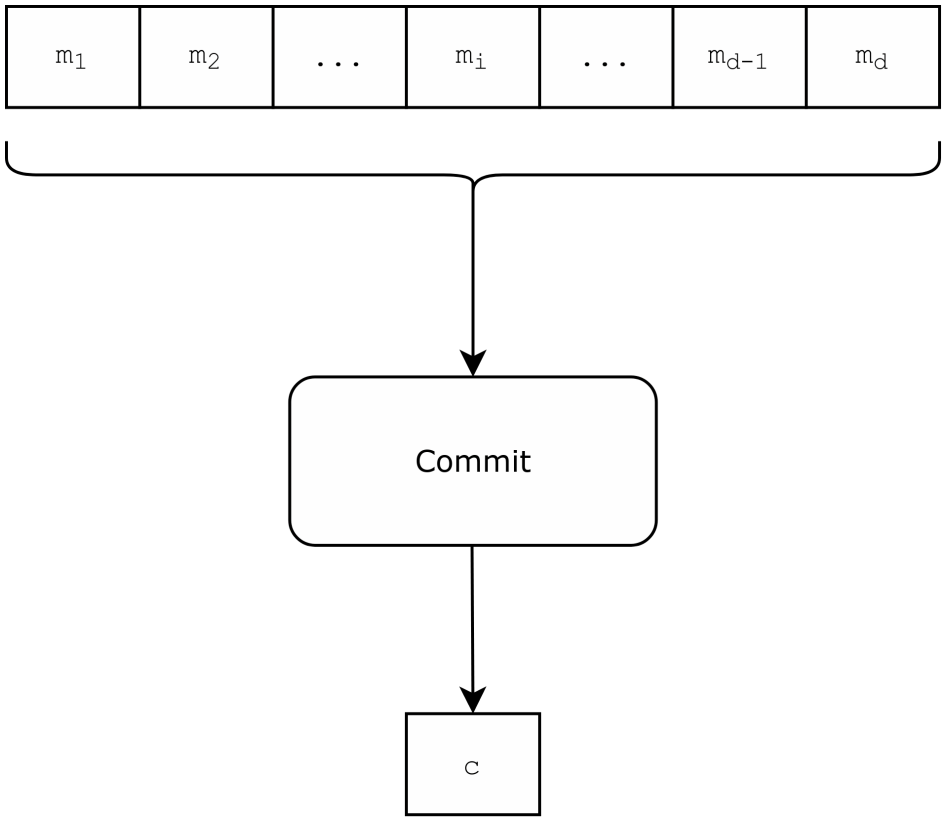
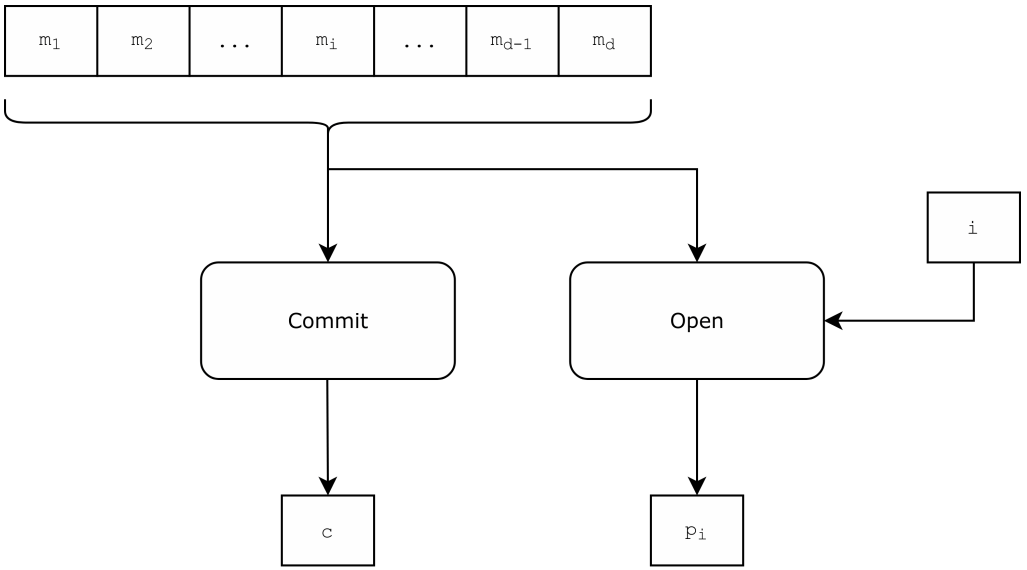
Vector and Functional Commitments from Lattices

Chris Peikert, Zachary Pepin, Chad Sharp

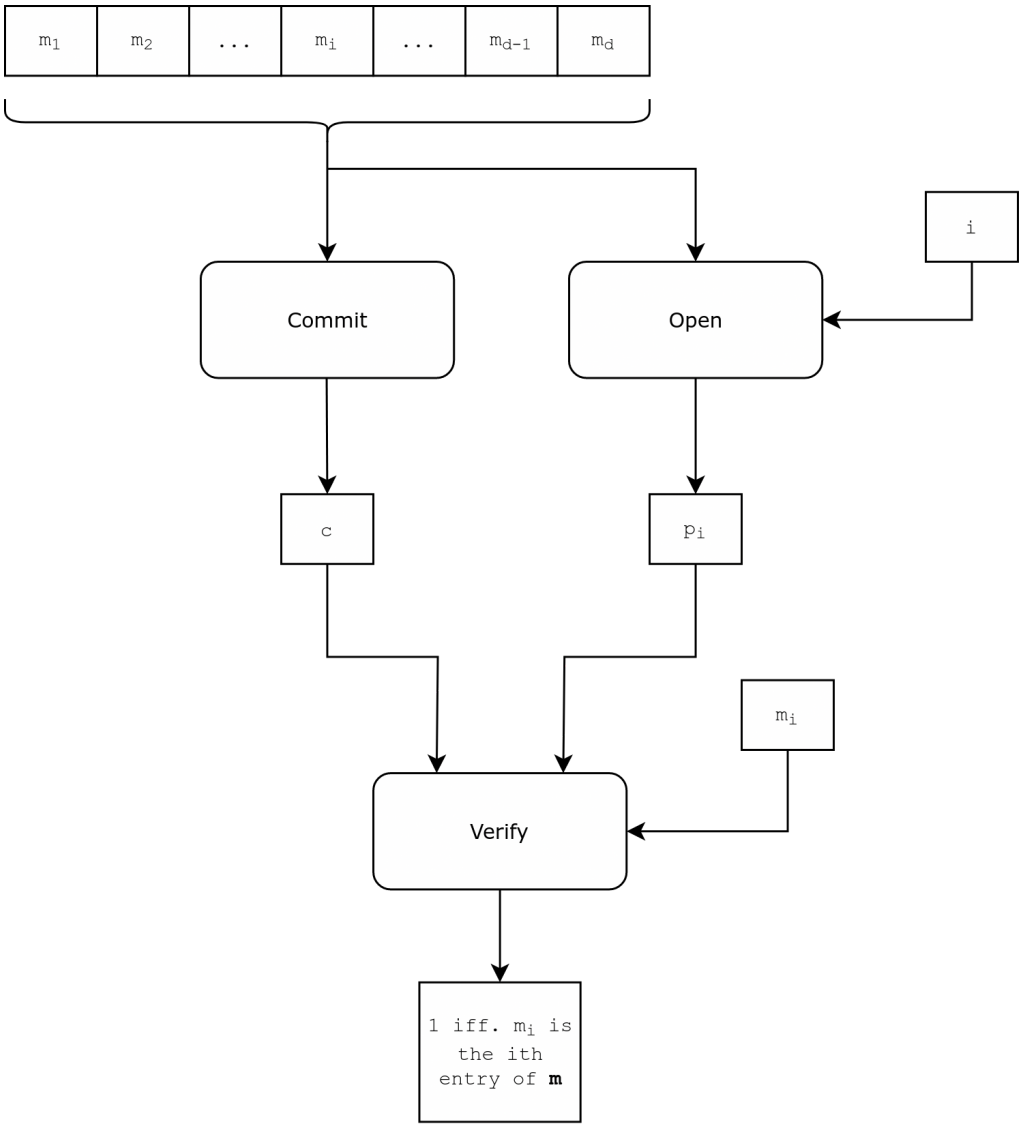
Vector Commitments [LY10, CF13]



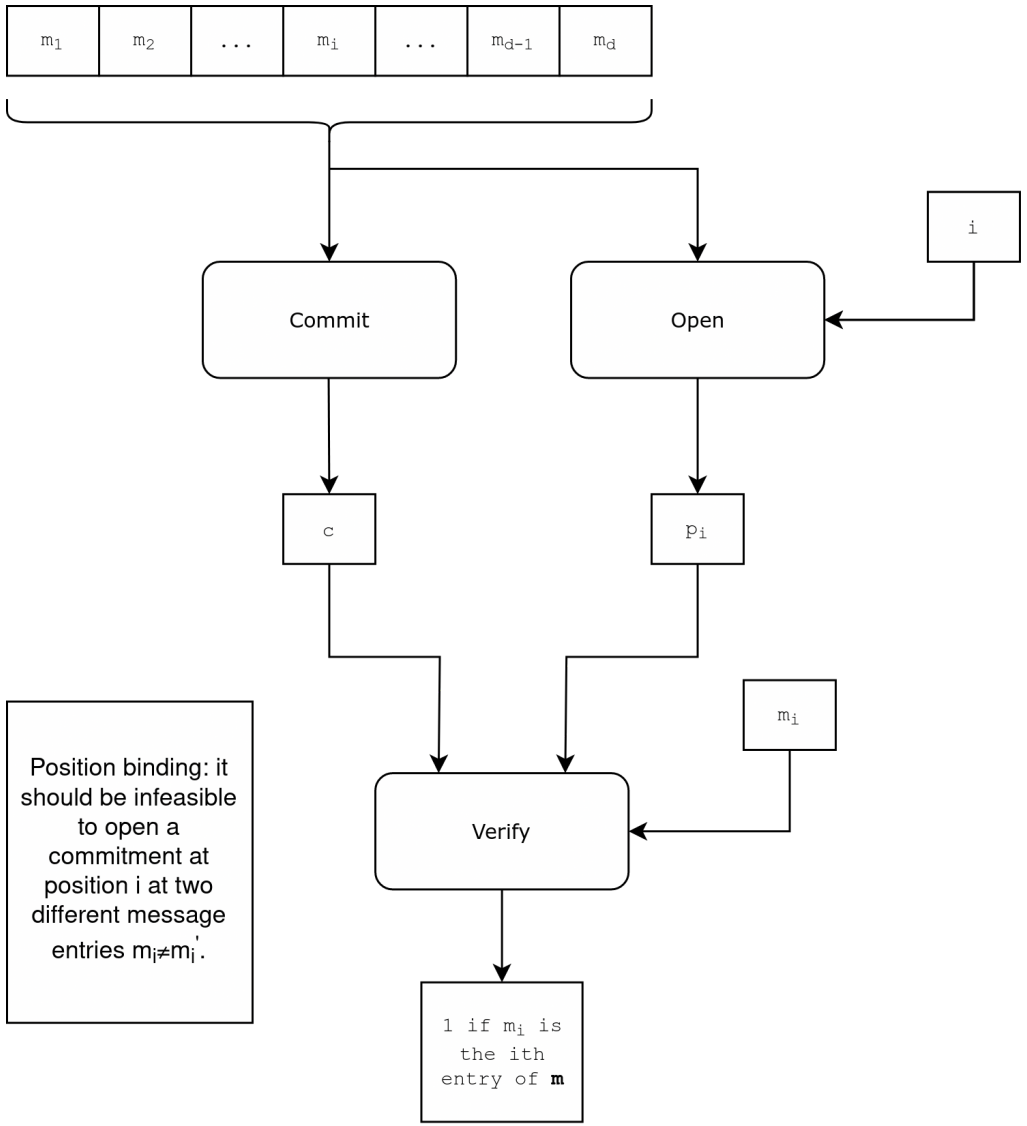
Vector Commitments [LY10, CF13]



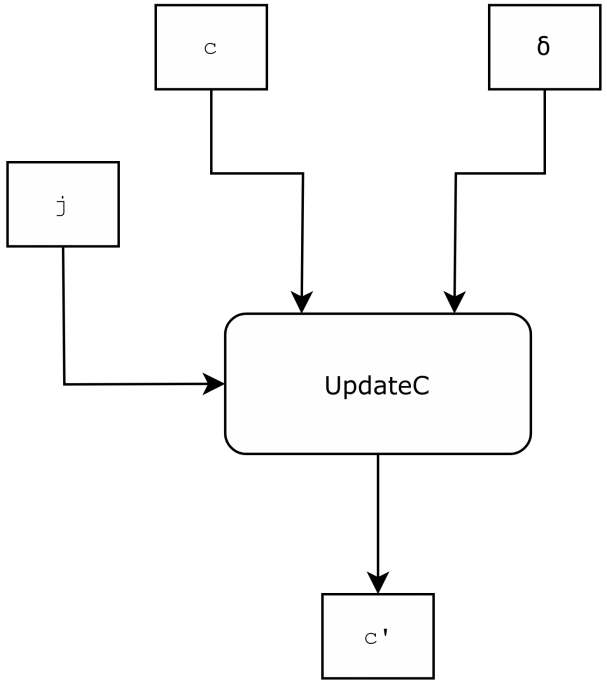
Vector Commitments [LY10, CF13]

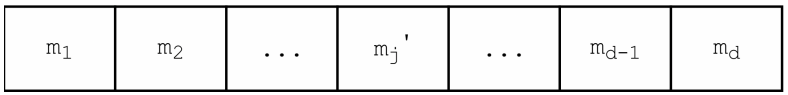


Vector Commitments [LY10, CF13]

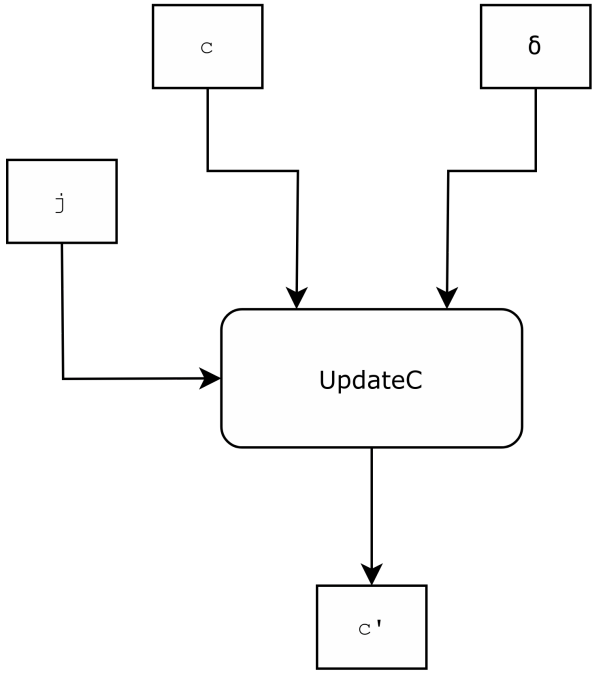


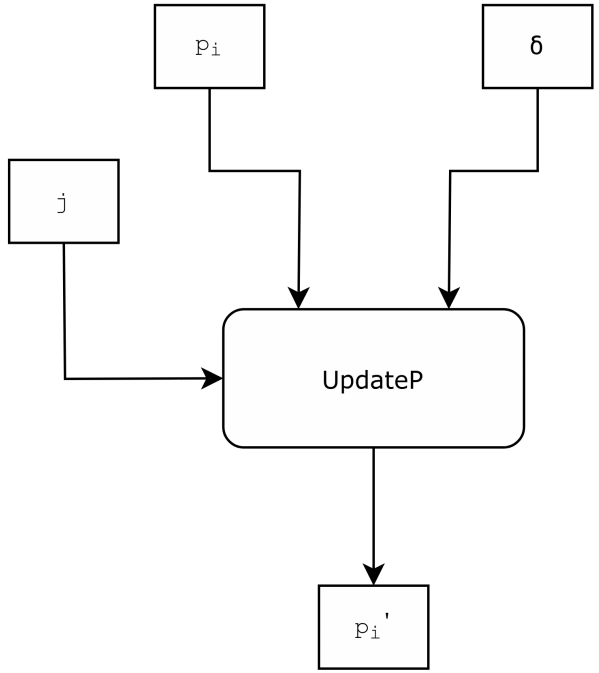
Vector Commitments: (Stateless) Updates

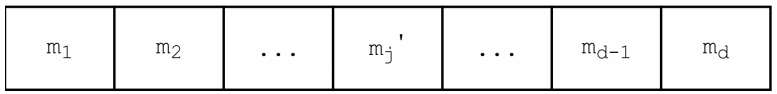




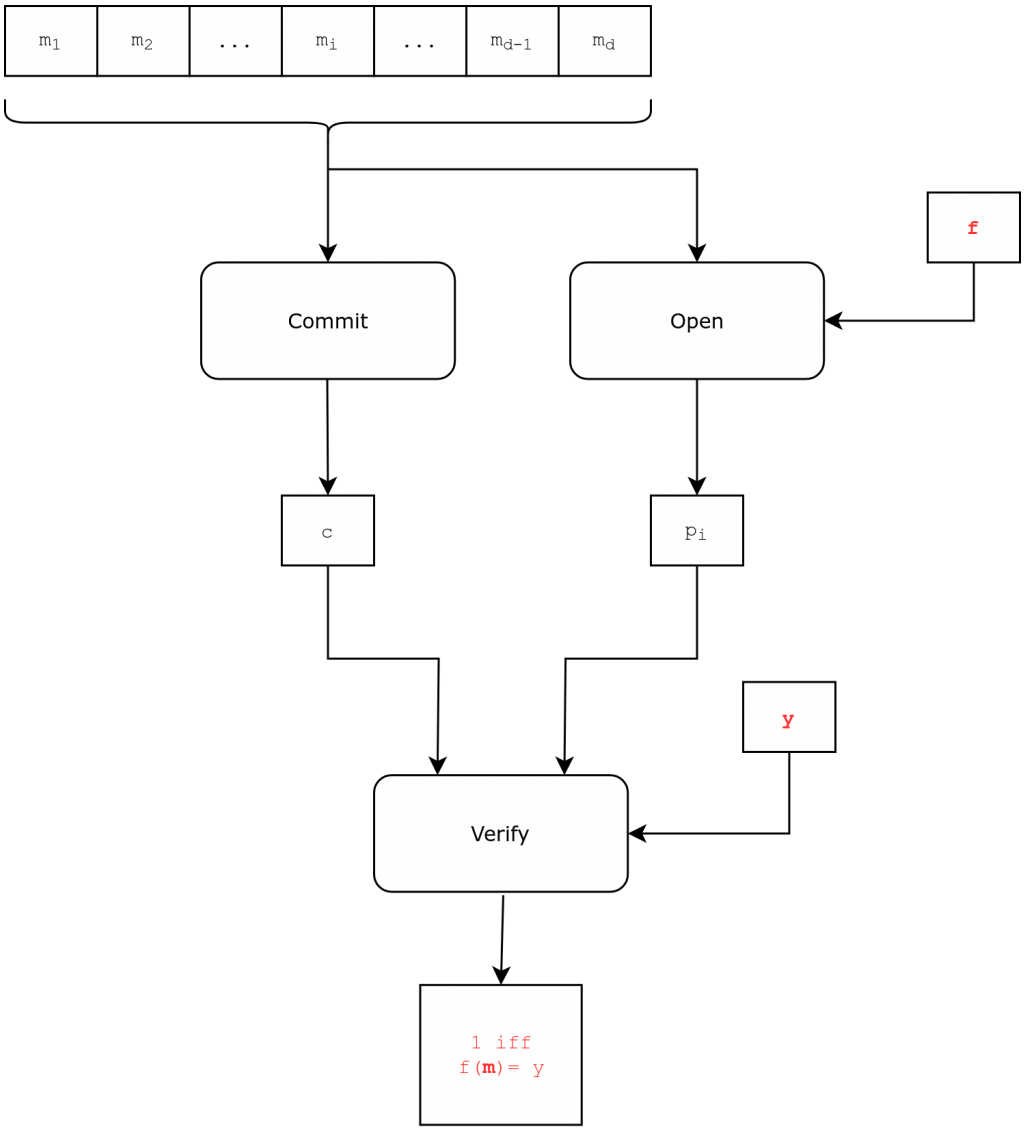
Vector Commitments: (Stateless) Updates



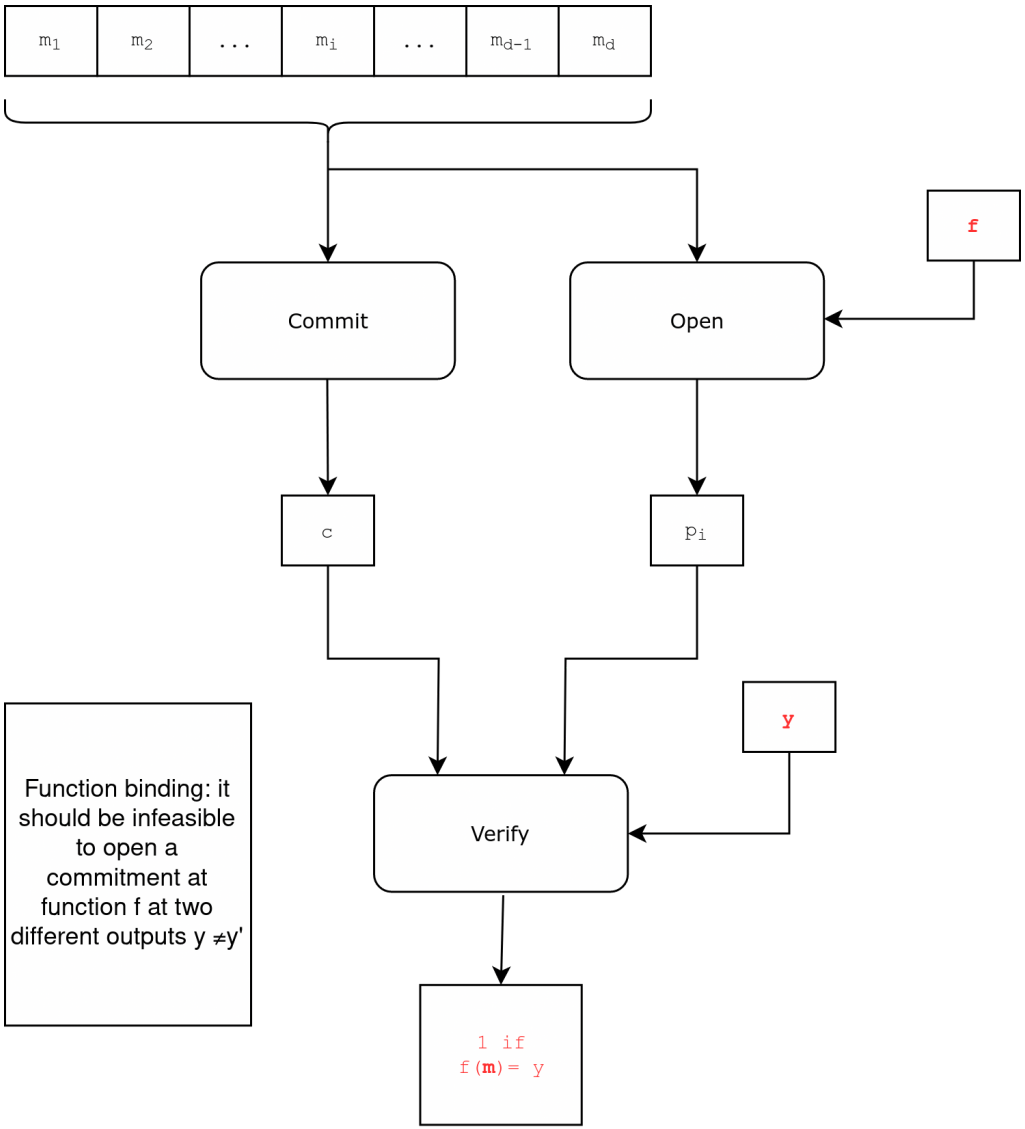




Functional Commitments [LRY16]



Functional Commitments [LRY16]



Prior work

- Merkle trees [Mer87], not statelessly updatable.

Prior work

- Merkle trees [Mer87], not statelessly updatable.
- Statelessly updatable VCs based on RSA, pairings [LY10, CF13].

Prior work

- Merkle trees [Mer87], not statelessly updatable.
- Statelessly updatable VCs based on RSA, pairings [LY10, CF13].
- Merkle-like statelessly updatable VC scheme based on SIS (post quantum) [PSTY13].

Prior work

- Merkle trees [Mer87], not statelessly updatable.
- Statelessly updatable VCs based on RSA, pairings [LY10, CF13].
- Merkle-like statelessly updatable VC scheme based on SIS (post quantum) [PSTY13].
- Applications: verifiable oursourcing of storage [CF13, BGV11], verifiable zero knowledge sets [MRK03], cryptographic accumulators [BdM93], pseudononymous credentials [KZG10], cryptocurrencies [CPSZ18].

Prior work

- FC scheme for linear functions [LRY16] based on pairings.

Prior work

- FC scheme for linear functions [LRY16] based on pairings.
- FC scheme for “sparse polynomials” [LP20].

Prior work

- FC scheme for linear functions [LRY16] based on pairings.
- FC scheme for “sparse polynomials” [LP20].

Prior work

- FC scheme for linear functions [LRY16] based on pairings.
- FC scheme for “sparse polynomials” [LP20].
- SNARKs for NP let us go further than linearizable functions [LRY16], but these cannot be constructed from falsifiable assumptions [GW11].

Prior work

- FC scheme for linear functions [LRY16] based on pairings.
- FC scheme for “sparse polynomials” [LP20].
- SNARKs for NP let us go further than linearizable functions [LRY16], but these cannot be constructed from falsifiable assumptions [GW11].
- Applications: verifiable secret sharing [LRY10], content extraction signatures [SBZ01], and zero-knowledge SNARKS [BFS20, BDFG20].

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.

Scheme	$\lvert pp\rvert$	$\lvert c\rvert$	$\lvert \pi\rvert$	Setup	PQ
[PSTY13] (SIS)	$h^2 d$	$h \log d$	$h^3 d \log^2 d$	Public	✓
Tree construction (SIS)	$h^2 d^2$	$h \log d$	$h^3 \log^2 d$	Private	✓

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.
   - First post-quantum FC scheme from a falsifiable assumption.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.
   - First post-quantum FC scheme from a falsifiable assumption.
   - Works in a new model in which a permanently online authority generates reusable “opening keys” for desired functions.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.
   - First post-quantum FC scheme from a falsifiable assumption.
   - Works in a new model in which a permanently online authority generates reusable “opening keys” for desired functions.

Secondary contributions:

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.
   - First post-quantum FC scheme from a falsifiable assumption.
   - Works in a new model in which a permanently online authority generates reusable “opening keys” for desired functions.

Secondary contributions:

1. Formal definition and generic construction of a zero-knowledge vector commitment scheme.

Contributions

1. New post-quantum (SIS), statelessly updatable VC with significantly shorter proofs than the only other one.
2. New SIS-based FC scheme for *arbitrary (bounded) Boolean circuits*.
   - First to go beyond linearizable functions based on a falsifiable assumption.
   - First post-quantum FC scheme from a falsifiable assumption.
   - Works in a new model in which a permanently online authority generates reusable “opening keys” for desired functions.

Secondary contributions:

1. Formal definition and generic construction of a zero-knowledge vector commitment scheme.
2. Formal analysis of a (folklore) Merkle-like tree transformation for VC schemes that makes them suitable for vectors of large arity.

Short Integer Solution ($\mathsf{SIS}_{n,m,q,\beta}$)

Given uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find nonzero $\mathbf{x} \in \mathbb{Z}^m$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$$

and

$$\|\mathbf{x}\| \leq \beta.$$

SIS-based VC scheme

Setup$(1^n, 1^d)$:

- Sample $\mathbf{U} = [\mathbf{u}_0 \mid \cdots \mid \mathbf{u}_{d-1}] \in \mathbb{Z}_q^{n \times d}$ u.a.r.

SIS-based VC scheme

Setup$(1^n, 1^d)$:

- Sample $\mathbf{U} = [\mathbf{u}_0 \mid \cdots \mid \mathbf{u}_{d-1}] \in \mathbb{Z}_q^{n \times d}$ u.a.r.
- For all distinct $i, j \in [d]$, sample $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ and (short) $\mathbf{r}_{i,j} \in \mathbb{Z}^m$ (via trapdoor pre-image sampling) such that:

$$\mathbf{A}_i \mathbf{r}_{i,j} = \mathbf{u}_j.$$

- Output $\mathbf{A}_i$s, $\mathbf{r}_{i,j}$s, and $\mathbf{U}$ as public parameters.

SIS-based VC scheme

Setup$(1^n, 1^d)$:

- Sample $\mathbf{U} = [\mathbf{u}_0 \mid \cdots \mid \mathbf{u}_{d-1}] \in \mathbb{Z}_q^{n \times d}$ u.a.r.
- For all distinct $i, j \in [d]$, sample $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ and (short) $\mathbf{r}_{i,j} \in \mathbb{Z}^m$ (via trapdoor pre-image sampling) such that:

$$\mathbf{A}_i \mathbf{r}_{i,j} = \mathbf{u}_j.$$

- Output $\mathbf{A}_i$s, $\mathbf{r}_{i,j}$s, and $\mathbf{U}$ as public parameters.

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{d-1} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{r}_{0,1} & \cdots & \mathbf{r}_{0,d-1} \\ \mathbf{r}_{1,0} & \mathbf{0} & \cdots & \mathbf{r}_{1,d-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_{d-1,0} & \mathbf{r}_{d-1,1} & \cdots & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{R}}} = \begin{bmatrix} \mathbf{0} & \mathbf{u}_1 & \cdots & \mathbf{u}_{d-1} \\ \mathbf{u}_0 & \mathbf{0} & \cdots & \mathbf{u}_{d-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{0} \end{bmatrix}$$

SIS-based VC scheme

$\mathsf{Commit}_{pp}(\mathbf{m} \in \{0,1\}^d)$:

$$\mathbf{c} := \mathbf{U}\mathbf{m}$$

SIS-based VC scheme

$\mathsf{Commit}_{pp}(\mathbf{m} \in \{0,1\}^d)$:

$$\mathbf{c} := \mathbf{U}\mathbf{m}$$

$\mathsf{Open}_{pp}(\mathbf{m} \in \{0,1\}^d, i)$:

$$\mathbf{p}_i = \tilde{\mathbf{R}}_{i,*}\mathbf{m}$$

SIS-based VC scheme

$\mathsf{Commit}_{pp}(\mathbf{m} \in \{0,1\}^d)$:

$$\mathbf{c} := \mathbf{U}\mathbf{m}$$

$\mathsf{Open}_{pp}(\mathbf{m} \in \{0,1\}^d, i)$:

$$\begin{aligned}\mathbf{p}_i &= \tilde{\mathbf{R}}_{i,*}\mathbf{m} \\ &= \begin{bmatrix}\mathbf{r}_{i,0} & \cdots & \mathbf{r}_{i,i-1} & \mathbf{0} & \mathbf{r}_{i,i+1} & \cdots & \mathbf{r}_{i,d-1}\end{bmatrix}\mathbf{m}\end{aligned}$$

SIS-based VC scheme

$\mathsf{Commit}_{pp}(\mathbf{m} \in \{0,1\}^d)$:

$$\mathbf{c} := \mathbf{U}\mathbf{m}$$

$\mathsf{Open}_{pp}(\mathbf{m} \in \{0,1\}^d, i)$:

$$\begin{aligned}\mathbf{p}_i &= \tilde{\mathbf{R}}_{i,*}\mathbf{m} \\ &= \begin{bmatrix}\mathbf{r}_{i,0} & \cdots & \mathbf{r}_{i,i-1} & \mathbf{0} & \mathbf{r}_{i,i+1} & \cdots & \mathbf{r}_{i,d-1}\end{bmatrix}\mathbf{m}\end{aligned}$$

$\mathsf{Verify}_{pp}(\mathbf{c}, i, m_i, \mathbf{p}_i)$:

accept iff. $\mathbf{p}_i$ is sufficiently short and $\mathbf{c} = \mathbf{A}_i\mathbf{p}_i + m_i\mathbf{u}_i$

SIS-based VC scheme

$\mathsf{Commit}_{pp}(\mathbf{m} \in \{0,1\}^d)$:

$$\mathbf{c} := \mathbf{U}\mathbf{m}$$

$\mathsf{Open}_{pp}(\mathbf{m} \in \{0,1\}^d, i)$:

$$\begin{aligned}\mathbf{p}_i &= \tilde{\mathbf{R}}_{i,*}\mathbf{m} \\ &= \begin{bmatrix}\mathbf{r}_{i,0} & \cdots & \mathbf{r}_{i,i-1} & \mathbf{0} & \mathbf{r}_{i,i+1} & \cdots & \mathbf{r}_{i,d-1}\end{bmatrix}\mathbf{m}\end{aligned}$$

$\mathsf{Verify}_{pp}(\mathbf{c}, i, m_i, \mathbf{p}_i)$:

accept iff. $\mathbf{p}_i$ is sufficiently short and $\mathbf{c} = \mathbf{A}_i\mathbf{p}_i + m_i\mathbf{u}_i$

$$\mathbf{A}_i(\tilde{\mathbf{R}}_{i,*}\mathbf{m}) + m_i\mathbf{u}_i = \sum_{j\neq i} m_j\mathbf{A}_i\mathbf{r}_{i,j} + m_i\mathbf{u}_i = \sum_{j\neq i} m_j\mathbf{u}_j + m_i\mathbf{u}_i = \mathbf{U}\mathbf{m} = \mathbf{c}$$

SIS-based VC scheme: Updates

UpdateC$_{pp}(\mathbf{c}, j, \boldsymbol{\delta})$:

$$\mathbf{c}' = \mathbf{c} + \boldsymbol{\delta}\mathbf{u}_j$$

SIS-based VC scheme: Updates

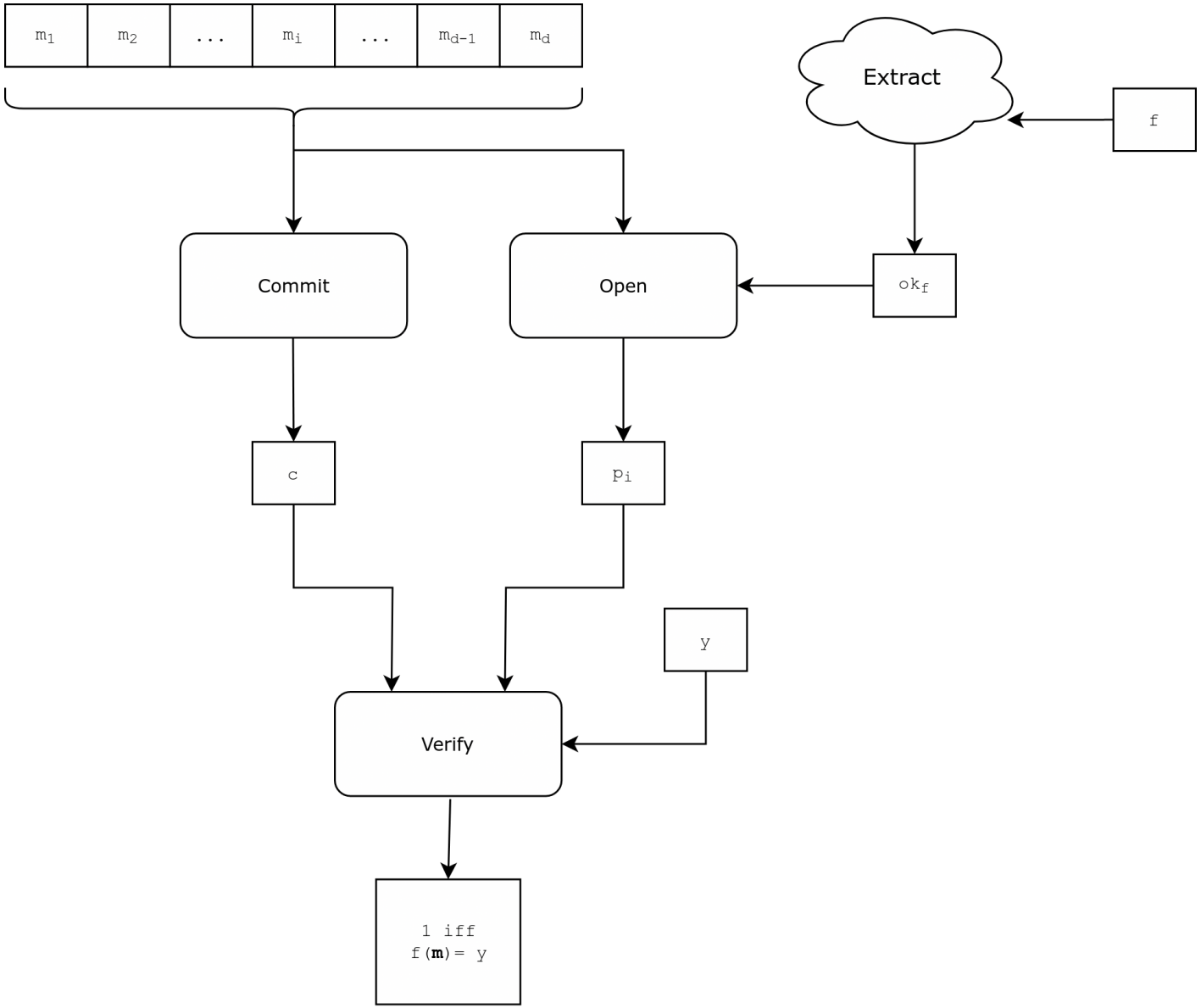
$\mathsf{UpdateC}_{pp}(\mathbf{c}, j, \delta)$:

$$\mathbf{c}' = \mathbf{c} + \delta \mathbf{u}_j$$

$\mathsf{UpdateP}_{pp}(\mathbf{p}_i, j, \delta)$:

$$\mathbf{p}'_i = \mathbf{p}_i + \delta \mathbf{r}_{i,j}$$

Functional Commitments with Authority



Homomorphic commitments [GSW13]

$$\mathsf{Com}_{\mathbf{A}}(\mathbf{x}; \mathbf{R}) = \mathbf{AR} + \mathsf{Encode}(\mathbf{x})$$

Homomorphic commitments [GSW13]

$$\mathrm{Com}_{\mathbf{A}}(\mathbf{x}; \mathbf{R}) = \mathbf{AR} + \mathrm{Encode}(\mathbf{x})$$

- $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is some public matrix.

Homomorphic commitments [GSW13]

$$\mathrm{Com}_{\mathbf{A}}(\mathbf{x}; \mathbf{R}) = \mathbf{AR} + \mathrm{Encode}(\mathbf{x})$$

- $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is some public matrix.
- $\mathbf{R} \in \mathbb{Z}^{m \times w}$ is sufficiently "short" randomness.

Homomorphic commitments [GSW13]

$$\mathrm{Com}_{\mathbf{A}}(\mathbf{x}; \mathbf{R}) = \mathbf{AR} + \mathrm{Encode}(\mathbf{x})$$

- $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is some public matrix.
- $\mathbf{R} \in \mathbb{Z}^{m \times w}$ is sufficiently "short" randomness.
- Additive and multiplicative homomorphisms.

Homomorphic commitments [GSW13]

$$\mathsf{Eval}(f, \mathbf{C}_x[, \mathbf{R}_x]) \text{ outputs } (\mathbf{C}_{x,f}[, \mathbf{R}_{x,f}])$$

such that $\mathbf{R}_{x,f}$ is still short and if

$$\mathbf{C}_x = \mathsf{Com}_{\mathbf{A}}(x; \mathbf{R}_x),$$

then

$$\mathbf{C}_{x,f} = \mathsf{Com}_{\mathbf{A}}(f(x), \mathbf{R}_{x,f}).$$

SIS-based FC scheme

Setup(1^n):

- Generate matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathbf{T}$ and uniformly random $\mathbf{C}$.

SIS-based FC scheme

Setup(1^n):

- Generate matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathbf{T}$ and uniformly random $\mathbf{C}$.
- Store $ek = (\mathbf{C}, \mathbf{A}, \mathbf{T})$, output $pp = (\mathbf{C}, \mathbf{A})$.

SIS-based FC scheme

Setup(1^n):

- Generate matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathbf{T}$ and uniformly random $\mathbf{C}$.
- Store $ek = (\mathbf{C}, \mathbf{A}, \mathbf{T})$, output $pp = (\mathbf{C}, \mathbf{A})$.
- We use tagged trapdoor techniques to map $\mathbf{A}$ to a unique $\mathbf{A}_f$ for each $f \in \mathcal{F}$.

SIS-based FC scheme

Setup(1^n):

- Generate matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathbf{T}$ and uniformly random $\mathbf{C}$.
- Store $ek = (\mathbf{C}, \mathbf{A}, \mathbf{T})$, output $pp = (\mathbf{C}, \mathbf{A})$.
- We use tagged trapdoor techniques to map $\mathbf{A}$ to a unique $\mathbf{A}_f$ for each $f \in \mathcal{F}$.
- Think of $\mathbf{C}$ as a superposition of commitments to all $f \in \mathcal{F}$.

SIS-based FC scheme

$\mathsf{Extract}_{ek}(f)$:

- Use $\mathbf{T}$ to generate a witness $\mathbf{R}_f$ that $\mathbf{C}$ is a commitment to f wrt $\mathbf{A}_f$.
- That is, generate an $\mathbf{R}_f$ such that $\mathbf{C} = \mathsf{Com}_{\mathbf{A}_f}(f; \mathbf{R}_f)$.

SIS-based FC scheme

$\mathsf{Extract}_{ek}(f)$:

- Use $\mathbf{T}$ to generate a witness $\mathbf{R}_f$ that $\mathbf{C}$ is a commitment to f wrt $\mathbf{A}_f$.
- That is, generate an $\mathbf{R}_f$ such that $\mathbf{C} = \mathsf{Com}_{\mathbf{A}_f}(f; \mathbf{R}_f)$.

$\mathsf{Commit}_{pp}(\mathbf{m})$:

- output $\mathbf{C_m} = \mathsf{Eval}(U_{\mathbf{m}}, \mathbf{C})$ where $U_{\mathbf{m}}(f) = f(\mathbf{m})$.
- Think of $\mathbf{C_m}$ as a superposition of commitments to $f(\mathbf{m})$ for all $f \in \mathcal{F}$.

SIS-based FC scheme

$\mathsf{Extract}_{ek}(f)$:

- Use $\mathbf{T}$ to generate a witness $\mathbf{R}_f$ that $\mathbf{C}$ is a commitment to f wrt $\mathbf{A}_f$.
- That is, generate an $\mathbf{R}_f$ such that $\mathbf{C} = \mathsf{Com}_{\mathbf{A}_f}(f; \mathbf{R}_f)$.

$\mathsf{Commit}_{pp}(\mathbf{m})$:

- output $\mathbf{C_m} = \mathsf{Eval}(U_{\mathbf{m}}, \mathbf{C})$ where $U_{\mathbf{m}}(f) = f(\mathbf{m})$.
- Think of $\mathbf{C_m}$ as a superposition of commitments to $f(\mathbf{m})$ for all $f \in \mathcal{F}$.

$\mathsf{Open}_{pp}(\mathbf{m}, f, ok_f = \mathbf{R}_f)$:

- compute $(\mathbf{C_m}; \mathbf{R}_{\mathbf{m},f}) = \mathsf{Eval}(U_{\mathbf{m}}, \mathbf{C}; \mathbf{R}_f)$ and output $\mathbf{R}_{\mathbf{m},f}$.

SIS-based FC scheme

$\mathsf{Extract}_{ek}(f)$:

- Use $\mathbf{T}$ to generate a witness $\mathbf{R}_f$ that $\mathbf{C}$ is a commitment to f wrt $\mathbf{A}_f$.
- That is, generate an $\mathbf{R}_f$ such that $\mathbf{C} = \mathsf{Com}_{\mathbf{A}_f}(f; \mathbf{R}_f)$.

$\mathsf{Commit}_{pp}(\mathbf{m})$:

- output $\mathbf{C_m} = \mathsf{Eval}(U_{\mathbf{m}}, \mathbf{C})$ where $U_{\mathbf{m}}(f) = f(\mathbf{m})$.
- Think of $\mathbf{C_m}$ as a superposition of commitments to $f(\mathbf{m})$ for all $f \in \mathcal{F}$.

$\mathsf{Open}_{pp}(\mathbf{m}, f, ok_f = \mathbf{R}_f)$:

- compute $(\mathbf{C_m}; \mathbf{R}_{\mathbf{m},f}) = \mathsf{Eval}(U_{\mathbf{m}}, \mathbf{C}; \mathbf{R}_f)$ and output $\mathbf{R}_{\mathbf{m},f}$.

$\mathsf{Verify}_{pp}(\mathbf{C_m}, f, \mathbf{y}, p_{\mathbf{m},f} = \mathbf{R}_{\mathbf{m},f})$:

- accept if $\mathbf{R}_{\mathbf{m},f}$ is short and $\mathbf{C_m} = \mathsf{Com}_{\mathbf{A}_f}(\mathbf{y}; \mathbf{R}_{\mathbf{m},f})$