Fully-Succinct Multi-Key Homomorphic Signatures from Standard Assumptions

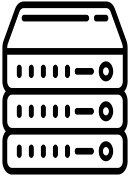
Gaspard Anthoine, **David Balbás**, Dario Fiore
IMDEA Software Institute, Madrid, Spain

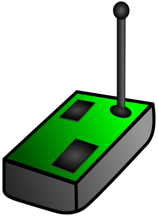
19th August 2024

CRYPTO 2024

Computing on Authenticated Data

A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.

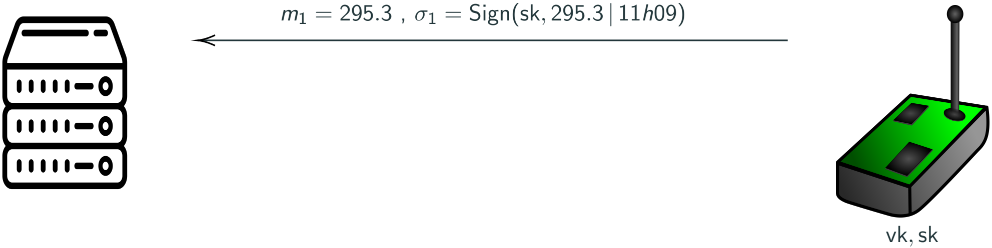




vk, sk

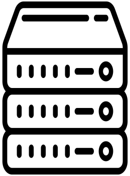
Computing on Authenticated Data

A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.



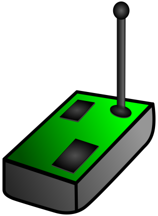
Computing on Authenticated Data

A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.



$m_1 = 295.3$, $\sigma_1 = \mathsf{Sign}(\mathsf{sk}, 295.3 \,|\, 11h09)$

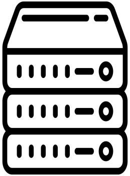
$m_2 = 295.6$, $\sigma_2 = \mathsf{Sign}(\mathsf{sk}, 295.6 \,|\, 11h10)$



vk, sk

Computing on Authenticated Data

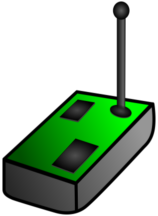
A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.



$m_1 = 295.3$, $\sigma_1 = \mathsf{Sign}(\mathsf{sk}, 295.3 \,|\, 11h09)$

$m_2 = 295.6$, $\sigma_2 = \mathsf{Sign}(\mathsf{sk}, 295.6 \,|\, 11h10)$

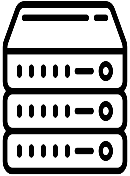
$m_3 = 295.8$, $\sigma_3 = \mathsf{Sign}(\mathsf{sk}, 295.8 \,|\, 11h11)$



vk, sk

Computing on Authenticated Data

A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.

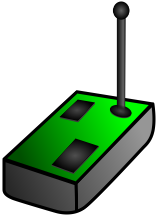


Compute MAX temp
from 0h00 to 23h59

$m_1 = 295.3\ ,\ \sigma_1 = \mathsf{Sign}(\mathsf{sk}, 295.3 \,|\, 11h09)$

$m_2 = 295.6\ ,\ \sigma_2 = \mathsf{Sign}(\mathsf{sk}, 295.6 \,|\, 11h10)$

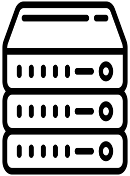
$m_3 = 295.8\ ,\ \sigma_3 = \mathsf{Sign}(\mathsf{sk}, 295.8 \,|\, 11h11)$



vk, sk

Computing on Authenticated Data

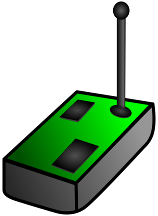
A sensor sends temperature data m_i every minute. m_i and a timestamp ℓ_i are signed.



$m_1 = 295.3$, $\sigma_1 = \mathsf{Sign}(\mathsf{sk}, 295.3 \,|\, 11h09)$

$m_2 = 295.6$, $\sigma_2 = \mathsf{Sign}(\mathsf{sk}, 295.6 \,|\, 11h10)$

$m_3 = 295.8$, $\sigma_3 = \mathsf{Sign}(\mathsf{sk}, 295.8 \,|\, 11h11)$



Compute MAX temp
from 0h00 to 23h59

vk, sk

Can we have a short, publicly verifiable proof that the MAX temperature is computed correctly on **today***'s* **authentic** *temperatures?*

Homomorphic Signatures [JMSW02]

vk, sk

Homomorphic Signatures [JMSW02]

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$

vk, sk

Homomorphic Signatures [JMSW02]

Evaluate $\boldsymbol{y} = f(m_1, \ldots, m_n)$

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$ ←

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$ ←

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$ ←

vk, sk

Homomorphic Signatures [JMSW02]

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$ ⟵

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$ ⟵

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$ ⟵

vk, sk

Evaluate $\boldsymbol{y} = f(m_1, \ldots, m_n)$

$\mathsf{Eval}(f, \boldsymbol{\ell}, \boldsymbol{m}, \mathsf{vk}, \boldsymbol{\sigma}) \longrightarrow \boldsymbol{y}, \sigma_{f,y}$

Homomorphic Signatures [JMSW02]

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$

vk, sk

Evaluate $\boldsymbol{y} = f(m_1, \ldots, m_n)$

$\mathsf{Eval}(f, \boldsymbol{\ell}, \boldsymbol{m}, \mathsf{vk}, \boldsymbol{\sigma}) \longrightarrow \boldsymbol{y}, \sigma_{f,y}$

Homomorphic Signatures [JMSW02]

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$

vk, sk

Evaluate $\boldsymbol{y} = f(m_1, \ldots, m_n)$

$\mathsf{Eval}(f, \boldsymbol{\ell}, \boldsymbol{m}, \mathsf{vk}, \boldsymbol{\sigma}) \longrightarrow \boldsymbol{y}, \sigma_{f,y}$

- $\sigma_{f,y}$ is publicly verifiable from f, vk, $\boldsymbol{y}$ and labels ℓ_i.

Homomorphic Signatures [JMSW02]

$m_1, \sigma_1 = \mathsf{Sign}(\mathsf{sk}, m_1, \ell_1)$

$m_2, \sigma_2 = \mathsf{Sign}(\mathsf{sk}, m_2, \ell_2)$

$m_3, \sigma_3 = \mathsf{Sign}(\mathsf{sk}, m_3, \ell_3)$

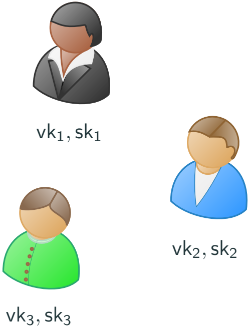
vk, sk

Evaluate $\boldsymbol{y} = f(m_1, \ldots, m_n)$

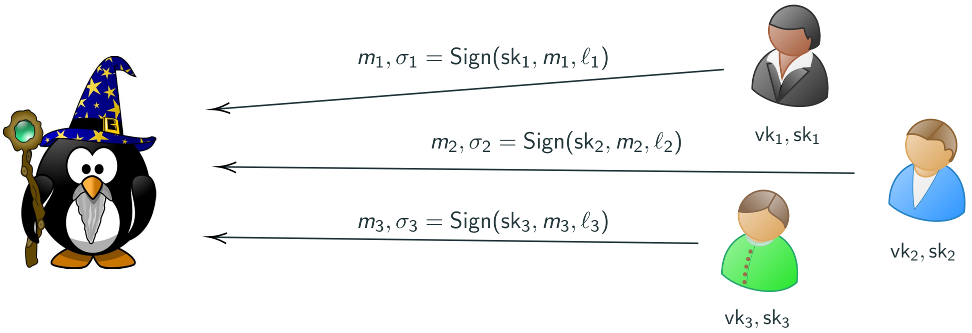
$\mathsf{Eval}(f, \boldsymbol{\ell}, \boldsymbol{m}, \mathsf{vk}, \boldsymbol{\sigma}) \longrightarrow \boldsymbol{y}, \sigma_{f,y}$

- $\sigma_{f,y}$ is publicly verifiable from f, vk, $\boldsymbol{y}$ and labels ℓ_i.
- $\sigma_{f,y}$ is *succinct*: does not grow with n or $|f|$.

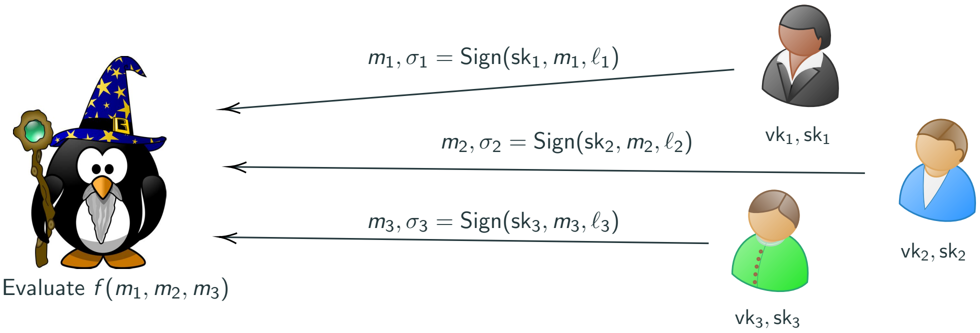
Multi-Key Homomorphic Signatures [FMNP16]



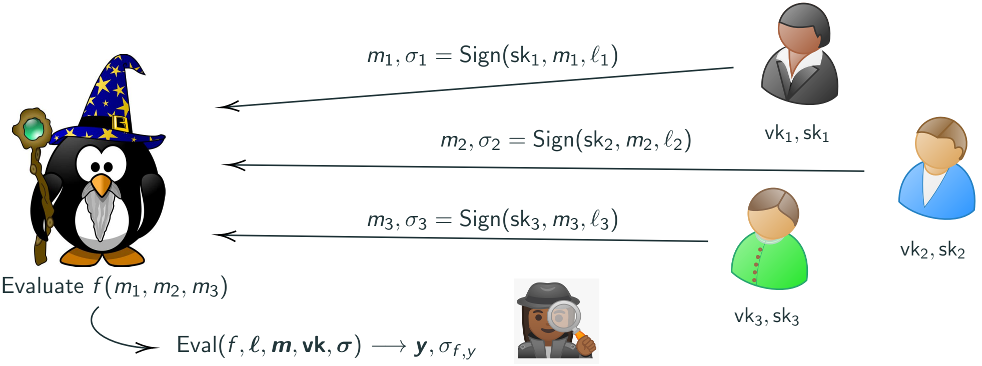
Multi-Key Homomorphic Signatures [FMNP16]



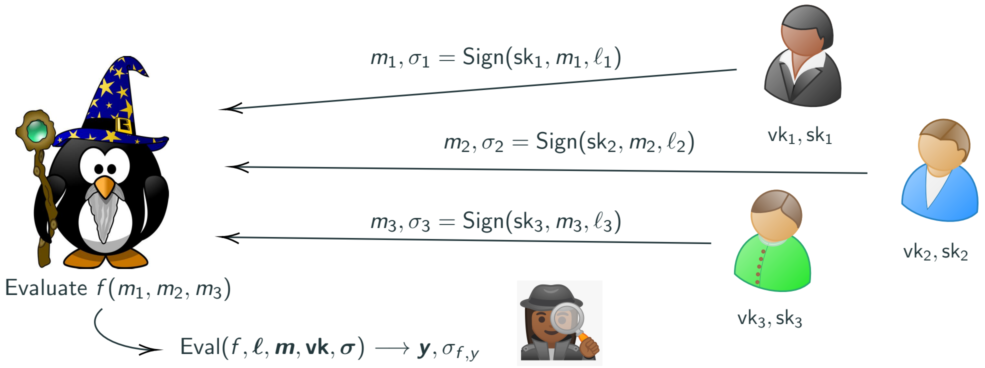
Multi-Key Homomorphic Signatures [FMNP16]



Multi-Key Homomorphic Signatures [FMNP16]

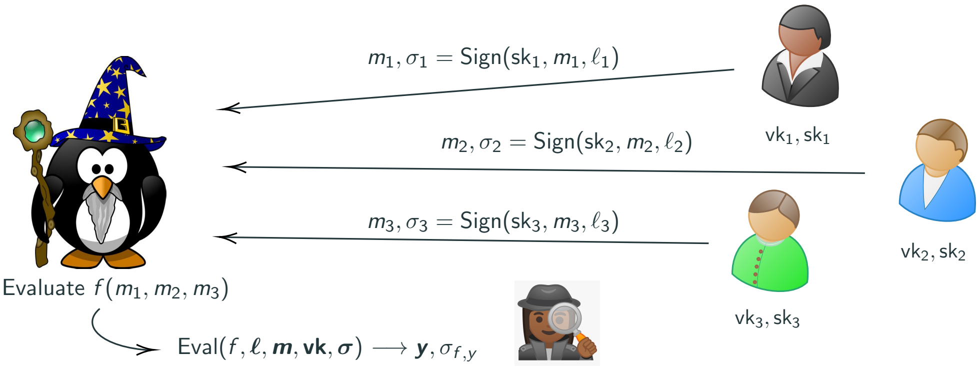


Multi-Key Homomorphic Signatures [FMNP16]



- $\mathsf{Setup}(1^\lambda)$, $\mathsf{KeyGen}(\mathsf{pp})$
- $\mathsf{Sign}(\mathsf{sk}, m, \ell) \to \sigma$
- $\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \to \sigma_{f,y}$
- $\mathsf{Ver}(\mathsf{pp}, (f, \boldsymbol{\ell}), \mathbf{vk}, \boldsymbol{y}, \sigma_{f,y}) \to 0/1$

Multi-Key Homomorphic Signatures [FMNP16]



- $\mathsf{Setup}(1^\lambda)$, $\mathsf{KeyGen}(\mathsf{pp})$
- $\mathsf{Sign}(\mathsf{sk}, m, \ell) \to \sigma$
- $\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \to \sigma_{f,y}$
- $\mathsf{Ver}(\mathsf{pp}, (f, \boldsymbol{\ell}), \mathbf{vk}, \boldsymbol{y}, \sigma_{f,y}) \to 0/1$

- **Succinctness:** $|\sigma_{f,y}| \leq p(\lambda)$. Succinct in:
  - nº of inputs n,
  - function size $|f|$,
  - *nº of parties t.*

Our Contribution

- Succinct *single-key HS* from standard assumptions are known [BF11, CFW14, GVW15, CFT22, BCFL23, GU24, Goy24, . . .]

Our Contribution

- Succinct *single-key HS* from standard assumptions are known [BF11, CFW14, GVW15, CFT22, BCFL23, GU24, Goy24, ...]
- However, no MKHS for all functions was *fully-succinct.*
  - [FMNP16] standard model, $|\sigma_{f,y}| = \text{poly}(t, \log n)$
  - [LTWC18] fully succinct, SNARK-based.

Our Contribution

- Succinct *single-key HS* from standard assumptions are known [BF11, CFW14, GVW15, CFT22, BCFL23, GU24, Goy24, . . .]
- However, no MKHS for all functions was *fully-succinct.*
  - [FMNP16] standard model, $|\sigma_{f,y}| = \mathrm{poly}(t, \log n)$
  - [LTWC18] fully succinct, SNARK-based.

Our Result: *fully-succinct MKHS from standard and falsifiable assumptions*.

Our Contribution

- Succinct *single-key HS* from standard assumptions are known [BF11, CFW14, GVW15, CFT22, BCFL23, GU24, Goy24, ...]
- However, no MKHS for all functions was *fully-succinct.*
  - [FMNP16] standard model, $|\sigma_{f,y}| = \text{poly}(t, \log n)$
  - [LTWC18] fully succinct, SNARK-based.

Our Result: *fully-succinct MKHS from standard and falsifiable assumptions*.

✓ Adaptive security, (sequential) multi-hop evaluation, pre-processing.

✓ Instantiations from e.g. k-Lin or LWE.

✗ Non black-box use of cryptographic primitives.

Roadmap

Batch arguments for NP: aggregating signatures

Roadmap

Batch arguments for NP: aggregating signatures $\Longrightarrow$ **Homomorphic evaluation: functional commitments**

Roadmap

Batch arguments for NP: aggregating signatures

$\Rightarrow$

Homomorphic evaluation: functional commitments

$\Downarrow$

Achieving security: Linking messages and commitments

Roadmap

Batch arguments for NP:
aggregating signatures

$\Rightarrow$

Homomorphic evaluation:
functional commitments

$\Downarrow$

Achieving security:
Linking messages and commitments

$\Leftarrow$

Instantiations

Batch Arguments for NP [KPY19, CJJ21]

Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.

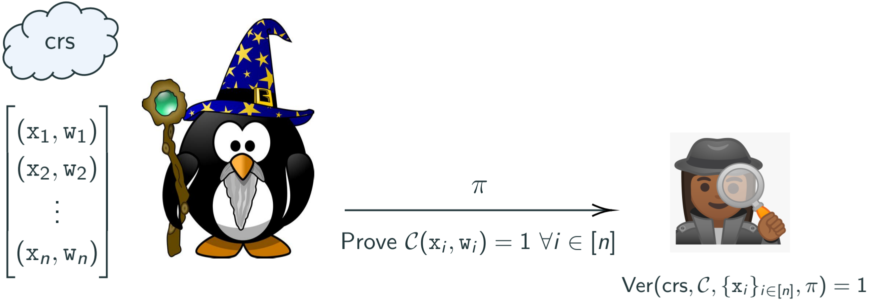
Batch Arguments for NP [KPY19, CJJ21]

Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.

$$\begin{bmatrix} (\mathtt{x}_1, \mathtt{w}_1) \\ (\mathtt{x}_2, \mathtt{w}_2) \\ \vdots \\ (\mathtt{x}_n, \mathtt{w}_n) \end{bmatrix}$$

Batch Arguments for NP [KPY19, CJJ21]

Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.



- **Succinctness:** $|\pi| = \mathrm{poly}(\lambda, |\mathcal{C}|, \log n)$.

Batch Arguments for NP [KPY19, CJJ21]

Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.

crs′

$$\begin{bmatrix} (\mathtt{x}_1, \mathtt{w}_1) \\ (\mathtt{x}_2, \mathtt{w}_2) \\ \vdots \\ (\mathtt{x}_n, \mathtt{w}_n) \end{bmatrix}$$

td, i^*

π

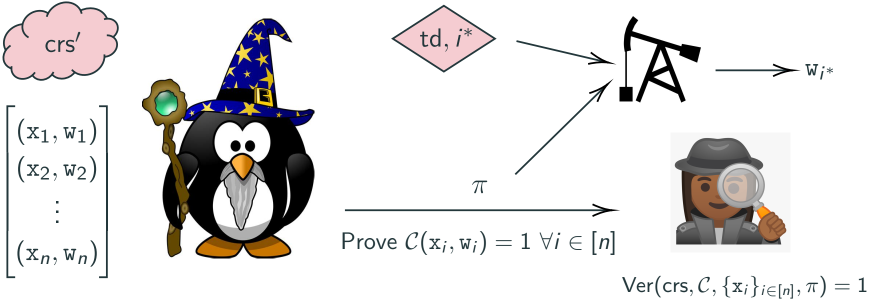
Prove $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1 \ \forall i \in [n]$

$\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i \in [n]}, \pi) = 1$

- **Succinctness:** $|\pi| = \mathrm{poly}(\lambda, |\mathcal{C}|, \log n)$.

Batch Arguments for NP [KPY19, CJJ21]

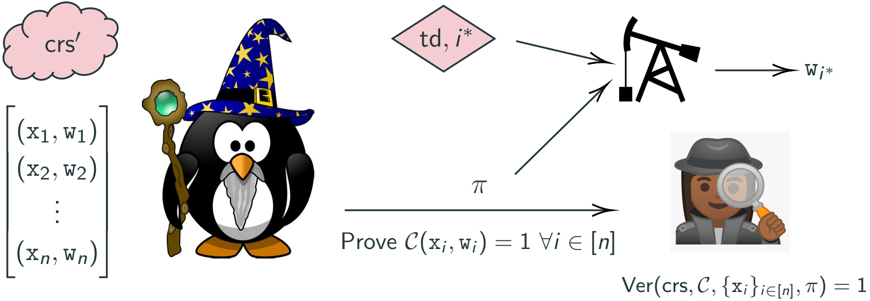
Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.



- **Succinctness:** $|\pi| = \mathrm{poly}(\lambda, |\mathcal{C}|, \log n)$.
- **Somewhere extractability:** td extracts a valid $\mathtt{w}_{i^*}$.

Batch Arguments for NP [KPY19, CJJ21]

Let $(\mathtt{x}_1, \mathtt{w}_1), \ldots, (\mathtt{x}_n, \mathtt{w}_n)$ be statement-witness pairs from an NP relation $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$.



Aggregate *n* signatures [WW22, DGKV22]: Let $\mathtt{x}_i = (\mathsf{vk}_i, m_i)$, $\mathtt{w}_i = \sigma_i$. Prove

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i, \sigma_i) = 1$$

Homomorphic Evaluation of f

- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \ \wedge \ \boldsymbol{y} = f(m_1, \ldots, m_n).$$

Homomorphic Evaluation of f

- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \; \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \;\wedge\; \boldsymbol{y} = f(m_1, \ldots, m_n).$$

Issue: f is not local - can't use a BARG.

Homomorphic Evaluation of f

- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \ \wedge \ \boldsymbol{y} = f(m_1, \ldots, m_n).$$

  Issue: f is not local - can't use a BARG.
- **Better:** Commit to $(m_1, \ldots, m_n)$, use a *Functional Commitment* for f!

Homomorphic Evaluation of f

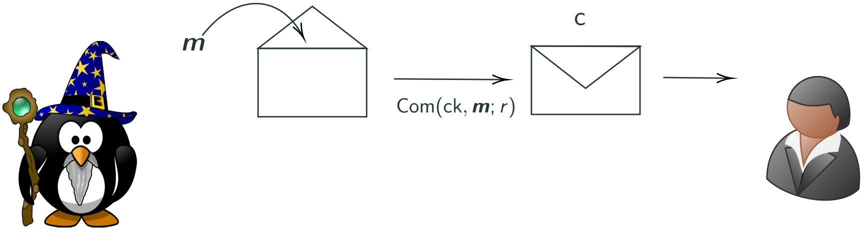
- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \ \wedge \ \boldsymbol{y} = f(m_1, \ldots, m_n).$$

  Issue: f is not local - can't use a BARG.

- **Better:** Commit to $(m_1, \ldots, m_n)$, use a *Functional Commitment* for f!

  A FC allows one to commit to $\boldsymbol{m}$ and later open the commitment to $f(\boldsymbol{m})$.

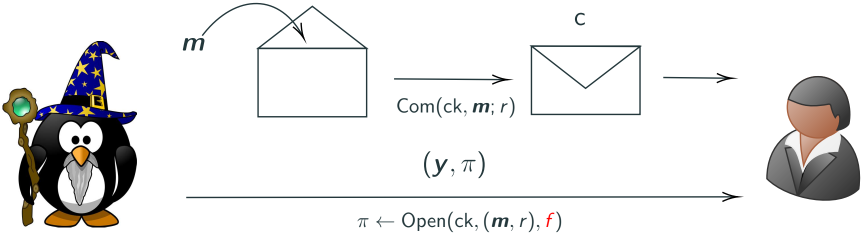


Homomorphic Evaluation of f

- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \ \wedge \ \boldsymbol{y} = f(m_1, \ldots, m_n).$$

  Issue: f is not local - can't use a BARG.
- **Better:** Commit to $(m_1, \ldots, m_n)$, use a *Functional Commitment* for f!
  A FC allows one to commit to $\boldsymbol{m}$ and later open the commitment to $f(\boldsymbol{m})$.

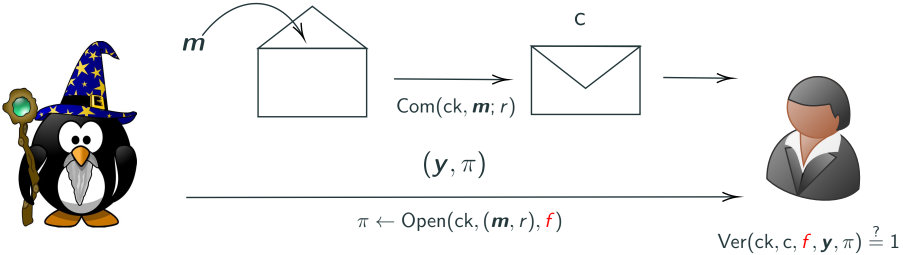


Homomorphic Evaluation of f

- **Naive attempt:** Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$, $\mathtt{w}_i = (m_i, \sigma_i)$ and prove:

$$\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :} \ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \ \wedge \ \boldsymbol{y} = f(m_1, \ldots, m_n).$$

  Issue: f is not local - can't use a BARG.
- **Better:** Commit to $(m_1, \ldots, m_n)$, use a *Functional Commitment* for f!
  A FC allows one to commit to $\boldsymbol{m}$ and later open the commitment to $f(\boldsymbol{m})$.



Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}$ $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}\ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}\ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Correct, but insecure...

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}\ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Correct, but insecure... the committed m_i may differ from $\mathtt{w}_i$!

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}\ \Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Correct, but insecure... the committed m_i may differ from $\mathtt{w}_i$!

Solution: iteratively compute c inside $\mathcal{C}$.

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}$ $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Correct, but insecure... the committed m_i may differ from $\mathtt{w}_i$!

Solution: iteratively compute c inside $\mathcal{C}$.

- Define *partial* $\mathsf{c}_0, \ldots, \mathsf{c}_n$, where c_i commits to $(m_1, \ldots, m_i, 0, \ldots, 0)$.

Putting Everything Together

Candidate MKHS:

- Let $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i)$ and $\mathtt{w}_i = (m_i, \sigma_i)$.
- Do a *BARG proof* for $\underline{\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) :}$ $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1$.
- Obtain $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$ and *open* c to $f(m_1, \ldots, m_n)$.

Correct, but insecure... the committed m_i may differ from $\mathtt{w}_i$!

Solution: iteratively compute c inside $\mathcal{C}$.

- Define *partial* $\mathsf{c}_0, \ldots, \mathsf{c}_n$, where c_i commits to $(m_1, \ldots, m_i, 0, \ldots, 0)$.
- $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ checks that *c_i and c_{i-1} differ on m_i* at position i.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i | \ell_i, \sigma_i) = 1 \wedge$

  $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$

- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$
  $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \rightarrow \sigma_{f,y}$:

Compute:

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \rightarrow \sigma_{f,y}$:

Compute:

- $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \rightarrow \sigma_{f,y}$:

Compute:

- $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.
- A BARG proof π_σ for $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \rightarrow \sigma_{f,y}$:

Compute:

- $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.
- A BARG proof π_σ for $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$.
- A FC opening proof π_f that c opens to $\boldsymbol{y} = f(m_1, \ldots, m_n)$ on f.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \rightarrow \sigma_{f,y}$:

Compute:

- $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.
- A BARG proof π_σ for $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$.
- A FC opening proof π_f that c opens to $\boldsymbol{y} = f(m_1, \ldots, m_n)$ on f.

Output $\sigma_{f,y} = (\mathsf{c}, \pi_\sigma, \pi_f)$.

Our MKHS Construction

Description of $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ (simplified):

Statement: $\mathtt{x}_i = (\mathsf{vk}_i, \ell_i, \mathsf{ck}_i, i)$

Witness: $\mathtt{w}_i = (m_i, \sigma_i, \pi_i, \mathsf{c}_{i-1}, \mathsf{c}_i)$

- Check $\Sigma.\mathsf{Ver}(\mathsf{vk}_i, m_i|\ell_i, \sigma_i) = 1 \wedge$ $\mathsf{FC.VerUpd}(\mathsf{ck}_i, i, \mathsf{c}_{i-1}, 0, \mathsf{c}_i, m_i, \pi_i) = 1$
- If $i = 1$, check $\mathsf{c}_{i-1} = \mathsf{FC.Com}(\mathsf{ck}, \mathbf{0})$.
- If $i = n$, check $\mathsf{c}_i = \mathsf{c}$.

$\mathsf{Sign}(\mathsf{sk}_i, m_i, \ell_i)$:

Output $\sigma_i \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_i, m_i|\ell_i)$

$\mathsf{Eval}(\mathsf{pp}, (f, \boldsymbol{\ell}), \boldsymbol{m}, \mathbf{vk}, \boldsymbol{\sigma}) \to \sigma_{f,y}$:

Compute:

- $\mathsf{c} \leftarrow \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.
- A BARG proof π_σ for $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$.
- A FC opening proof π_f that c opens to $\boldsymbol{y} = f(m_1, \ldots, m_n)$ on f.

Output $\sigma_{f,y} = (\mathsf{c}, \pi_\sigma, \pi_f)$.

For the security proof to work, we also need a somewhere extractable commitment (SEC).

Instantiations

Instantiations

MKHS with optimal succinctness (via KLVW23*)

From subexponential DDH or LWE, there exists a MKHS for boolean circuits where:

- $|\mathsf{pp}| = |\sigma_{f,y}| = \mathsf{poly}(\lambda, \log n)$

Instantiations

MKHS with optimal succinctness (via KLVW23*)

From subexponential DDH or LWE, there exists a MKHS for boolean circuits where:

- $|\mathsf{pp}| = |\sigma_{f,y}| = \mathsf{poly}(\lambda, \log n)$

MKHS from algebraic primitives (via WW22, BCFL23)

From HiKer and k-Lin for $k \geq 2$, there exists a MKHS for arithmetic circuits of width w where:

- $|\mathsf{pp}| = \mathcal{O}(w^5)$
- $|\sigma_{f,y}| = \mathcal{O}(\lambda \cdot d^2) + \mathsf{poly}(\lambda)$.

Instantiations

MKHS with optimal succinctness (via KLVW23*)

From subexponential DDH or LWE, there exists a MKHS for boolean circuits where:

- $|\mathsf{pp}| = |\sigma_{f,y}| = \mathsf{poly}(\lambda, \log n)$

MKHS from algebraic primitives (via WW22, WW24)

From k-Lin for $k \geq 2$, there exists a MKHS for arithmetic circuits *of size s* where:

- $|\mathsf{pp}| = \mathcal{O}(s^5)$
- $|\sigma_{f,y}| = \mathsf{poly}(\lambda)$.

Open Questions

Make a **fully algebraic** and practical construction?

Support arbitrary **multi-hop evaluation**? [ACG24]

Apply similar techniques to **other advanced primitives**?

Takeaways

- Multi-key homomorphic signatures:
  verifiable computation on signed data.

Takeaways

- Multi-key homomorphic signatures:
  verifiable computation on signed data.
- We build **fully succinct** MKHS from *standard* assumptions (LWE, k-Lin).

Takeaways

- Multi-key homomorphic signatures:
  verifiable computation on signed data.
- We build **fully succinct** MKHS from *standard* assumptions (LWE, k-Lin).
- We combine batch arguments for NP (**BARGs**) and functional commitments (**FCs**).

Takeaways

- Multi-key homomorphic signatures:
  verifiable computation on signed data.
- We build **fully succinct** MKHS from *standard* assumptions (LWE, k-Lin).
- We combine batch arguments for NP (**BARGs**) and functional commitments (**FCs**).
- *Exciting open questions* - let's chat!

Takeaways

- Multi-key homomorphic signatures: *verifiable computation on signed data*.
- We build **fully succinct** MKHS from *standard* assumptions (LWE, k-Lin).
- We combine batch arguments for NP (**BARGs**) and functional commitments (**FCs**).
- *Exciting open questions* - let's chat!

Thank you!

ia.cr/2024/895

david.balbas@imdea.org

Security: Unforgeability

- Security is game-based [FMNP16]. Adversary $\mathcal{A}$ and challenger interact via oracles:
  - $\mathcal{O}^{\mathsf{KeyGen}}(\mathsf{id}) \to \mathsf{pk}_{\mathsf{id}}$
  - $\mathcal{O}^{\mathsf{Sign}}(\mathsf{id}, m, \ell) \to \sigma$. Only *one query per label* is allowed!
  - $\mathcal{O}^{\mathsf{Corr}}(\mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$

Security: Unforgeability

- Security is game-based [FMNP16]. Adversary $\mathcal{A}$ and challenger interact via oracles:
  - $\mathcal{O}^{\mathsf{KeyGen}}(\mathsf{id}) \rightarrow \mathsf{pk}_{\mathsf{id}}$
  - $\mathcal{O}^{\mathsf{Sign}}(\mathsf{id}, m, \ell) \rightarrow \sigma$. Only *one query per label* is allowed!
  - $\mathcal{O}^{\mathsf{Corr}}(\mathsf{id}) \rightarrow \mathsf{sk}_{\mathsf{id}}$
- At the end, $\mathcal{A}$ outputs $\left(f^*, (\ell_1^*, \ldots, \ell_n^*), (\mathsf{vk}_1, \ldots, \mathsf{vk}_n), \boldsymbol{y}^*, \sigma_{f,y}^*\right)$ where no vk_i can be corrupted.

Security: Unforgeability

- Security is game-based [FMNP16]. Adversary $\mathcal{A}$ and challenger interact via oracles:
  - $\mathcal{O}^{\mathsf{KeyGen}}(\mathsf{id}) \rightarrow \mathsf{pk}_{\mathsf{id}}$
  - $\mathcal{O}^{\mathsf{Sign}}(\mathsf{id}, m, \ell) \rightarrow \sigma$. Only *one query per label* is allowed!
  - $\mathcal{O}^{\mathsf{Corr}}(\mathsf{id}) \rightarrow \mathsf{sk}_{\mathsf{id}}$
- At the end, $\mathcal{A}$ outputs $\left(f^*, (\ell_1^*, \ldots, \ell_n^*), (\mathsf{vk}_1, \ldots, \mathsf{vk}_n), \boldsymbol{y}^*, \sigma_{f,y}^*\right)$ where no vk_i can be corrupted.
- $\mathcal{A}$ wins if $\sigma_{f,y}^*$ *verifies* and either:
  1. Exists i such that $\mathcal{O}^{\mathsf{Sign}}(\ell_i^*, \cdot)$ was never queried.
  2. For all i, (ℓ_i^*, m_i) honest but $\boldsymbol{y}^* \neq f^*(m_1, \ldots, m_n)$.

Proving Security

- The proof proceeds by partitioning the winning condition in multiple events.
- **Interesting event:** when $\boldsymbol{y} \neq f(m_1, \ldots, m_n)$ and the (deterministic) commitment to the messages c^* *is dishonest*, $\mathsf{c}^* \neq \mathsf{FC.Com}(\mathsf{ck}, (m_1, \ldots, m_n))$.
- Strategy is to *gradually* show that *each partial c_i must be honest*. Multiple hybrids for each $i \in [n]$, where:
  1. Program the BARG crs and extract at i,
  2. Compare the extracted c_i to the honest one,
  3. Extract m_i and σ_i (a potential forgery) and certify the validity of the commitment update from c_{i-1} to c_i.
  4. "Reboot" the extraction to step $i + 1$.
- Add a *somewhere extractable commitment* to follow a sliding window approach.