Folding Schemes with Privacy Preserving Selective Verification

Joan Boyar & **Simon Erfurth**

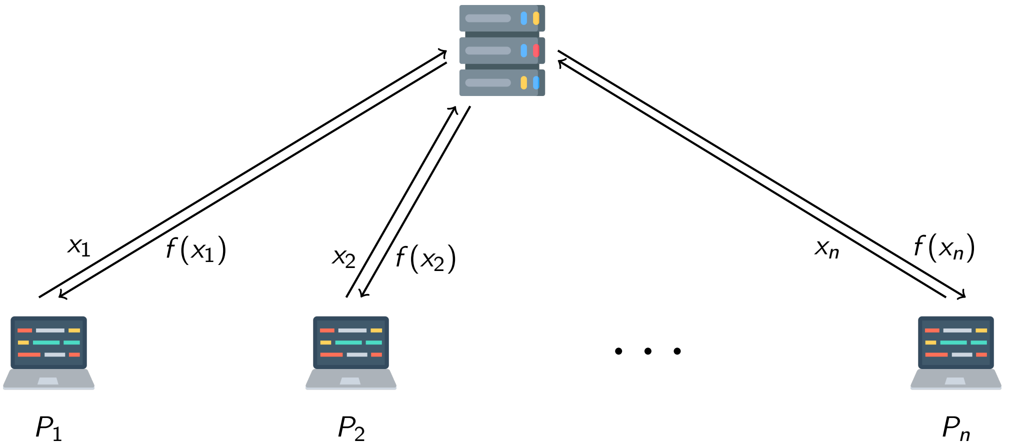
University of Southern Denmark

simon@serfurth.dk www.serfurth.dk

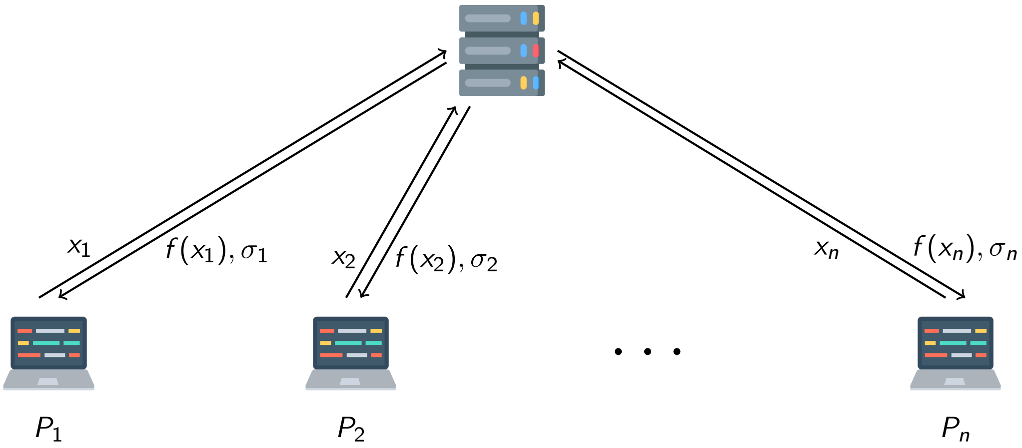
@SimonErfurth @SimonSErfurth

Motivating Example: Computation as a Service

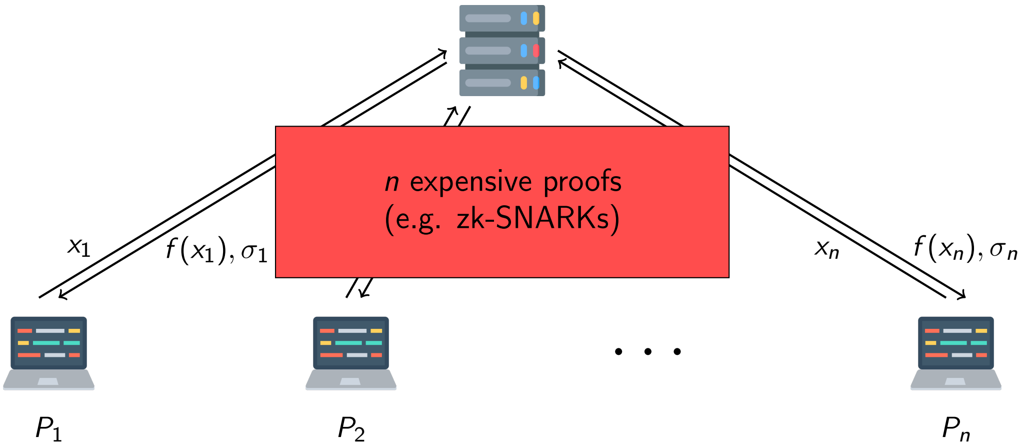
Motivating Example: Computation as a Service



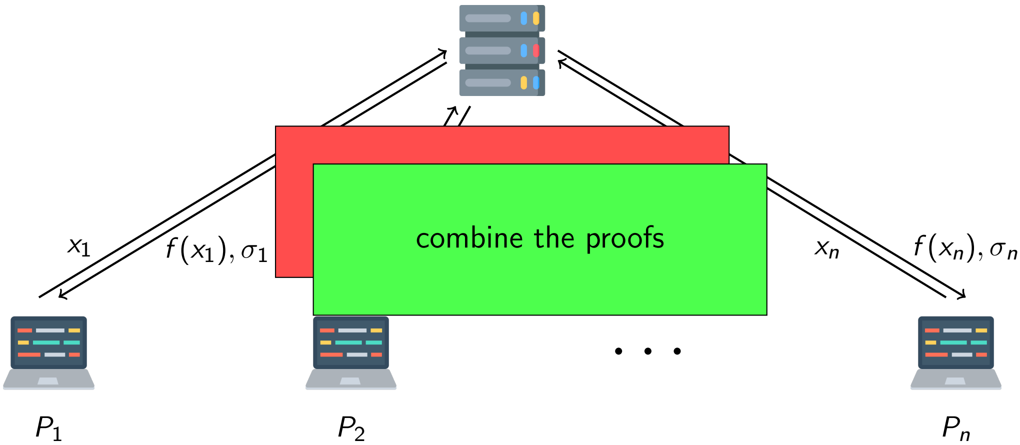
Motivating Example: Verifiable Computation as a Service



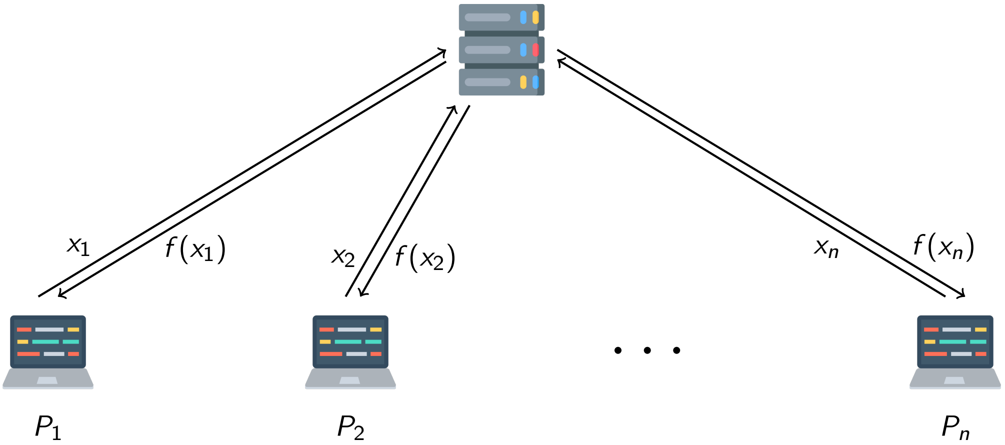
Motivating Example: Verifiable Computation as a Service



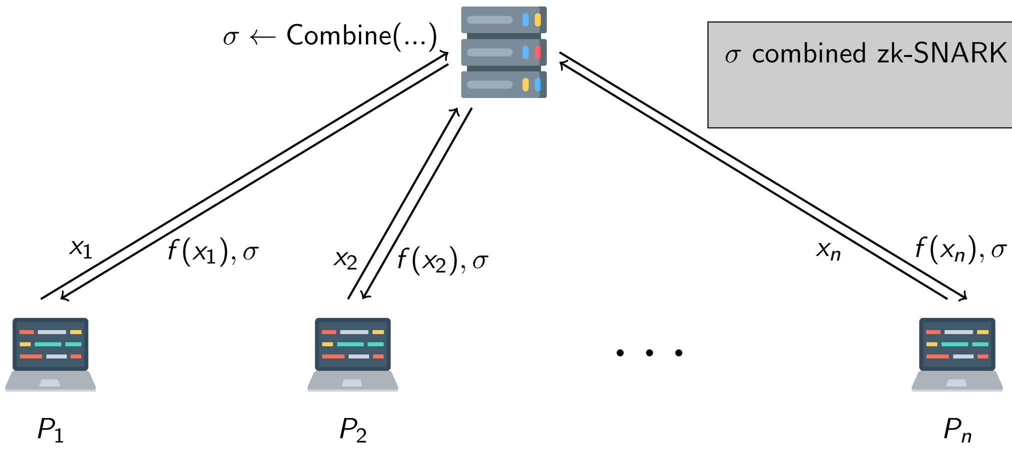
Motivating Example: Verifiable Computation as a Service



Motivating Example: Verifiable Computation as a Service

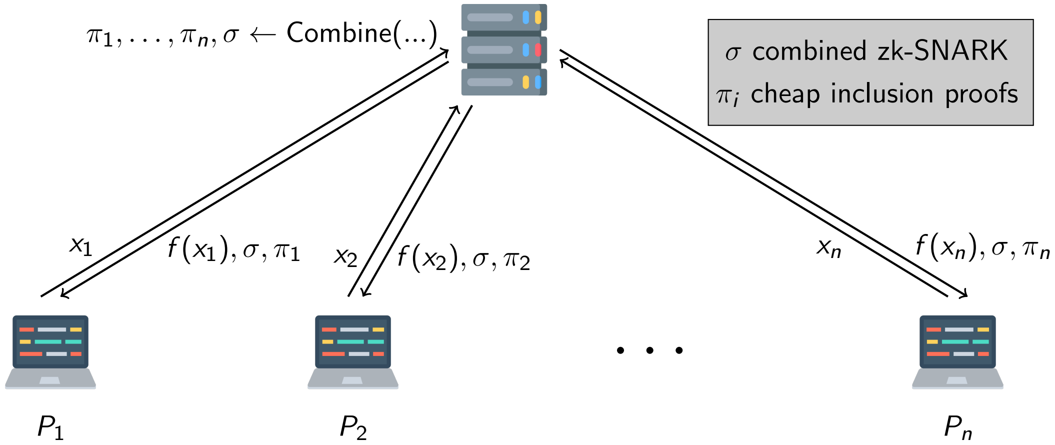


Motivating Example: Verifiable Computation as a Service



σ combined zk-SNARK

Motivating Example: Verifiable Computation as a Service



σ combined zk-SNARK
π_i cheap inclusion proofs

Folding Scheme with Selective Verification [KST22]

Folding Scheme

For NP-language $\mathcal{L}$ with relation

$$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\},$$

folding scheme FS which

Folding Scheme with Selective Verification [KST22]

Folding Scheme

For NP-language $\mathcal{L}$ with relation

$$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\},$$

folding scheme FS which

- Combines instances:
  $\text{Fold} : ((x_1, v_1), (x_2, v_2)) \rightarrow (x, v, \pi)$
  $(x, v) \in \mathcal{R} \iff (x_1, v_1), (x_2, v_2) \in \mathcal{R}$

Folding Scheme with Selective Verification [KST22]

Folding Scheme

For NP-language $\mathcal{L}$ with relation

$$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\},$$

folding scheme FS which

- Combines instances:
  $\mathsf{Fold}\colon ((x_1, v_1), (x_2, v_2)) \to (x, v, \pi)$
  $(x, v) \in \mathcal{R} \iff (x_1, v_1), (x_2, v_2) \in \mathcal{R}$
- Check statement inclusion
  $\mathsf{FoldVerify}\colon (x_1, x_2, x, \pi) \to 0/1$
  1 if π is proof that x_1 and x_2 were folded into x

Folding Scheme with Selective Verification [KST22]

Folding Scheme

For NP-language $\mathcal{L}$ with relation

$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\}$,

folding scheme FS which

- Combines instances:
  $\mathsf{Fold}\colon ((x_1, v_1), (x_2, v_2)) \to (x, v, \pi)$
  $(x, v) \in \mathcal{R} \iff (x_1, v_1), (x_2, v_2) \in \mathcal{R}$
- Check statement inclusion
  $\mathsf{FoldVerify}\colon (x_1, x_2, x, \pi) \to 0/1$
  1 if π is proof that x_1 and x_2 were folded into x

Example

For $A \in \mathbb{F}^{n \times m}$; $\mathcal{L}_A = \{x \mid \exists v\colon Av = x\}$.

Folding Scheme

For NP-language $\mathcal{L}$ with relation

$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\}$,

folding scheme FS which

- Combines instances:
  $\mathsf{Fold}\colon ((x_1, v_1), (x_2, v_2)) \to (x, v, \pi)$
  $(x, v) \in \mathcal{R} \iff (x_1, v_1), (x_2, v_2) \in \mathcal{R}$
- Check statement inclusion
  $\mathsf{FoldVerify}\colon (x_1, x_2, x, \pi) \to 0/1$
  1 if π is proof that x_1 and x_2 were folded into x

Example

For $A \in \mathbb{F}^{n\times m}$; $\mathcal{L}_A = \{x \mid \exists v\colon Av = x\}$.

- $\mathsf{Fold}((x_1, v_1), (x_2, v_2))$:
  $\rho \leftarrow_{\$} \mathbb{F}; \pi = \rho;$

$$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$

Folding Scheme with Selective Verification [KST22]

Folding Scheme

For NP-language $\mathcal{L}$ with relation
$\mathcal{R} = \{(x, v) \mid v \text{ is a proof that } x \in \mathcal{L}\}$,
folding scheme FS which

- Combines instances:
  $\mathsf{Fold}\colon ((x_1, v_1), (x_2, v_2)) \to (x, v, \pi)$
  $(x, v) \in \mathcal{R} \iff (x_1, v_1), (x_2, v_2) \in \mathcal{R}$
- Check statement inclusion
  $\mathsf{FoldVerify}\colon (x_1, x_2, x, \pi) \to 0/1$
  1 if π is proof that x_1 and x_2 were folded into x

Example

For $A \in \mathbb{F}^{n\times m}$; $\mathcal{L}_A = \{x \mid \exists v\colon Av = x\}$.

- $\mathsf{Fold}((x_1, v_1), (x_2, v_2))$:
  $\rho \leftarrow_{\$} \mathbb{F}; \pi = \rho;$
  $$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$
- $\mathsf{FoldVerify}(x_1, x_2, x, \pi)$: check that
  $$x = x_1 + \rho x_2.$$

Folding Scheme: Security

Example

Folding Scheme with Selective Verification

Folding Scheme: Security

- **Completeness**: No Adv. can output input to Fold in $\mathcal{R}$, which gives output not in $\mathcal{R}$ (or invalid folding proof).

Example

- **Completeness**: $(x_1, v_1), (x_2, v_2) \in \mathcal{R}$ then

$$\begin{aligned} Av = & A(v_1 + \rho v_2) = Av_1 + \rho A v_2 \\ = & x_1 + \rho x_2 = x \end{aligned}$$

Folding Scheme: Security

- **Completeness**: No Adv. can output input to Fold in $\mathcal{R}$, which gives output not in $\mathcal{R}$ (or invalid folding proof).
- **Knowledge Soundness**: From Adv. giving x_1, x_2, x, v, π where $(x, v) \in \mathcal{R}$ and π is accepted, we can extract witness for x_1, x_2.

Example

- **Completeness**: $(x_1, v_1), (x_2, v_2) \in \mathcal{R}$ then

$$\begin{aligned} Av =& A(v_1 + \rho v_2) = Av_1 + \rho A v_2 \\ =& x_1 + \rho x_2 = x \end{aligned}$$

- **Knowledge Soundness**: Run to get $x, v, \pi = \rho$ and $x', v', \pi' = \rho'$ for same input.

$$\begin{aligned} v =& v_1 + \rho v_2 \\ v' =& v_1 + \rho' v_2 \\ \Rightarrow v_2 =& (\rho' - \rho)^{-1}(v' - v) \end{aligned}$$

From 2-folding to 4-folding

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R} \Rightarrow$ **Bootstrapping**

(x_1, v_1) (x_2, v_2) (x_3, v_3) (x_4, v_4)

Folding Scheme with Selective Verification

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

$$\begin{tikzcd}[column sep=small]
& (x', v', \pi') & & & \\
(x_1, v_1) \arrow[ur, "\text{Fold}"] & & (x_2, v_2) \arrow[ul] & (x_3, v_3) & (x_4, v_4)
\end{tikzcd}$$

Folding Scheme with Selective Verification

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

$$\begin{tikzcd}
 & (x', v', \pi') & & & (x'', v'', \pi'') & \\
(x_1, v_1) \arrow[ur] & & (x_2, v_2) \arrow[ul, "\text{Fold}"'] & (x_3, v_3) \arrow[ur] & & (x_4, v_4) \arrow[ul, "\text{Fold}"']
\end{tikzcd}$$

Folding Scheme with Selective Verification

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

$$\begin{tikzcd}
& & & (x, v, \pi''') & & & \\
& (x', v', \pi') \arrow[urr, "\mathsf{Fold}"] & & & & (x'', v'', \pi'') \arrow[ull] & \\
(x_1, v_1) \arrow[ur, "\mathsf{Fold}"] & & (x_2, v_2) \arrow[ul] & & (x_3, v_3) \arrow[ur, "\mathsf{Fold}"] & & (x_4, v_4) \arrow[ul]
\end{tikzcd}$$

Folding Scheme with Selective Verification

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

$$\begin{tikzcd}
 & & & (x, v, \pi''') & & & \\
 & (x', v', \pi') \arrow[urr, "\text{Fold}"] & & & & (x'', v'', \pi'') \arrow[ull] & \\
(x_1, v_1) \arrow[ur, "\text{Fold}"] & & (x_2, v_2) \arrow[ul] & & (x_3, v_3) \arrow[ur, "\text{Fold}"] & & (x_4, v_4) \arrow[ul]
\end{tikzcd}$$

$$\pi = (\pi', \pi'', \pi''')$$

Folding Scheme with Selective Verification

From 2-folding to 4-folding

Output of Fold is in $\mathcal{R}$ $\Rightarrow$ **Bootstrapping**

$$\begin{tikzcd}
 & & & (x, v, \pi''') & & & \\
 & (x', v', \pi') \arrow[urr, "\mathrm{Fold}"] & & & & (x'', v'', \pi'') \arrow[ull] & \\
(x_1, v_1) \arrow[ur] & & (x_2, v_2) \arrow[ul, "\mathrm{Fold}"'] & & (x_3, v_3) \arrow[ur, "\mathrm{Fold}"] & & (x_4, v_4) \arrow[ul]
\end{tikzcd}$$

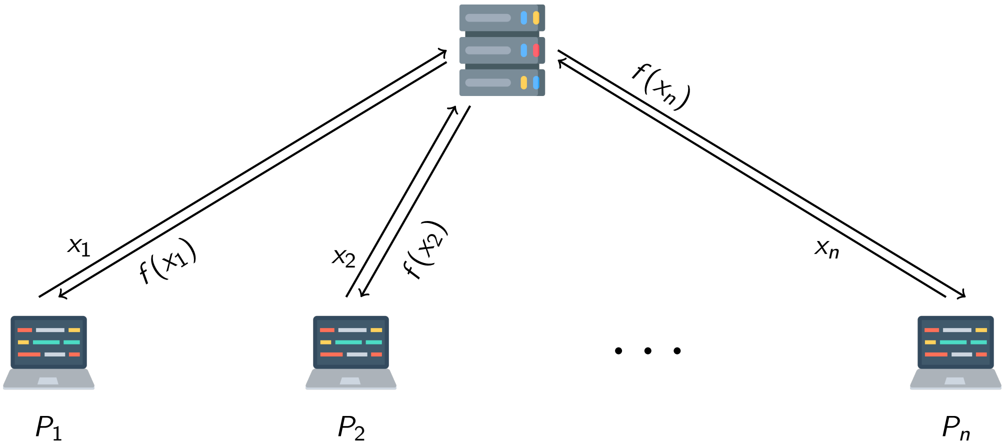
$$\pi = (\pi', \pi'', \pi''')$$

From 2-folding to n-folding

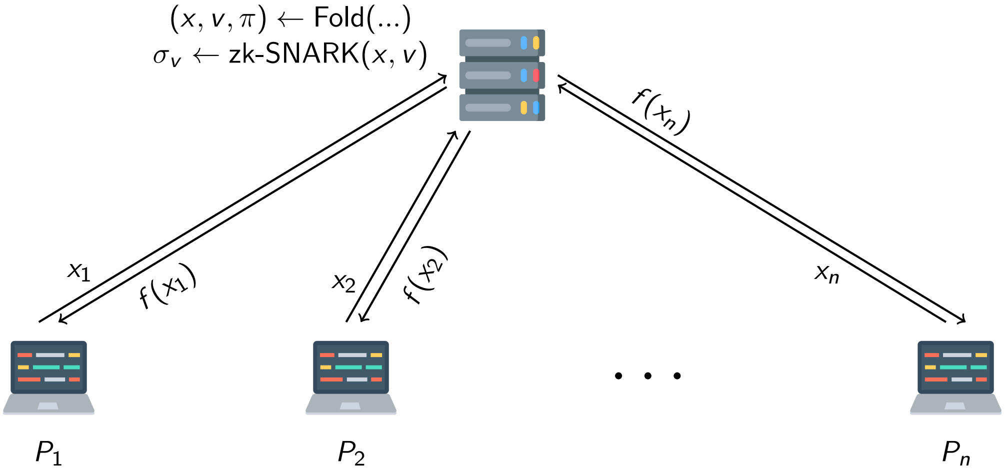
Bigger binary tree construction or:

1. $\mathrm{Fold}((x_1, v_1), (x_2, v_2)) \to (x', v', \pi')$
2. $\mathrm{Fold}((x', v'), (x_3, v_3)) \to (x'', v'', \pi'')$ and let $\pi = (\pi', \pi'')$.
3. ...
4. $\mathrm{Fold}((x^{(n-2)}, v^{(n-2)}), (x_n, v_n)) \to (x, v, \pi^{(n-1)})$; $\pi = (\pi', \pi'', \ldots, \pi^{(n-1)})$

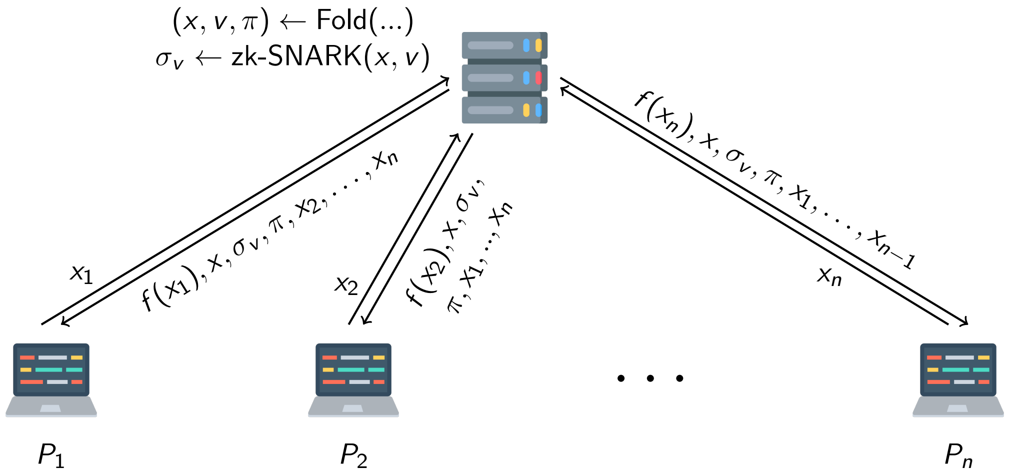
Motivating Example: Verifiable Computation as a Service



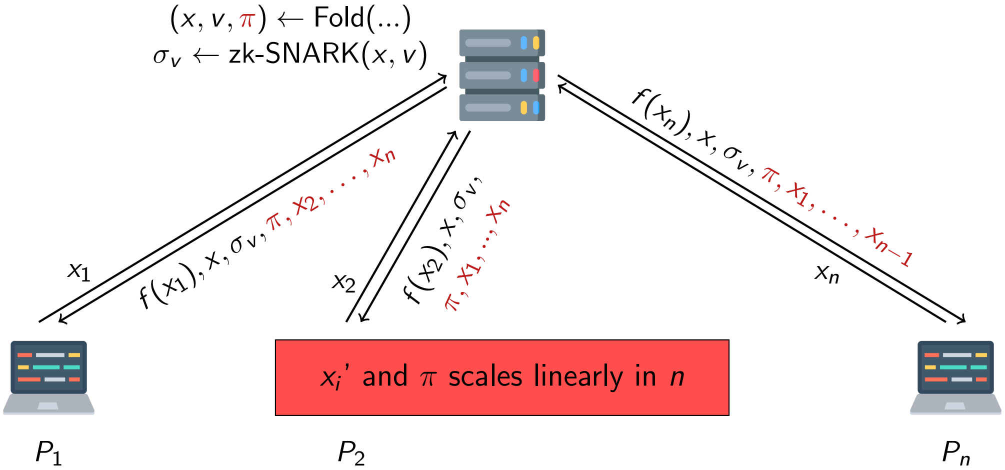
Motivating Example: Verifiable Computation as a Service



Motivating Example: Verifiable Computation as a Service



Motivating Example: Verifiable Computation as a Service



Folding Scheme **with Selective Verification** [RZ23]

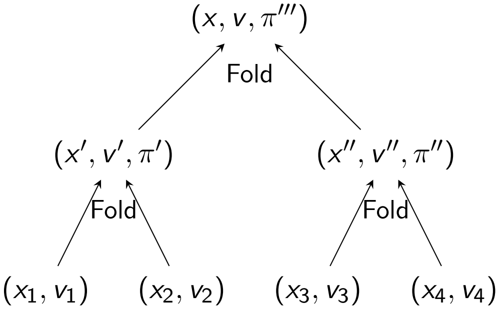
Idea

Generate n proofs π_i, each containing $O(\log n)$ folding proofs and statements.

Folding Scheme **with Selective Verification** [RZ23]

Idea

Generate n proofs π_i, each containing $O(\log n)$ folding proofs and statements.



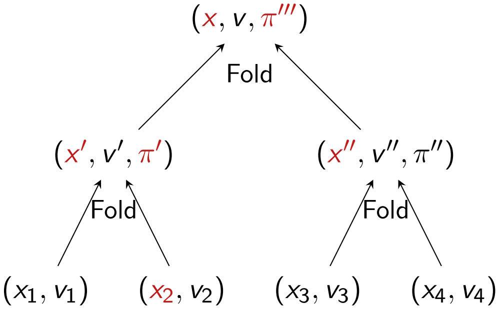
Folding Scheme **with Selective Verification** [RZ23]

Idea

Generate n proofs π_i, each containing $O(\log n)$ folding proofs and statements.

Example

- $\pi_1 = \{x_2, x', \pi', x'', x, \pi'''\}$



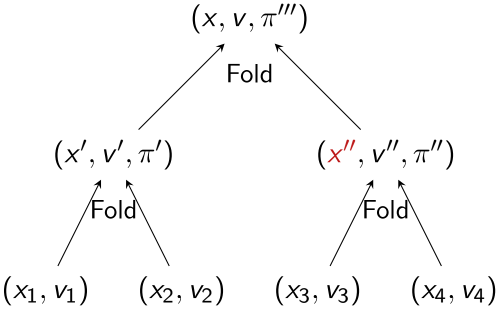
Folding Scheme **with Selective Verification** [RZ23]

Idea

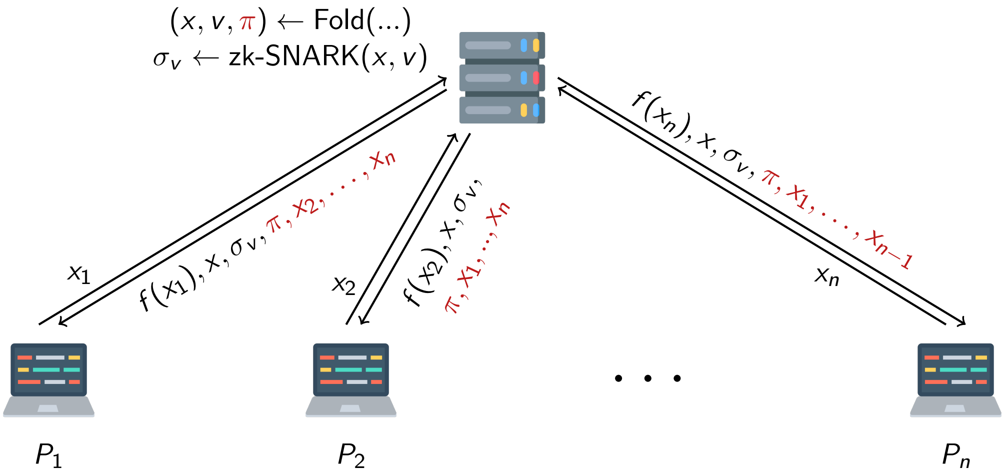
Generate n proofs π_i, each containing $O(\log n)$ folding proofs and statements.

Example

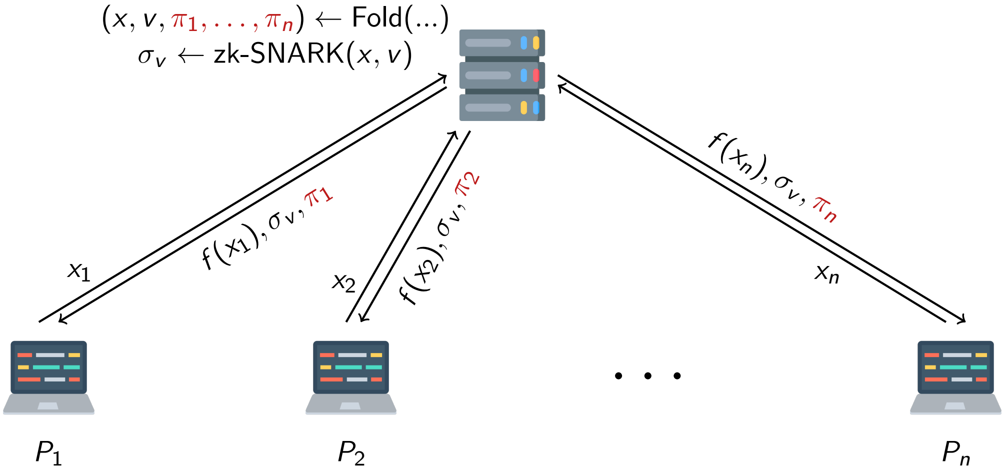
- $\pi_1 = \{x_2, x', \pi', x'', x, \pi'''\}$
- $\pi_2 = \{x_1, x', \pi', x'', x, \pi'''\}$
- $\pi_3 = \{x_4, x'', \pi'', x', x, \pi'''\}$
- $\pi_4 = \{x_3, x'', \pi'', x', x, \pi'''\}$



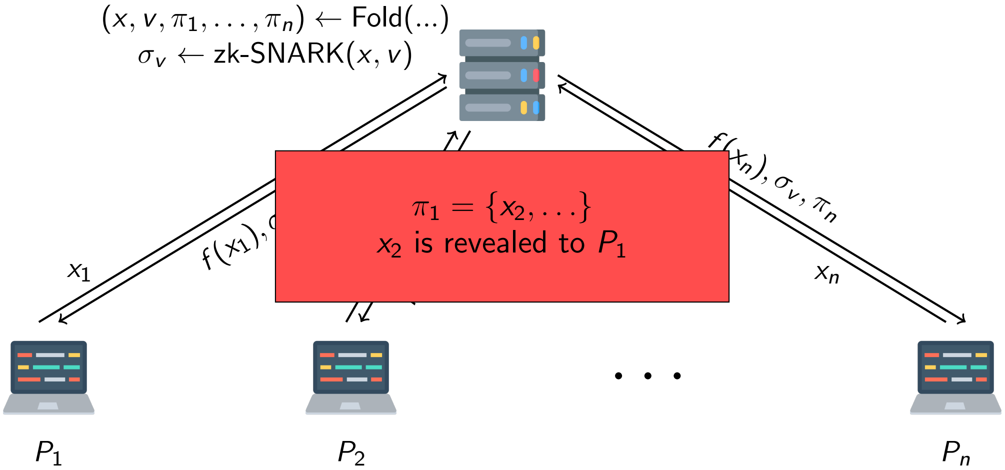
Motivating Example: Verifiable Computation as a Service



Motivating Example: Verifiable Computation as a Service



Motivating Example: Verifiable Computation as a Service



Folding Scheme with **Privacy Preserving** Selective Verification [B**E**24]

Idea

Folding scheme hiding others' statements.

Folding Scheme with **Privacy Preserving** Selective Verification [BE24]

Idea

Folding scheme hiding others' statements.

NP statement hider

Hide each instance (x, v) as another instance (x', v') and generate certificate c that x' hides x. More on these later

Folding Scheme with **Privacy Preserving** Selective Verification [BE24]

Idea

Folding scheme hiding others' statements.

NP statement hider

Hide each instance (x, v) as another instance (x', v') and generate certificate c that x' hides x. More on these later

(x_1, v_1) (x_2, v_2) (x_3, v_3) (x_4, v_4)

Folding Scheme with **Privacy Preserving** Selective Verification [BE24]

Idea

Folding scheme hiding others' statements.

NP statement hider

Hide each instance (x, v) as another instance (x', v') and generate certificate c that x' hides x. More on these later

Example

- $\pi_1 = \{x_1', c_1 \quad \}$
- $\pi_2 = \{x_2', c_2 \quad \}$
- $\pi_3 = \{x_3', c_3 \quad \}$
- $\pi_4 = \{x_4', c_4 \quad \}$

$$
\begin{tikzcd}
(x_1', v_1', c_1) & (x_2', v_2', c_2) & (x_3', v_3', c_3) & (x_4', v_4', c_4) \\
(x_1, v_1) \arrow[u, "\text{Hide}"] & (x_2, v_2) \arrow[u, "\text{Hide}"] & (x_3, v_3) \arrow[u, "\text{Hide}"] & (x_4, v_4) \arrow[u, "\text{Hide}"]
\end{tikzcd}
$$

Folding Scheme with **Privacy Preserving** Selective Verification [BE24]

Idea

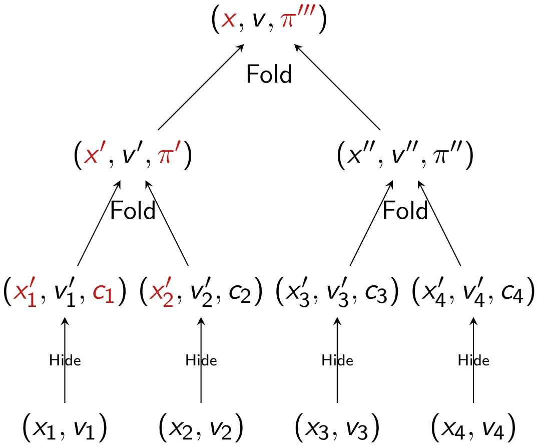
Folding scheme hiding others' statements.

NP statement hider

Hide each instance (x, v) as another instance (x', v') and generate certificate c that x' hides x. More on these later

Example

- $\pi_1 = \{x'_1, c_1, x'_2, x', \pi', x, \pi'''\}$
- $\pi_2 = \{x'_2, c_2, x'_1, x', \pi', x, \pi'''\}$
- $\pi_3 = \{x'_3, c_3, x'_4, x'', \pi'', x, \pi'''\}$
- $\pi_4 = \{x'_4, c_4, x'_3, x'', \pi'', x, \pi'''\}$



Folding Scheme with Privacy Preserving Selective Verification [BE24]

Security of Privacy Preserving FS

IND-CMA flavor:

Folding Scheme with Privacy Preserving Selective Verification [BE24]

Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j

$$(x_1, v_1) \quad (x_2{}^0, v_2^0) \quad (x_3, v_3) \quad (x_4, v_4)$$
$$(x_2{}^1, v_2^1)$$

Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j
2. Entry j chosen at random

$$(x_1, v_1) \quad \begin{matrix}(x_2{}^0, v_2^0)\\(x_2{}^1, v_2^1)\end{matrix} \quad (x_3, v_3) \quad (x_4, v_4)$$

$$b \leftarrow_{\$} \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j
2. Entry j chosen at random

$$(x_1, v_1) \quad (x_2^b, v_2^b) \quad (x_3, v_3) \quad (x_4, v_4)$$

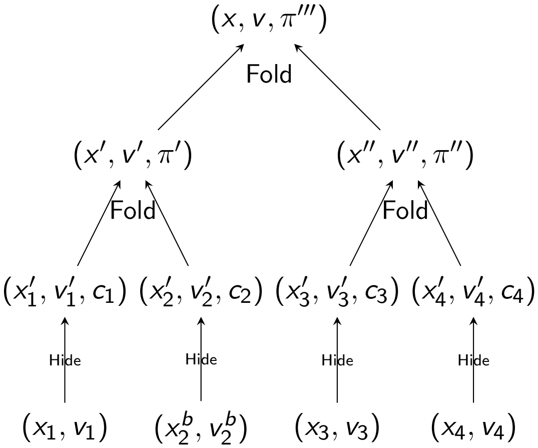
$$b \leftarrow_{\$} \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j
2. Entry j chosen at random
3. Everything is hidden and folded



$b \leftarrow_{\$} \{0, 1\}$

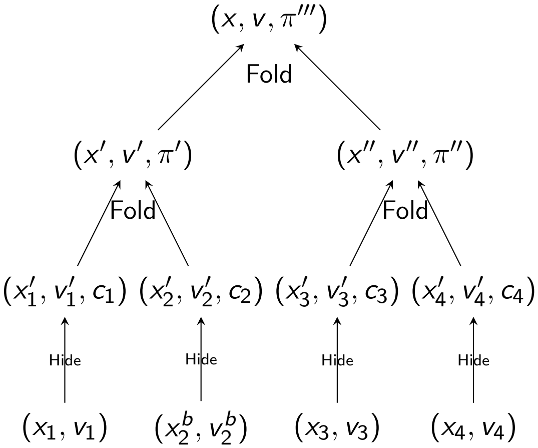
Folding Scheme with Privacy Preserving Selective Verification [BE24]

Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j
2. Entry j chosen at random
3. Everything is hidden and folded
4. Adv chooses index ℓ and receives π_ℓ

$$\mathrm{Adv} \longleftarrow \pi_1$$



$$b \leftarrow_{\$} \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [B**E**24]

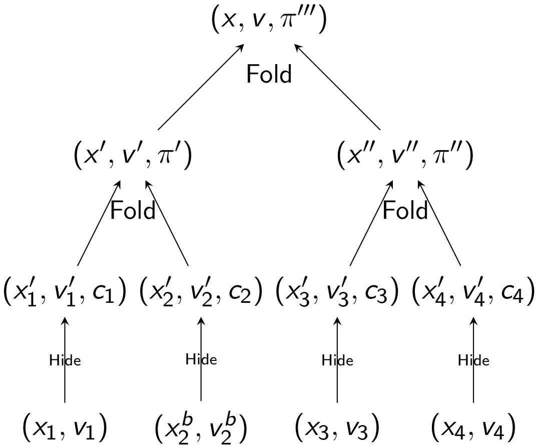
Security of Privacy Preserving FS

IND-CMA flavor:

1. Adv choose input with 2 options for entry j
2. Entry j chosen at random
3. Everything is hidden and folded
4. Adv chooses index ℓ and receives π_ℓ
5. Guess which (x_j, v_j) was used

$$b' \longleftarrow \text{Adv} \longleftarrow \pi_1$$

Win if $b' = b$



$b \leftarrow_\$ \{0, 1\}$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

Folding Scheme with Privacy Preserving Selective Verification [B**E**24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_0, v_0)$$
$$(x_1, v_1)$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_0, v_0)$$
$$(x_1, v_1)$$
$$b \leftarrow_\$ \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_b, v_b)$$

$$b \leftarrow_{\$} \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_b, v_b) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

$$b \leftarrow_{\$} \{0, 1\}$$

Folding Scheme with Privacy Preserving Selective Verification [B**E**24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_b, v_b) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

$$b \leftarrow_{\$} \{0, 1\}$$

$$(x', v') \to \text{Adv}$$

Folding Scheme with Privacy Preserving Selective Verification [BE24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_b, v_b) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

$$b \leftarrow_{\$} \{0, 1\}$$

$$(x', v') \rightarrow \text{Adv} \longrightarrow b'$$

Folding Scheme with Privacy Preserving Selective Verification [B**E**24]

NP statement hider

$$(x, v) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

- Completeness
- Knowledge Soundness
- Hiding

$$(x_b, v_b) \longrightarrow \text{Hide} \longrightarrow (x', v', c)$$

$$b \leftarrow_{\$} \{0, 1\}$$

$$(x', v') \rightarrow \text{Adv} \longrightarrow b'$$

Claim

Composing a Folding Scheme with an NP statement hider gives a Folding Scheme with Privacy Preserving Selective Verification.

NP statement hider: Example

Folding with random instance

To hide (x, v):

1. Generate random instance (x_r, v_r).
2. Fold:
$$(x', v', \pi) \leftarrow \mathrm{Fold}((x, v), (x_r, v_r))$$
3. Output $(x', v', c = (\pi, x_r))$.

NP statement hider: Example

Folding with random instance

To hide (x, v):

1. Generate random instance (x_r, v_r).
2. Fold:
$$(x', v', \pi) \leftarrow \mathrm{Fold}((x, v), (x_r, v_r))$$
3. Output $(x', v', c = (\pi, x_r))$.

Recall

$$\mathcal{L}_A = \{x \mid \exists v \colon Av = x\}$$

$\mathrm{Fold}((x_1, v_1), (x_2, v_2))$: $\rho \leftarrow_{\$} \mathbb{F}; \pi = \rho;$

$$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$

Example

NP statement hider: Example

Folding with random instance

To hide (x, v):

1. Generate random instance (x_r, v_r).
2. Fold:
$$(x', v', \pi) \leftarrow \mathsf{Fold}((x, v), (x_r, v_r))$$
3. Output $(x', v', c = (\pi, x_r))$.

Recall

$$\mathcal{L}_A = \{x \mid \exists v \colon Av = x\}$$

$\mathsf{Fold}((x_1, v_1), (x_2, v_2))$: $\rho \leftarrow_{\$} \mathbb{F}; \pi = \rho;$

$$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$

Example

1. Generate random instance in $\mathcal{R}$ as
$$v_r \leftarrow_{\$} \mathbb{F}^m; \qquad x_r = Av.$$

NP statement hider: Example

Folding with random instance

To hide (x, v):

1. Generate random instance (x_r, v_r).
2. Fold:
$$(x', v', \pi) \leftarrow \mathsf{Fold}((x, v), (x_r, v_r))$$
3. Output $(x', v', c = (\pi, x_r))$.

Recall

$$\mathcal{L}_A = \{x \mid \exists v \colon Av = x\}$$

$\mathsf{Fold}((x_1, v_1), (x_2, v_2))$: $\rho \leftarrow_\$ \mathbb{F}; \pi = \rho;$

$$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$

Example

1. Generate random instance in $\mathcal{R}$ as
$$v_r \leftarrow_\$ \mathbb{F}^m; \qquad x_r = Av.$$
2. Hide by folding
$\mathsf{Hide}((x, v), (x_r, v_r))$:
$$\rho \leftarrow_\$ \mathbb{F}$$
$$x' = x_1 + \rho x_r$$
$$v' = v_1 + \rho v_r.$$

NP statement hider: Example

Folding with random instance

To hide (x, v):

1. Generate random instance (x_r, v_r).
2. Fold:
$$(x', v', \pi) \leftarrow \mathsf{Fold}((x, v), (x_r, v_r))$$
3. Output $(x', v', c = (\pi, x_r))$.

Recall

$$\mathcal{L}_A = \{x \mid \exists v \colon Av = x\}$$

$\mathsf{Fold}((x_1, v_1), (x_2, v_2))$: $\rho \leftarrow_\$ \mathbb{F}; \pi = \rho;$
$$x = x_1 + \rho x_2; \qquad v = v_1 + \rho v_2.$$

Example

1. Generate random instance in $\mathcal{R}$ as
$$v_r \leftarrow_\$ \mathbb{F}^m; \qquad x_r = Av.$$
2. Hide by folding
$\mathsf{Hide}((x, v), (x_r, v_r))$:
$$\rho \leftarrow_\$ \mathbb{F}$$
$$x' = x_1 + \rho x_r$$
$$v' = v_1 + \rho v_r.$$
3. Output $(x', v', c = (\rho, x_r))$.

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

(x, v) is equally likely to hide (x_2, v_2) if there is $(x'_r, v'_r) \in \mathcal{R}_A$ such that:

$$x_2 + \rho' x'_r = x_1 + \rho x_r$$
$$v_2 + \rho' v'_r = v_1 + \rho v_r.$$

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

(x, v) is equally likely to hide (x_2, v_2) if there is $(x'_r, v'_r) \in \mathcal{R}_A$ such that:

$$x_2 + \rho' x'_r = x_1 + \rho x_r$$
$$v_2 + \rho' v'_r = v_1 + \rho v_r.$$

So we must have

$$x'_r = (\rho')^{-1}(x_1 + \rho x_r - x_2)$$
$$v'_r = (\rho')^{-1}(v_1 + \rho v_r - v_2)$$

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

(x, v) is equally likely to hide (x_2, v_2) if there is $(x'_r, v'_r) \in \mathcal{R}_A$ such that:

$$x_2 + \rho' x'_r = x_1 + \rho x_r$$
$$v_2 + \rho' v'_r = v_1 + \rho v_r.$$

So we must have

$$x'_r = (\rho')^{-1}(x_1 + \rho x_r - x_2)$$
$$v'_r = (\rho')^{-1}(v_1 + \rho v_r - v_2)$$

But is this in $\mathcal{R}_A$?

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

(x, v) is equally likely to hide (x_2, v_2) if there is $(x'_r, v'_r) \in \mathcal{R}_A$ such that:

$$x_2 + \rho' x'_r = x_1 + \rho x_r$$
$$v_2 + \rho' v'_r = v_1 + \rho v_r.$$

So we must have

$$x'_r = (\rho')^{-1}(x_1 + \rho x_r - x_2)$$
$$v'_r = (\rho')^{-1}(v_1 + \rho v_r - v_2)$$

But is this in $\mathcal{R}_A$?

$$\begin{aligned} A v'_r &= A(\rho')^{-1}(v_1 + \rho v_r - v_2) \\ &= (\rho')^{-1}(Av_1 + \rho A v_r - Av_2) \\ &= (\rho')^{-1}(x_1 \rho x_r - x_2) \\ &= x'_r \end{aligned}$$

NP statement hider: Example

Example is secure

Can Adv distinguish if (x, v) hides (x_1, v_1) or (x_2, v_2)?

Assume (x, v) hides (x_1, v_1) using (x_r, v_r):

$$x = x_1 + \rho x_r$$
$$v = v_1 + \rho v_r.$$

Theorem

There is a folding scheme with privacy preserving selective verification for $\mathcal{L}_A = \mathrm{Im}(A)$.

(x, v) is equally likely to hide (x_2, v_2) if there is $(x'_r, v'_r) \in \mathcal{R}_A$ such that:

$$x_2 + \rho' x'_r = x_1 + \rho x_r$$
$$v_2 + \rho' v'_r = v_1 + \rho v_r.$$

So we must have

$$x'_r = (\rho')^{-1}(x_1 + \rho x_r - x_2)$$
$$v'_r = (\rho')^{-1}(v_1 + \rho v_r - v_2)$$

But is this in $\mathcal{R}_A$?

$$\begin{aligned} Av'_r &= A(\rho')^{-1}(v_1 + \rho v_r - v_2) \\ &= (\rho')^{-1}(Av_1 + \rho A v_r - Av_2) \\ &= (\rho')^{-1}(x_1 \rho x_r - x_2) \\ &= x'_r \end{aligned}$$

Conclusion

NP statement hider

If there is a folding scheme for $\mathcal{L}$, $\mathcal{R}$ supports efficient random sampling, and for any three instances $(x_1, v_1), (x_2, v_2), (x, v) \in \mathcal{R}$ there are equally many ways to fold (x_1, v_1) into (x, v) as there is to fold (x_2, v_2) into (x, v), then there is an NP statement hider for $\mathcal{L}$.

Conclusion

NP statement hider

If there is a folding scheme for $\mathcal{L}$, $\mathcal{R}$ supports efficient random sampling, and for any three instances $(x_1, v_1), (x_2, v_2), (x, v) \in \mathcal{R}$ there are equally many ways to fold (x_1, v_1) into (x, v) as there is to fold (x_2, v_2) into (x, v), then there is an NP statement hider for $\mathcal{L}$.

Privacy Preserving Folding Scheme

As above: There is a Privacy Preserving Folding Scheme for $\mathcal{L}$.

Conclusion

NP statement hider

If there is a folding scheme for $\mathcal{L}$, $\mathcal{R}$ supports efficient random sampling, and for any three instances $(x_1, v_1), (x_2, v_2), (x, v) \in \mathcal{R}$ there are equally many ways to fold (x_1, v_1) into (x, v) as there is to fold (x_2, v_2) into (x, v), then there is an NP statement hider for $\mathcal{L}$.

Privacy Preserving Folding Scheme

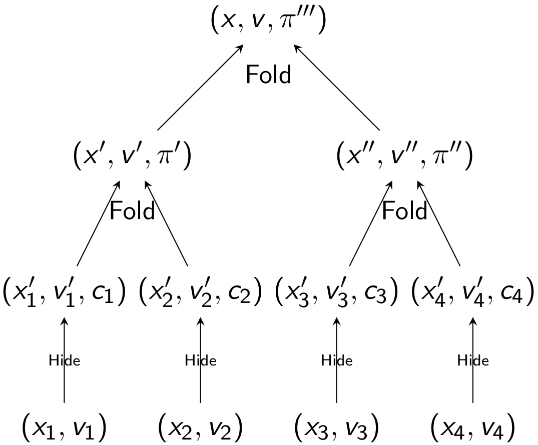
As above: There is a Privacy Preserving Folding Scheme for $\mathcal{L}$.

Languages

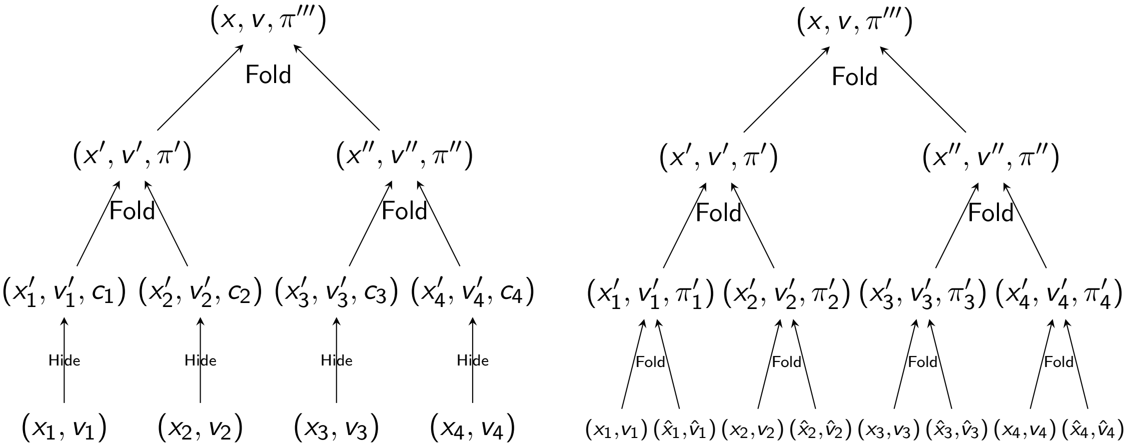
We show that this is satisfied by folding schemes schemes for

- Inner Product Relation of Committed Values [RZ23]
- Committed Relaxed R1CS [KST22]

Folding Scheme with Privacy Preserving Selective Verification [BE24]



Folding Scheme with Privacy Preserving Selective Verification [BE24]



Thank you for listening.

ePrint 2024/1530

←

[BE24]

References

[BE24] Joan Boyar and Simon Erfurth. *Folding Schemes with Privacy Preserving Selective Verification*. Cryptology ePrint Archive, Paper 2024/1530. 2024. URL: `https://eprint.iacr.org/2024/1530`.

[KST22] Abhiram Kothapalli, Srinath T. V. Setty, and Ioanna Tzialla. "Nova: Recursive Zero-Knowledge Arguments from Folding Schemes". In: *Advances in Cryptology - CRYPTO 2022*. Vol. 13510. Lecture Notes in Computer Science. Springer, 2022, pp. 359–388. DOI: 10.1007/978-3-031-15985-5_13.

[RZ23] Carla Ràfols and Alexandros Zacharakis. "Folding Schemes with Selective Verification". In: *Progress in Cryptology - LATINCRYPT 2023*. Vol. 14168. Lecture Notes in Computer Science. Springer, 2023, pp. 229–248. DOI: 10.1007/978-3-031-44469-2_12.

Made using icons from `flaticon.com`.