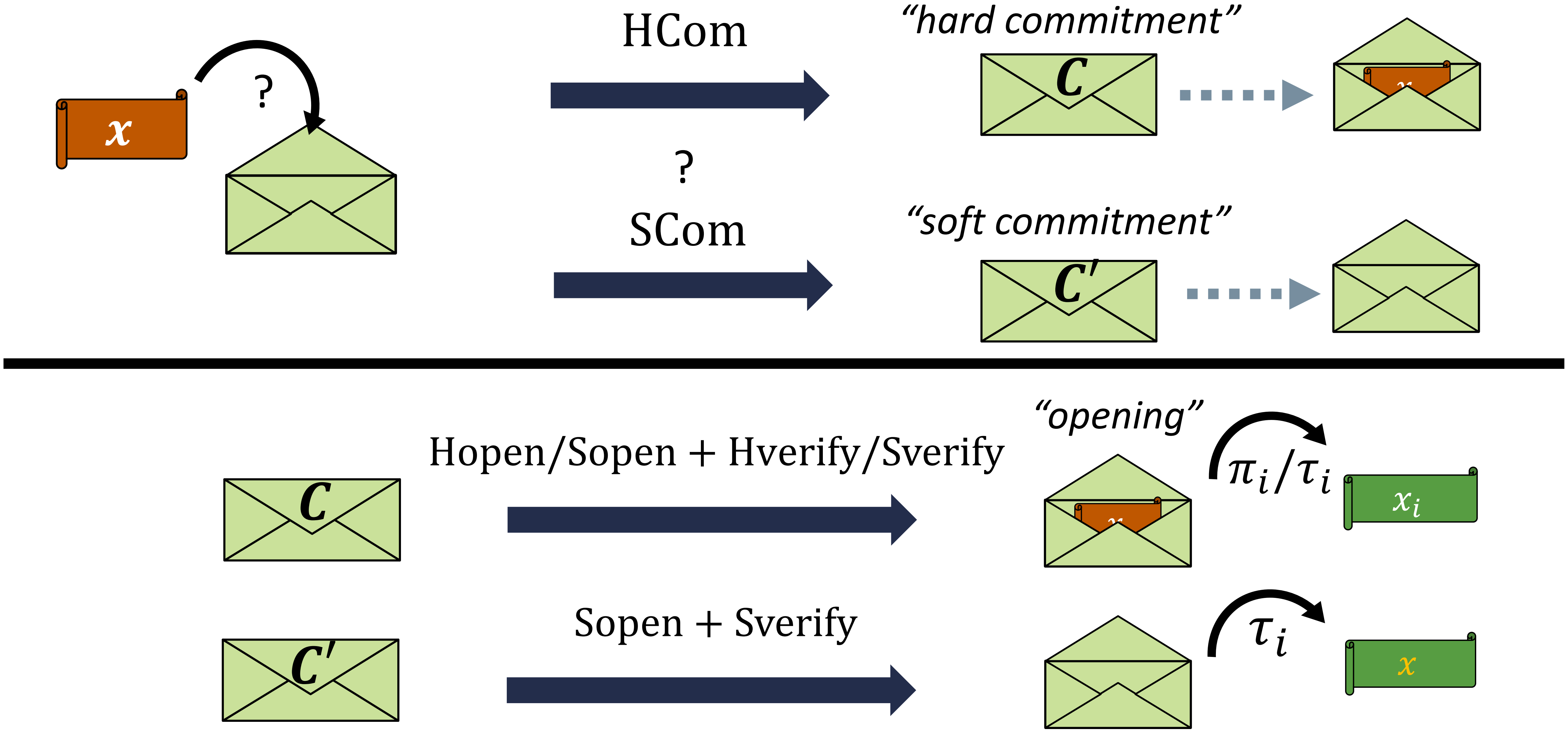
April 2024

Vector Commitment



Wee, Hoeteck, and David J. Wu. "Lattice-based functional commitments: Fast verification and cryptanalysis." *ASIACRYPT* 2023. PowerPoint slides: https://www.cs.utexas.edu/~dwu4/lattice-fc-fast.html

Mercurial Vector Commitment



Mercurial Vector Commitment

$$\mathrm{HCom}(\mathrm{crs}, \boldsymbol{x}) \rightarrow (C, aux)$$

Takes a common reference string and commits to a vector $\boldsymbol{x}$

Outputs a hard commitment C and auxiliary information aux

Mercurial Vector Commitment

$\text{HCom}(\text{crs}, x) \rightarrow (C, aux)$

$\text{HOpen}(aux, i) \rightarrow \pi_i$

Takes the auxiliary information and an index i and outputs a hard opening π_i

$\text{HVerify}(\text{pp}, C, (i, x_i), \pi_i) \rightarrow 0/1$

Checks whether π_i is valid opening of C to value x_i at index i

Mercurial Vector Commitment

$\mathrm{HCom}(\mathrm{crs}, x) \to (C, aux)$

$\mathrm{SOpen}(aux, \mathbb{H}, x_i, i) \to \tau_i$

Takes the auxiliary information, a flag $\mathbb{H}$, the value x_i at an index i and outputs a soft opening τ_i

$\mathrm{SVerify}(\mathrm{crs}, C, (i, x_i), \tau_i) \to 0/1$

Checks whether τ_i is valid opening of C to value x_i at index i

Mercurial Vector Commitment

$\mathrm{HCom}(\mathrm{crs}, x) \rightarrow (C, aux)$

$\mathrm{HOpen}(aux, i) \rightarrow \pi_i$

$\mathrm{SOpen}(aux, \mathbb{H}, x_i, i) \rightarrow \tau_i$

$\mathrm{HVerify}(\mathrm{pp}, C, (i, x_i), \pi_i) \rightarrow 0/1$

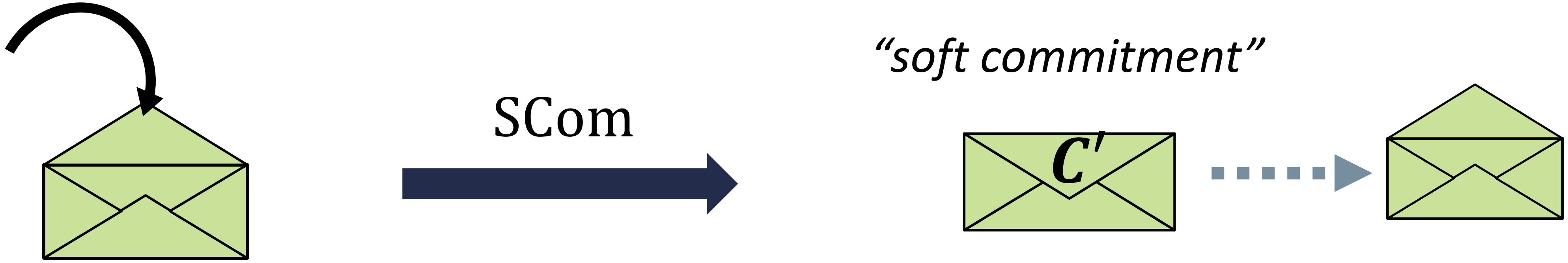
$\mathrm{SVerify}(\mathrm{crs}, C, (i, x_i), \tau_i) \rightarrow 0/1$

$\tau_i \subset \pi_i$

For all known constructions, soft opening τ_i is a proper subset of hard opening π_i to the same message, so as SVerify and HVerify.

Such MVC are called proper MVC.

Mercurial Vector Commitment



$\mathrm{SCom}(\mathrm{crs}) \rightarrow (C, aux)$

Takes a common reference string

Outputs soft commitment C and auxiliary information aux

Mercurial Vector Commitment

$\mathrm{SCom}(\mathrm{crs}) \rightarrow (C, aux)$

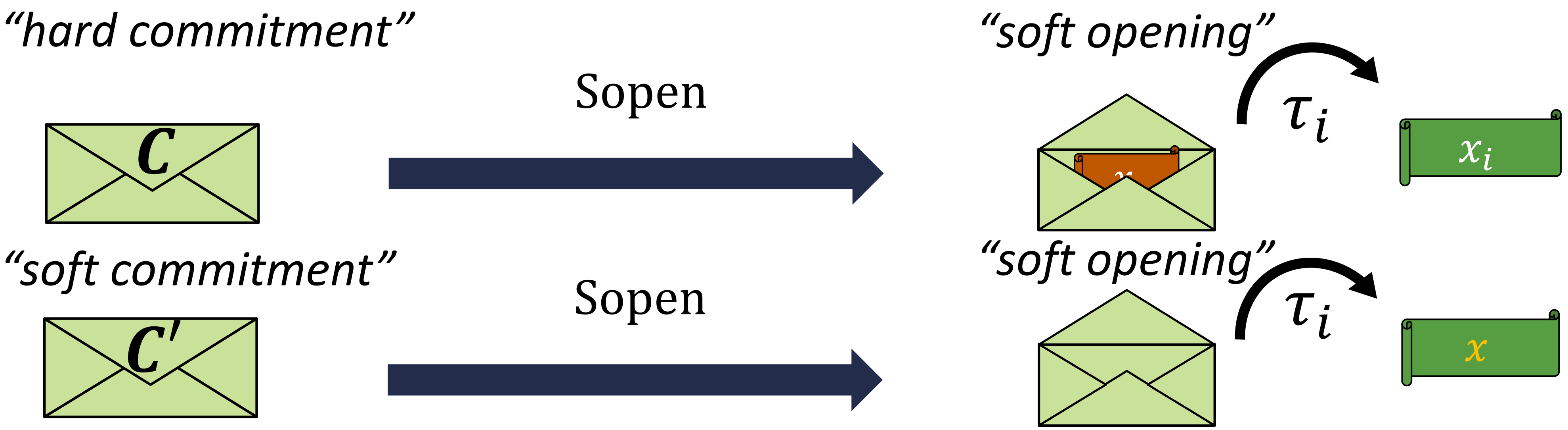
$\mathrm{SOpen}(aux, \mathbb{S}, x, i) \rightarrow \tau_i$

Takes the auxiliary information, a flag $\mathbb{S}$, a value x, and an index i and outputs a soft opening τ_i

$\mathrm{SVerify}(\mathrm{crs}, C, (i, x), \tau_i) \rightarrow 0/1$

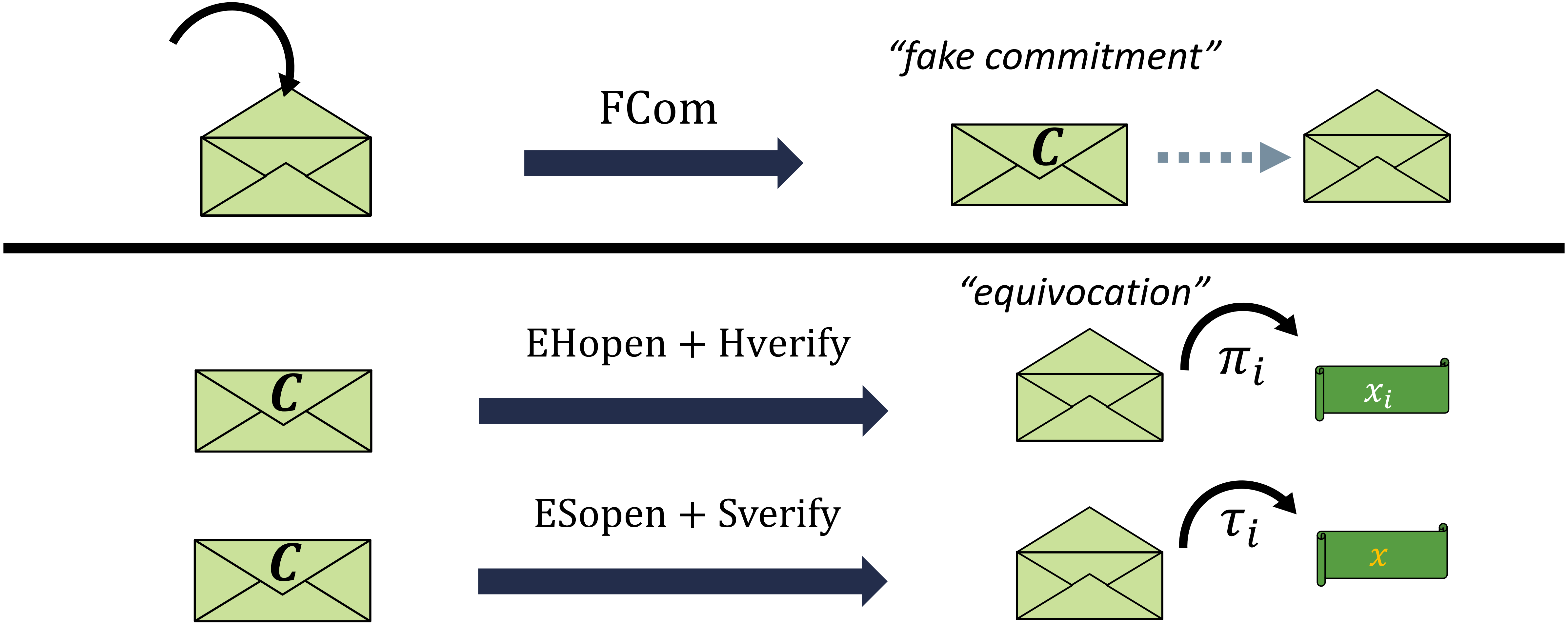
Checks whether τ_i is valid opening of C to value x at index i

Mercurial Vector Commitment



Mercurial Hiding: efficient adversary cannot distinguish between hard commitment C and soft commitment C' with their soft openings

Mercurial Vector Commitment



Simulating algorithms

Mercurial Vector Commitment

$\mathrm{FCom}(\mathrm{crs}) \rightarrow (C, aux)$

$\mathrm{EHOpen}(aux, tk, x, i) \rightarrow \pi$

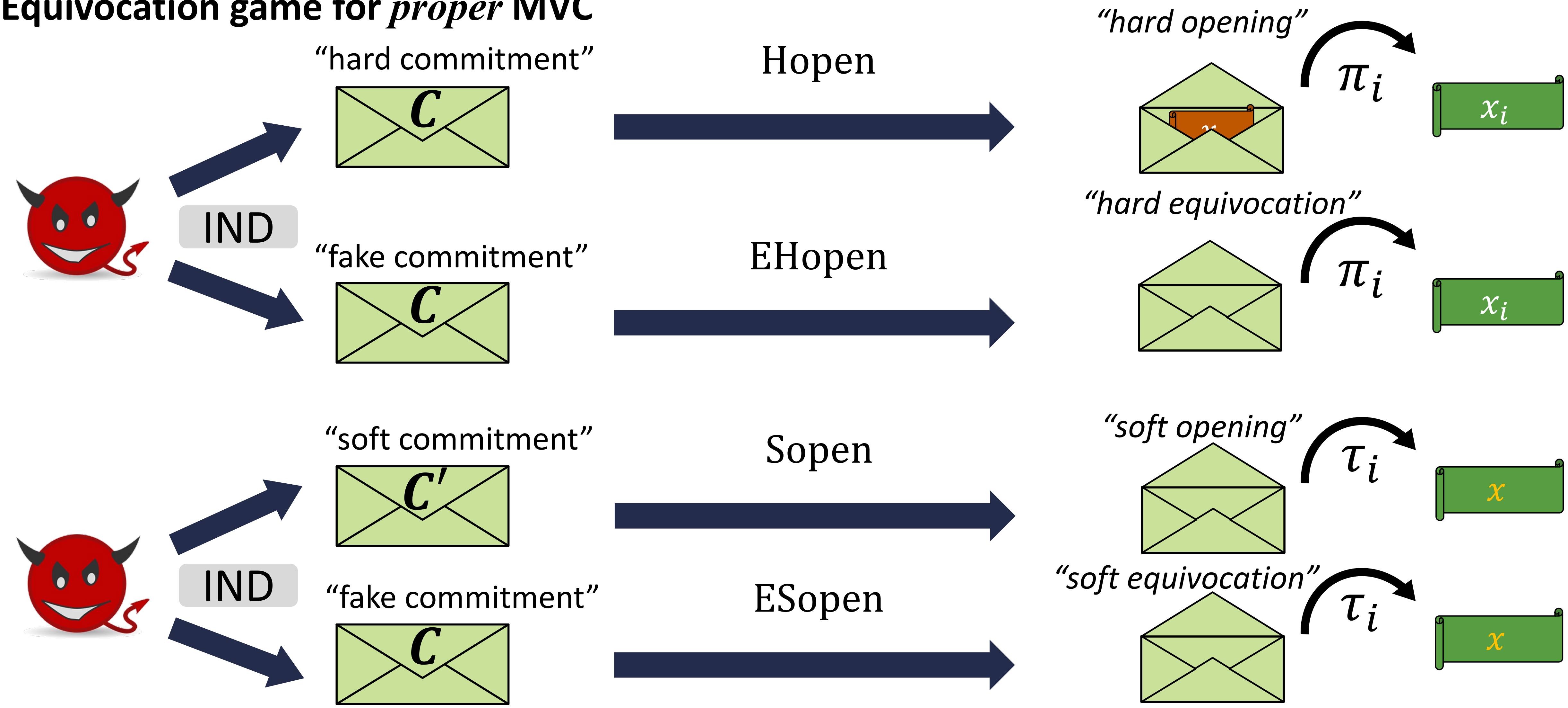
Takes the auxiliary information, the trapdoor key tk, a value x, and an index i and outputs a hard equivocation π

$\mathrm{ESOpen}(aux, tk, x, i) \rightarrow \tau$

Takes the auxiliary information, the trapdoor key tk, a value x, and an index i and outputs a soft equivocation τ

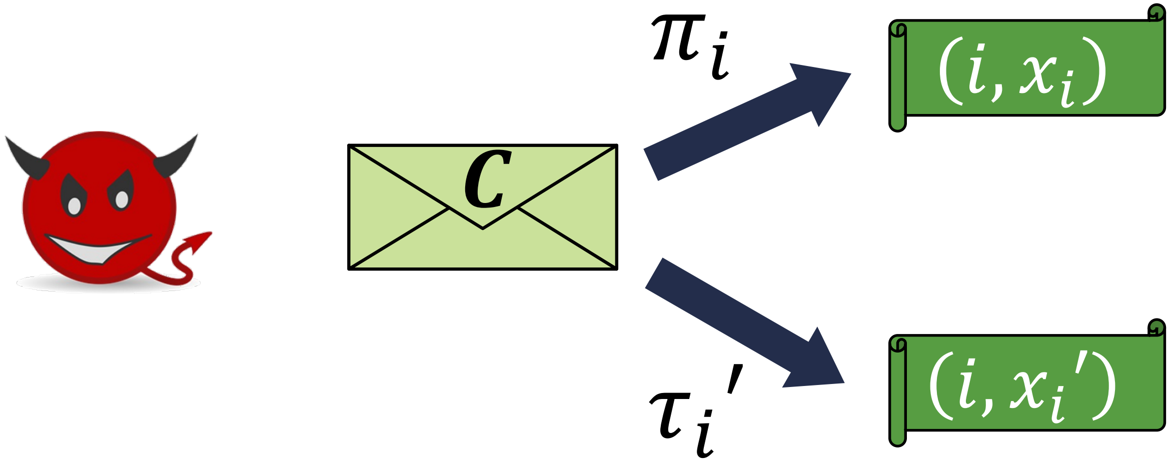
Mercurial Vector Commitment

Equivocation game for *proper* MVC



Mercurial Vector Commitment

Mercurial Binding: efficient adversary cannot open a **hard commitment** C to two different values at the **same index** i **successfully**



$$\text{HVerify}(\text{crs}, C, i, x_i, \pi_i) = 1$$

$$\text{SVerify}(\text{crs}, C, i, x_i, \tau_i) = 1$$

Mercurial Vector Commitment

Succinctness: all commitments and openings should be short

- **Short commitment:** $|C| = \text{poly}(\lambda, \log \ell)$
- **Short opening:** $|\pi_i| = \text{poly}(\lambda, \log \ell)$

Mercurial Vector Commitment

Scheme	AS	UD	AG	$\lvert \mathsf{crs} \rvert$	$\lvert C \rvert$	$\lvert aux \rvert$	$\lvert \pi \rvert$
[8]	RSA	✓	✗	$\Theta(\lambda\ell)$	$\Theta(\lambda)$	$\Theta(\lambda\ell)$	$\Theta(\lambda)$
[17]	l-DHE	✗	✓	$\Theta(\lambda\ell)$	$\Theta(\lambda)$	$\Theta(\lambda\ell)$	$\Theta(\lambda)$
[18]+[28]*	SIS	✗	✗	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta(\lambda^2\ell \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$
This work	SIS	✓	✗	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta(\lambda^2\ell \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$
This work	BASIS†	✓	✓	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta((\lambda\ell + \lambda^2) \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$

- ℓ is the input length
- $\mathcal{H} = \log^2\lambda + \log^2\ell$
- **UD:** scheme supports update both hard and soft commitment
- **FV:** scheme supports aggregate both hard and soft opening

*A lattice-based MVC can be trivially built by lattice-based components (e.g. [18] and [28]) in the generic framework [8].

†A new falsifiable family of basis-augmented SIS assumption (BASIS) proposed by Wee and Wu (EUROCRYPT '23)

Mercurial Vector Commitment

Scheme	AS	UD	AG	$\|\mathsf{crs}\|$	$\|C\|$	$\|aux\|$	$\|\pi\|$
[8]	RSA	✓	✗	$\Theta(\lambda\ell)$	$\Theta(\lambda)$	$\Theta(\lambda\ell)$	$\Theta(\lambda)$
[17]	l-DHE	✗	✓	$\Theta(\lambda\ell)$	$\Theta(\lambda)$	$\Theta(\lambda\ell)$	$\Theta(\lambda)$
[18]+[28]	SIS	✗	✗	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta(\lambda^2\ell \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$
This work	SIS	✓	✗	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta(\lambda^2\ell \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$
This work	BASIS	✓	✓	$\ell^2\text{poly}(\lambda, \log \ell)$	$\Theta(\lambda^2 \cdot \mathcal{H})$	$\Theta((\lambda\ell + \lambda^2) \cdot \mathcal{H})$	$\Theta(\lambda^2 \cdot \mathcal{H})$

The generic framework [8] of MVC including a **standard MC** and a **standard VC**

- First, generate ℓ MC C_i to each entry x_i with aux_i
- Second, generate a VC σ to the vector $(C_1, \dots, C_\ell)$ with aux_σ
- Third, publish σ as the MVC and store $\boldsymbol{aux = (aux_1, \dots, aux_\ell, aux\_\sigma)}$
- To generate an opening, it first opens VC σ at index i, then opens MC C_i
- To verify, it first verifies VC and then MC

Due to the generic framework, it **cannot support update and aggregate**

This Work

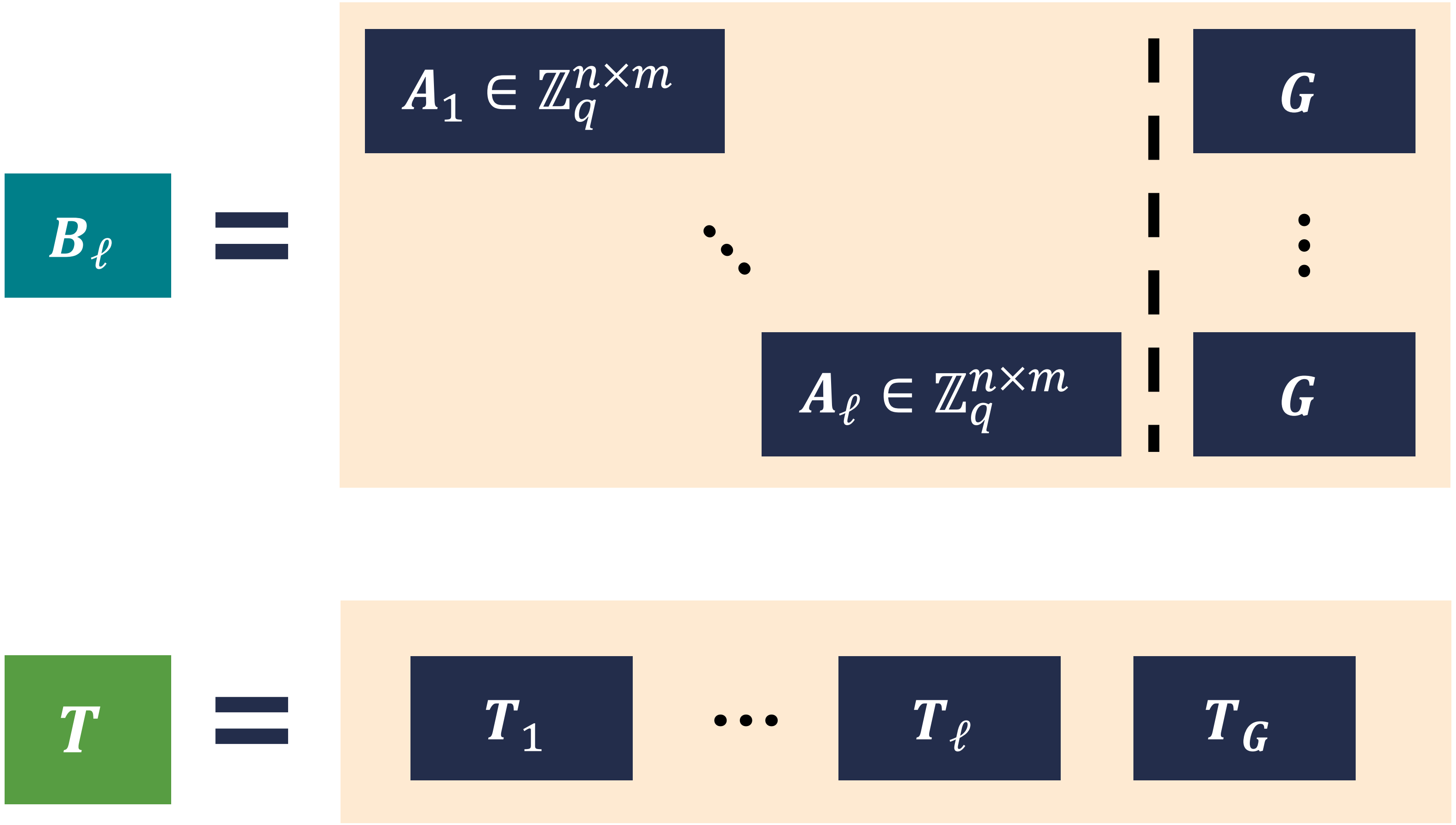
Non-black-box mercurial vector commitment based on BASIS framework

This talk

- Mercurial vector commitments based on $\mathrm{BASIS}_{\mathrm{struct}}$ with smaller auxiliary information support both update and aggregate
- Mercurial vector commitment based on SIS support update
- Redefine the property of update in mercurial vector commitment
  - Introduce new properties: stateless/differential update, updatable mercurial hiding
- Application on Zero-Knowledge Set (ZKS) and Zero-Knowledge Elementary Database(ZK-EDB)
  - Lattice-based updatable ℓ-ary ZKS (ZK-EDB) with batch verification

Starting Point: the BASIS Vector Commitment

Common reference string (CRS)



gadget matrix

trapdoor for **matrix B_ℓ**
$T^\top = B_\ell^{-1}(G_\ell)$

Starting Point: the BASIS Vector Commitment

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

commitment

Basis vector

$e_1 = (1,0,\dots,0)^\top$

opening

(vector with short entries)

Commitment to ℓ-dimensional vector $\boldsymbol{x} \in \mathbb{Z}_q^\ell$

Trapdoor in CRS allows for joint sampling of $(\boldsymbol{c}, \boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell)^\top$ by $\mathrm{SampPre}(\boldsymbol{B}_\ell, \boldsymbol{T}^\top, -\boldsymbol{x} \otimes \boldsymbol{e}_1, s_1)$

Private opening: the commitment **c is statistically close to uniform over $\mathbb{Z}_q^n$**, for all $i \in [\ell]$, the opening **v_i is statistically close to $A_i^{-1}(c - x_i e_1)$**

Our Approach: Extension to BASIS Framework

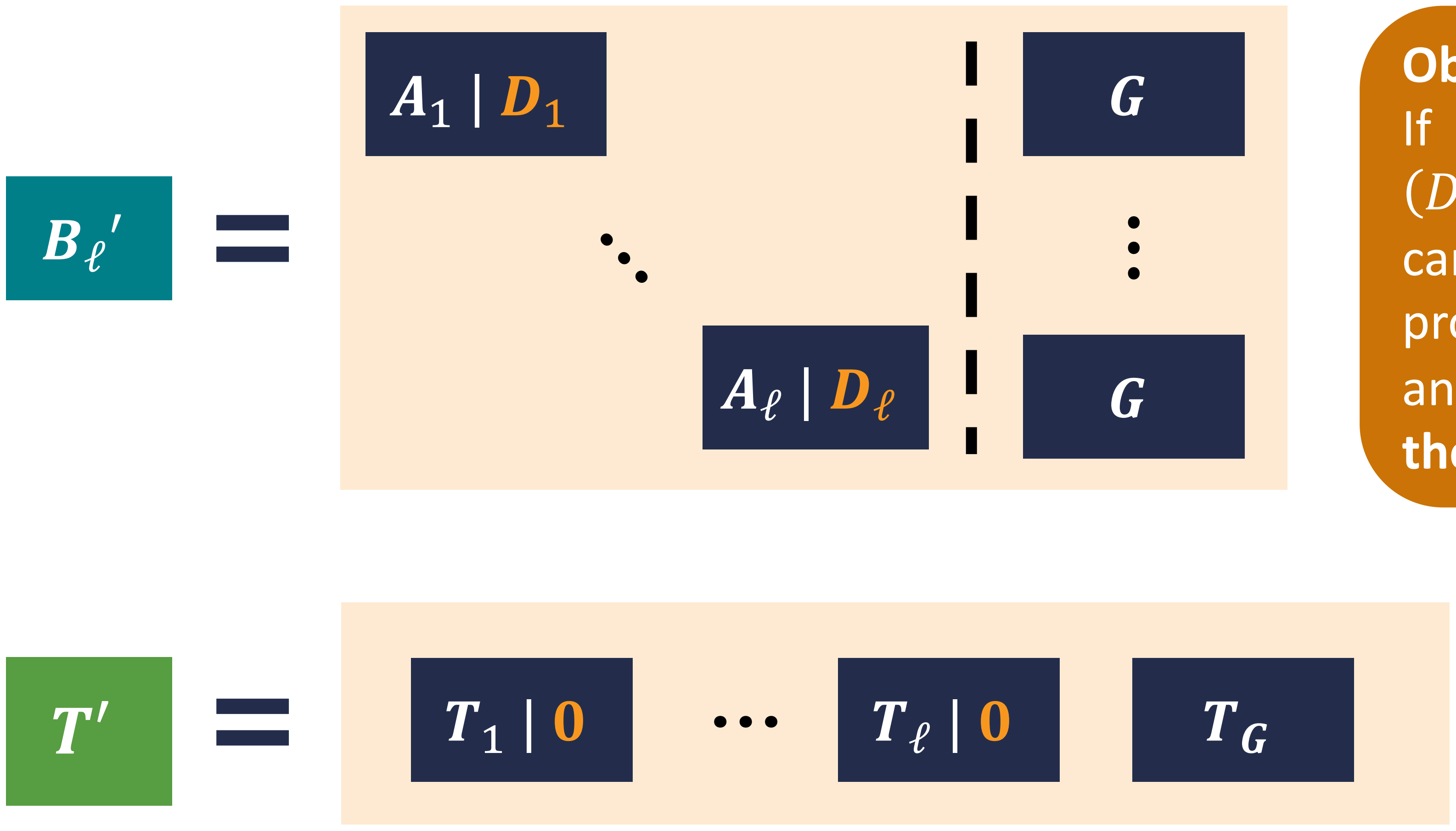
Commitment to ℓ-dimensional vectors $\boldsymbol{x} \in \mathbb{Z}_q^\ell$

Trapdoor in CRS allow for joint sampling of $(\boldsymbol{c}, \boldsymbol{v}_1, \dots, \boldsymbol{v}_\ell)^\top$ by $\mathrm{SampPre}(\boldsymbol{B}_\ell, \boldsymbol{T}^\top, -\boldsymbol{x} \otimes \boldsymbol{e}_1, s_1)$

Private opening: the commitment **$\boldsymbol{c}$ is statistically close to uniform over $\mathbb{Z}_q^n$**, for all $i \in [\ell]$, the opening **$\boldsymbol{v}_i$ is statistically close to $A_i^{-1}(\boldsymbol{c} - x_i \boldsymbol{e}_1)$**

Observation 1:
The private opening implies **simulating algorithms** that can generate a "fake" commitment $\boldsymbol{c}'$ **without any message** and its equivocation opening $\boldsymbol{v}_i{}'$ to x_i **with the trapdoor of A_i**. The distribution of them is **statistically close to the real one**.

Our Approach: Extension to BASIS Framework



Observation 2:
If we extend B_ℓ to B_ℓ' with any $(D_1, \ldots, D_\ell)$, the trapdoor T' of B_ℓ' can be naturally extended and the properties of corretness, binding, and **private opening still hold under the BASIS assumption**

$$B_\ell'(T')^\top = G_\ell, \|T\| = \|T'\|$$

Our Approach: Mercurial Vector Commitment

Mercurial Vector commitment $(\boldsymbol{c}, \boldsymbol{D} = (\boldsymbol{D}_1, \ldots, \boldsymbol{D}_\ell))$

Commitment relation, for all $i \in [\ell], \boldsymbol{c} = [\boldsymbol{A}_i | \boldsymbol{D}_i] \boldsymbol{v}_i + x_i \boldsymbol{e}_1$

In hard commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{A}_i \boldsymbol{R}_i$, the opening $\boldsymbol{v}_i$ can be joint sampled by $\mathrm{SampPre}(\boldsymbol{B}'_\ell, (\boldsymbol{T}')^\top, -\boldsymbol{x} \otimes \boldsymbol{e}_1, s)$

In soft commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{G} - \boldsymbol{A}_i \boldsymbol{R}_i{}'$, the opening $\boldsymbol{v}_i$ can be sampled by $\mathrm{SampPre}([\boldsymbol{A}_i | \boldsymbol{D}_i], \boldsymbol{R}_i{}', \boldsymbol{c} - x_i \boldsymbol{e}_1, s)$

- Since $\boldsymbol{R}_i, \boldsymbol{R}'_i$ are **randomly sampled** over $\{0,1\}^{m \times m'}$, $\boldsymbol{D}_i$ is **indistinguished** in hard commitment and soft commitment
- The **(soft) opening** $\boldsymbol{v}_i$ from **both** hard and soft commitment is **statistically close** to $[\boldsymbol{A}_i | \boldsymbol{D}_i]^{-1}(\boldsymbol{c} - x_i \boldsymbol{e}_1)$
- $\boldsymbol{R}_i$ as an additional part in **hard opening** to check $\boldsymbol{D}_i = \boldsymbol{A}_i \boldsymbol{R}_i$ in hard commitment

Our Approach: Mercurial Vector Commitment

Mercurial Vector commitment $(\boldsymbol{c}, \boldsymbol{D} = (\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell))$

$|D| = \Theta(\ell)$

Commitment relation, for all $i \in [\ell], \boldsymbol{c} = [\boldsymbol{A}_i|\boldsymbol{D}_i]\boldsymbol{v}_i + x_i\boldsymbol{e}_1$

In hard commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{A}_i\boldsymbol{R}_i$, the opening $\boldsymbol{v}_i$ can be joint sampled by $\mathrm{SampPre}(\boldsymbol{B}'_\ell, (\boldsymbol{T}')^\top, -\boldsymbol{x}\otimes\boldsymbol{e}_1, s)$

In soft commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{G} - \boldsymbol{A}_i{\boldsymbol{R}_i}'$, the opening $\boldsymbol{v}_i$ can be sampled by $\mathrm{SampPre}([\boldsymbol{A}_i|\boldsymbol{D}_i], {\boldsymbol{R}_i}', \boldsymbol{c} - x_i\boldsymbol{e}_1, s)$

- Since $\boldsymbol{R}_i, \boldsymbol{R}'_i$ are **randomly sampled** over $\{0,1\}^{m\times m\prime}$, $\boldsymbol{D}_i$ is **indistinguished** in hard commitment and soft commitment
- The **(soft) opening** $\boldsymbol{v}_i$ from **both** hard and soft commitment is **statistically close** to $[\boldsymbol{A}_i|\boldsymbol{D}_i]^{-1}(\boldsymbol{c} - x_i\boldsymbol{e}_1)$
- $\boldsymbol{R}_i$ as an additional part in **hard opening** to check $\boldsymbol{D}_i = \boldsymbol{A}_i\boldsymbol{R}_i$ in hard commitment

Our Approach: Instantiation on $\mathrm{BASIS}_{\mathrm{struct}}$

Mercurial Vector commitment $(\boldsymbol{c}, \boldsymbol{D} = (\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell))$

In hard commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{A}_i \boldsymbol{R}_i$

In soft commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{G} - \boldsymbol{A}_i \boldsymbol{R}_i{}'$

In $\mathrm{BASIS}_{\mathrm{struct}}$ assumption, $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell$ are structured by $\boldsymbol{A}_i = \boldsymbol{W}_i \boldsymbol{A}$, where $\boldsymbol{W}_i$ is a **pubilc** random invertible matrix for all $i \in [\ell]$ and $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ is sampled randomly.

So, $(D_1, \dots, D_\ell)$ can be structured by $\boldsymbol{D}_i = \boldsymbol{W}_i \widehat{\boldsymbol{D}}$ for all $i \in [\ell]$, where

$$\widehat{\boldsymbol{D}} = \boldsymbol{A}\boldsymbol{R} \text{ or } \widehat{\boldsymbol{D}} = \boldsymbol{G} - \boldsymbol{A}\boldsymbol{R},$$

where $\boldsymbol{R}$ is **randomly sampled** over $\{0, 1\}^{m \times m'}$.

Therefore, the commitment can be compressed to $(\boldsymbol{c}, \widehat{\boldsymbol{D}})$.

Our Approach: Instantiation on SIS

Mercurial Vector commitment $(\boldsymbol{c}, \boldsymbol{D} = (\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell))$

In hard commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{A}_i \boldsymbol{R}_i$

In soft commitment, for all $i \in [\ell], \boldsymbol{D}_i = \boldsymbol{G} - \boldsymbol{A}_i \boldsymbol{R}_i{}'$

Unlike $\mathrm{BASIS}_{\mathrm{struct}}$ assumption, $\boldsymbol{A}_1, \dots, \boldsymbol{A}_\ell$ are randomly sampled **independently**, so $\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell$ are **independent** as well.

We solve the problem using a **standard vector commitment**: we commit $(\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell)$ to σ, and then publish $(\boldsymbol{c}, \sigma)$ instead of $(\boldsymbol{c}, \boldsymbol{D} = (\boldsymbol{D}_1, \dots, \boldsymbol{D}_\ell))$.

Although this method will cause the **same size of the auxiliary information** as the generic framework, we want to emphasize this it can **support update** due to its ***non-black-box* construction**

This Work

This talk

Non-black-box mercurial vector commitment based on BASIS framework

- Mercurial vector commitments based on $\mathrm{BASIS}_{\mathrm{struct}}$ with smaller auxiliary information support both update and aggregate

  [see paper for details]

- Mercurial vector commitment based on SIS support update
- Redefine the property of update in mercurial vector commitment
  - Introduce new properties: stateless/differential update, updatable mercurial hiding
- Application on Zero-Knowledge Set (ZKS) and Zero-Knowledge Elementary Database(ZK-EDB)
  - Lattice-based updatable ℓ-ary ZKS (ZK-EDB) with batch verification

Thank you !