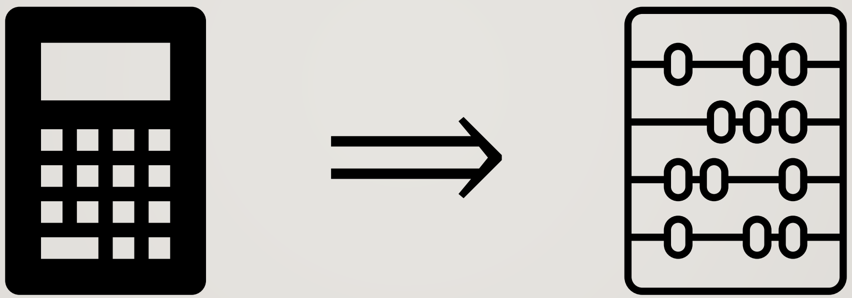
Robust Additive Randomized Encodings From IO And Pseudo-non-linear-codes

NIR BITANSKY   SAPIR FREIZEIT

Basic Paradigm In Secure Computation

[Yao82, BMR90, IK00, ...]

Reduce general secure computation to secure computation of simple functions.



Additive Randomized Encodings (ARE)

[Halevi - Ishai - Kushilevitz - Rabin 23]

Additive Randomized Encodings (ARE)

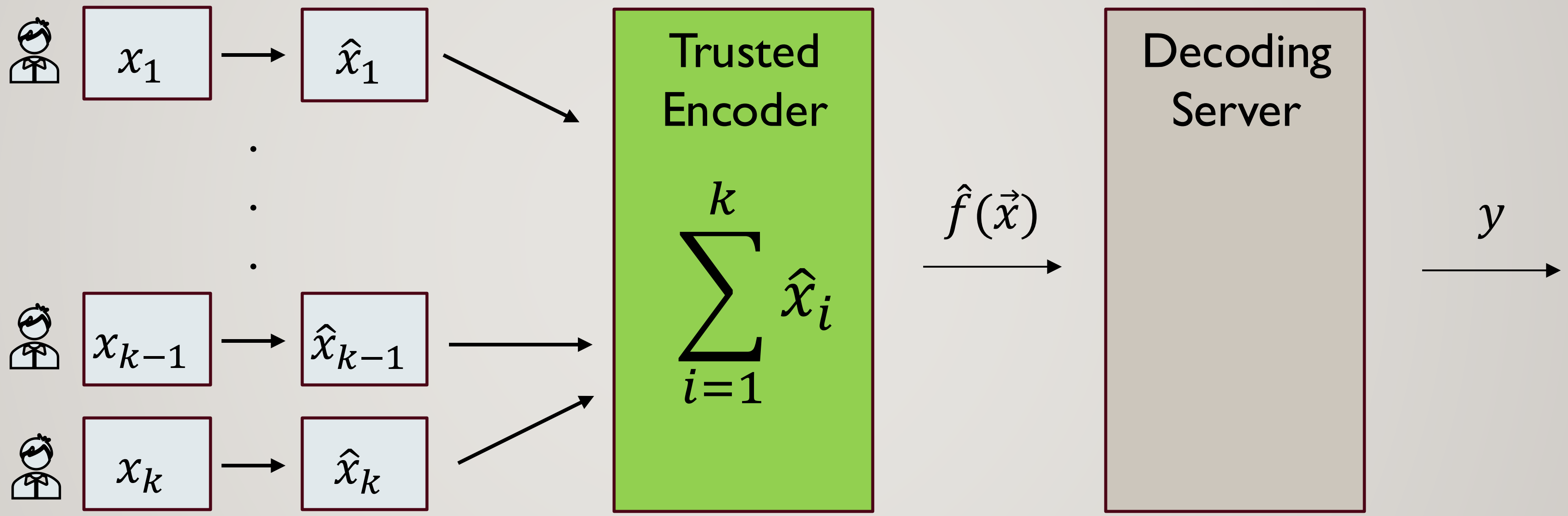
[Halevi - Ishai - Kushilevitz - Rabin 23]

$f:(x_1,\ldots,x_k)\rightarrow\{0,1\}^*, pp$ =public parameters

Additive Randomized Encodings (ARE)

[Halevi - Ishai - Kushilevitz - Rabin 23]

$f : (x_1, \ldots, x_k) \to \{0,1\}^*$, pp =public parameters

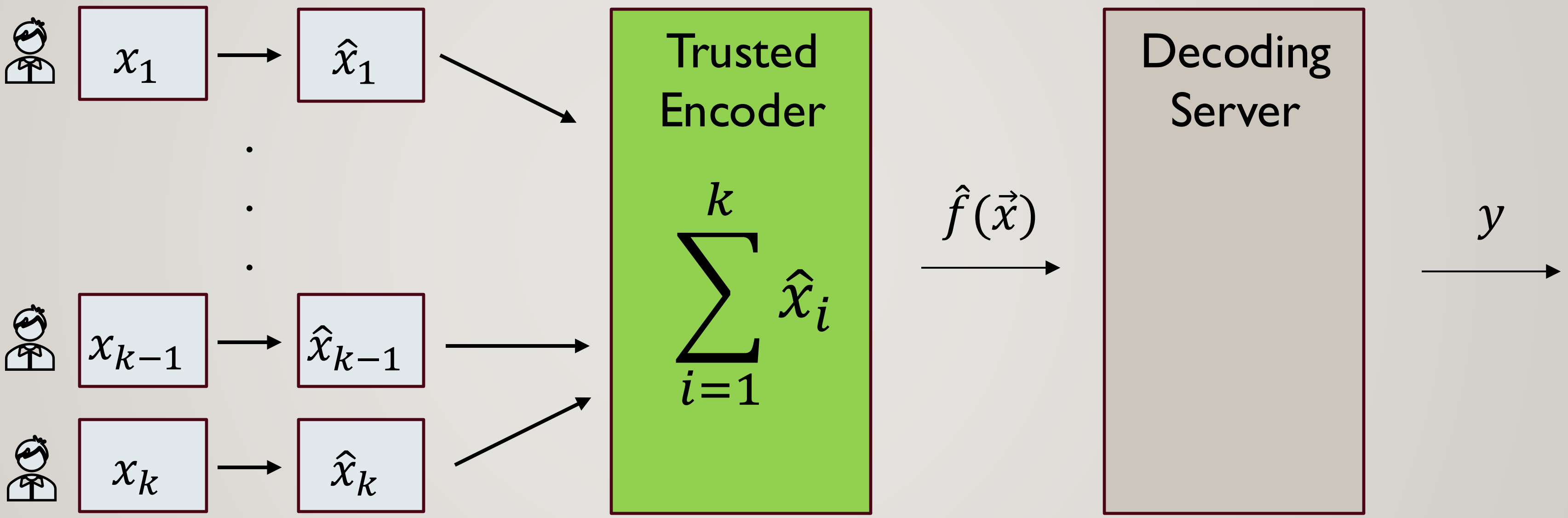


* The summation is over some abelian group

Additive Randomized Encodings (ARE)

[Halevi - Ishai - Kushilevitz - Rabin 23]

$f:(x_1,\ldots,x_k)\to\{0,1\}^*, pp$ =public parameters



Correctness
$y=f(\vec{x})$

Security
Decoder learns nothing, but $f(\vec{x})$

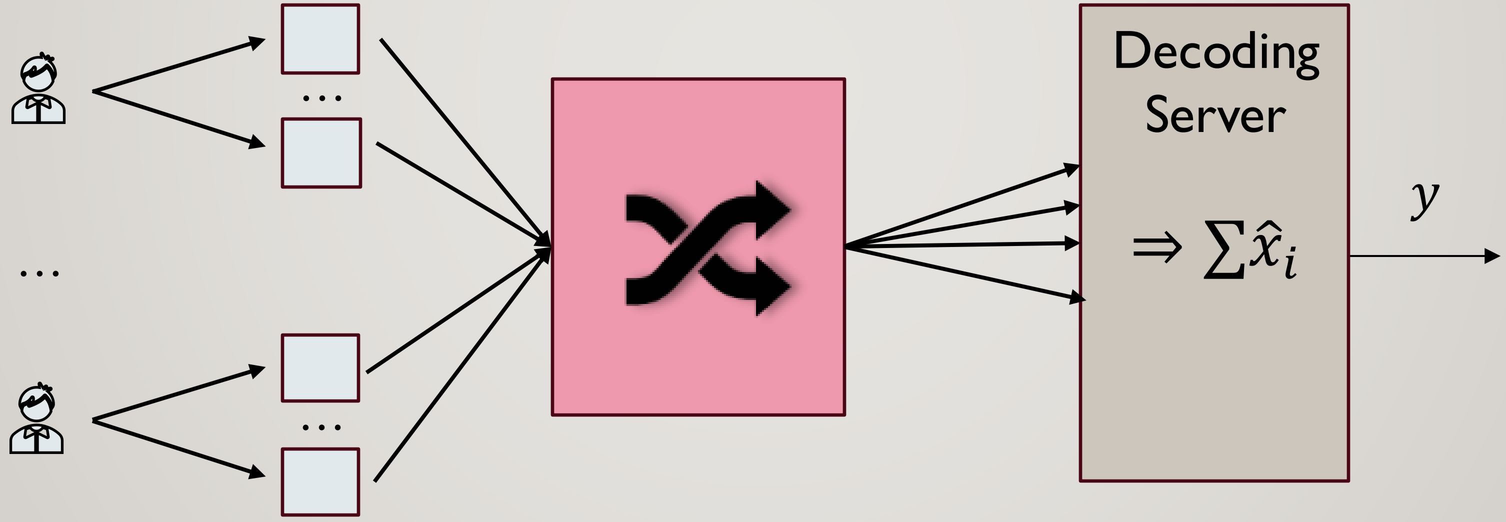
*The summation is over some abelian group

Addition Is Simple

- Simplest global encoder: **addition**

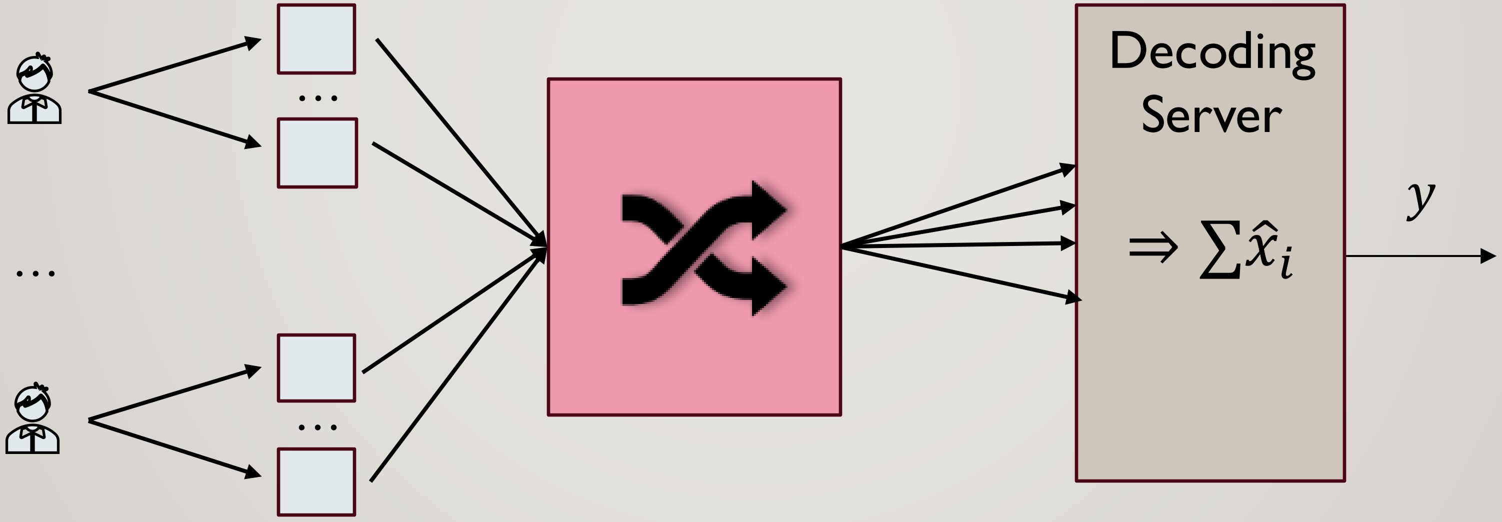
Addition Is Simple

- Simplest global encoder: **addition**
- Can be realized in the **shuffle model** (e.g., using annonymous communication) [Ishai Kushilevitz Ostrovsky Sahai 06]



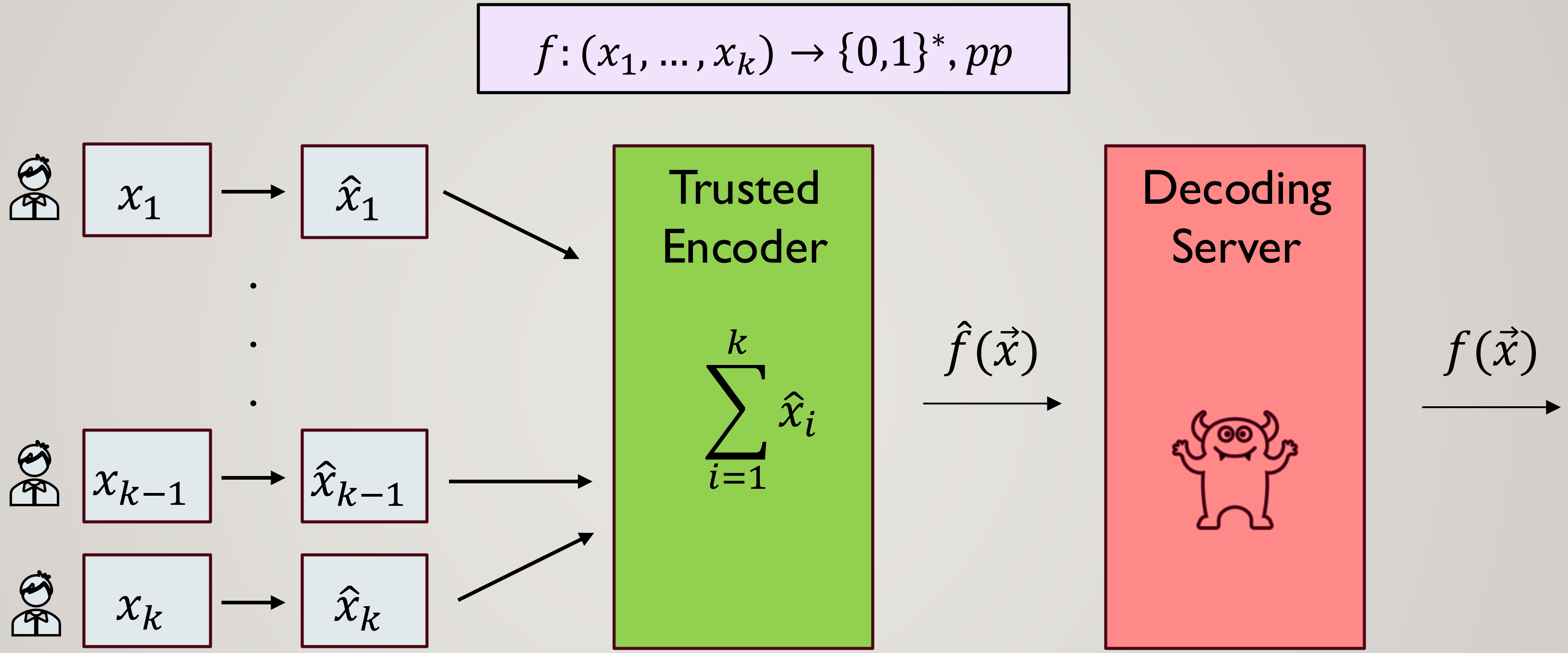
Addition Is Simple

- Simplest global encoder: **addition**
- Can be realized in the **shuffle model** (e.g., using annonymous communication) [Ishai Kushilevitz Ostrovsky Sahai 06]



- Yields NI-MPC (in the shuffle model) w/o correlated-randomness, nor public-key-infrastructure.

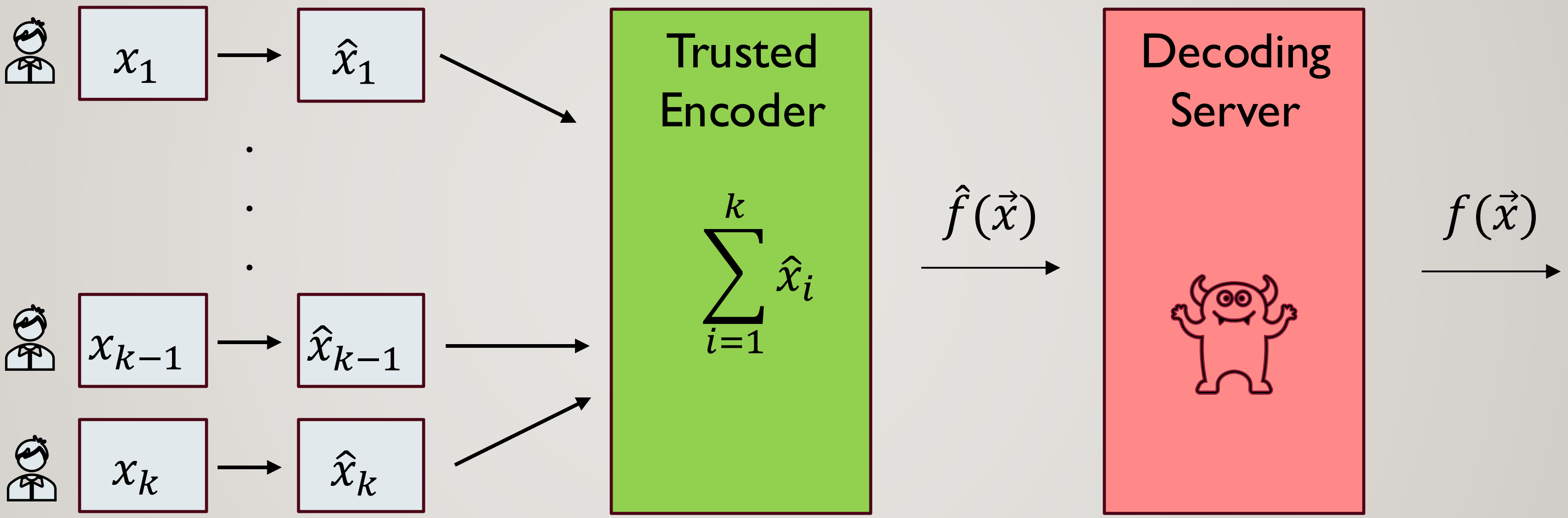
ARE Security



- Security against corrupted decoder (can't learn additional info about parties' inputs).

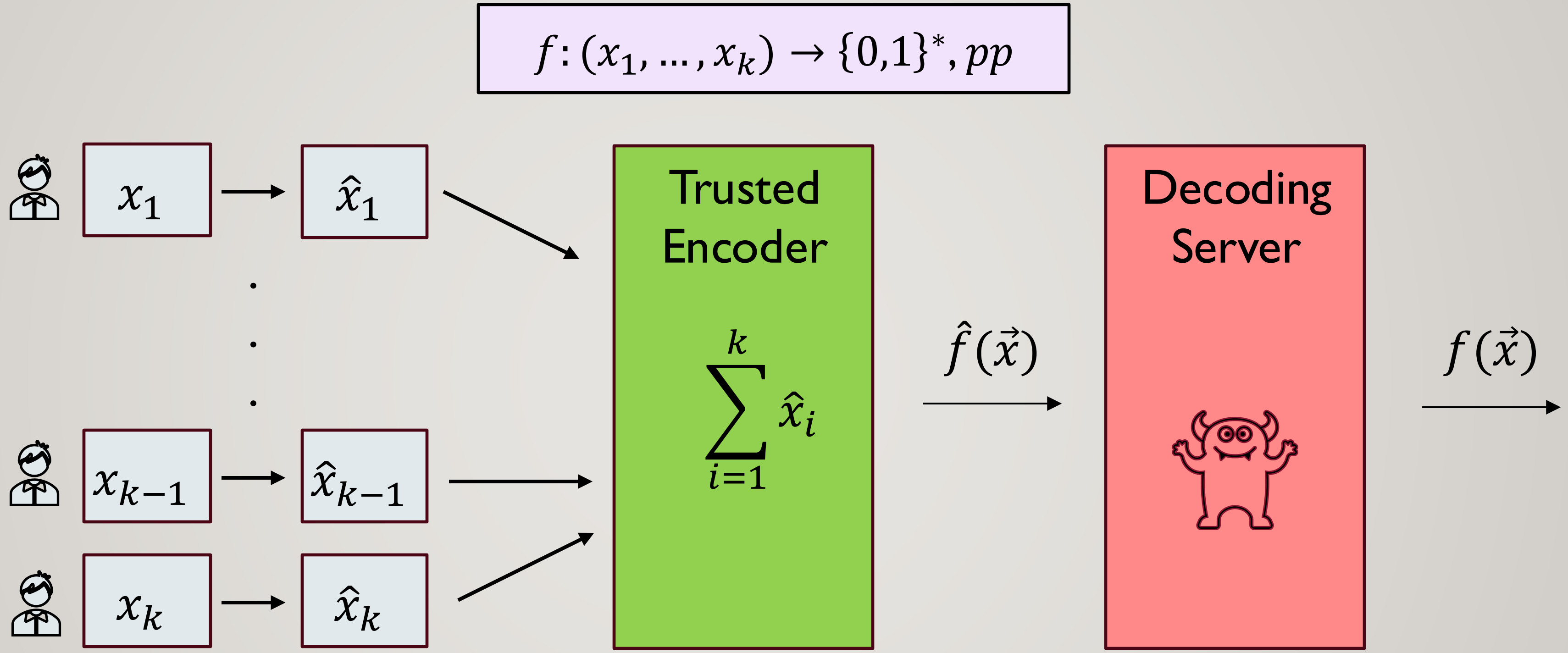
ARE Security

$$f : (x_1, \dots, x_k) \rightarrow \{0,1\}^*, pp$$



- Security against corrupted decoder (can't learn additional info about parties' inputs).
- Intuitively: decoder's view $\left(\hat{f}(\vec{x}), pp\right)$ can be recovered from $f(\vec{x})$.

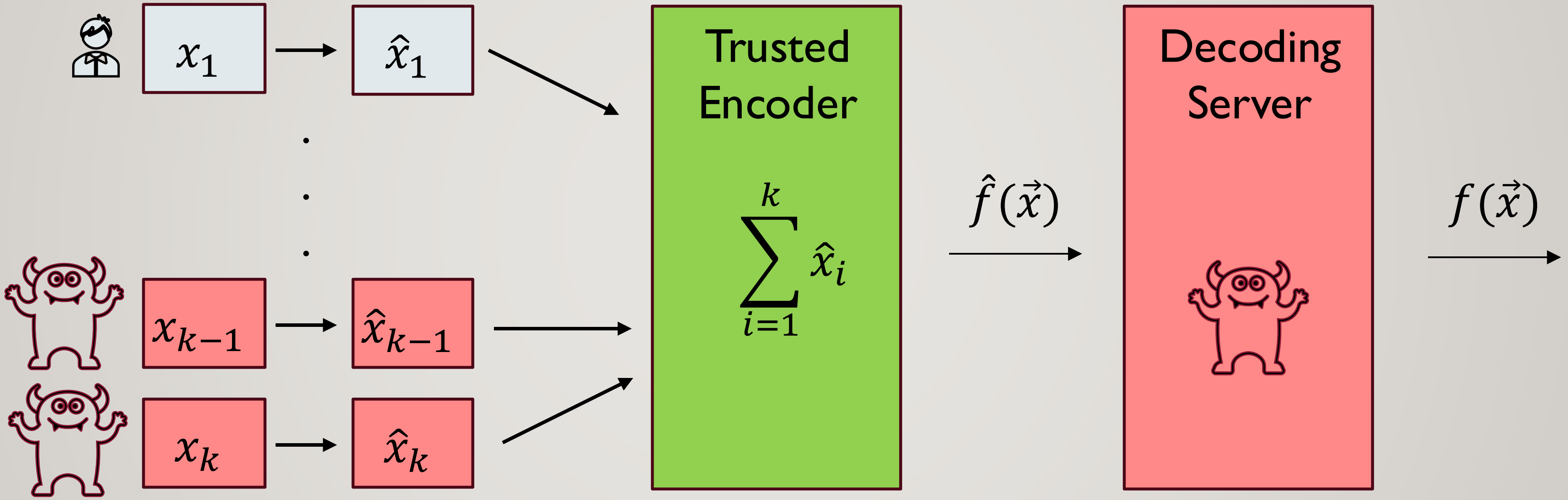
ARE Security



- Security against corrupted decoder (can't learn additional info about parties' inputs).
- Intuitively: decoder's view $\left(\hat{f}(\vec{x}), pp\right)$ can be recovered from $f(\vec{x})$.
- Simulation: $Sim(f(\vec{x})) \approx \left(\hat{f}(\vec{x}), pp\right)$ (perfect / statistical / computational).

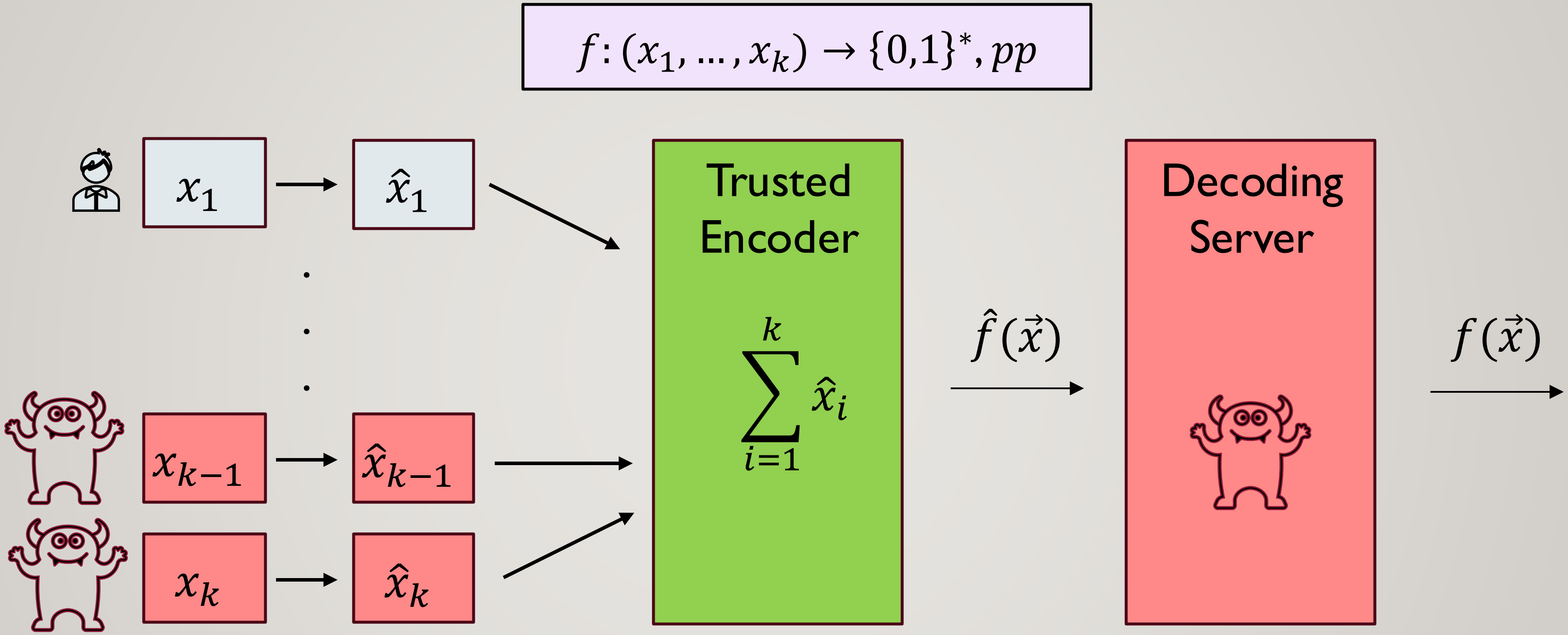
Our Focus: **Robust** ARE (RARE) [HIKR23]

$f: (x_1, \dots, x_k) \rightarrow \{0,1\}^*, pp$



- Security against **corrupted parties** C who collude with the corrupted server .

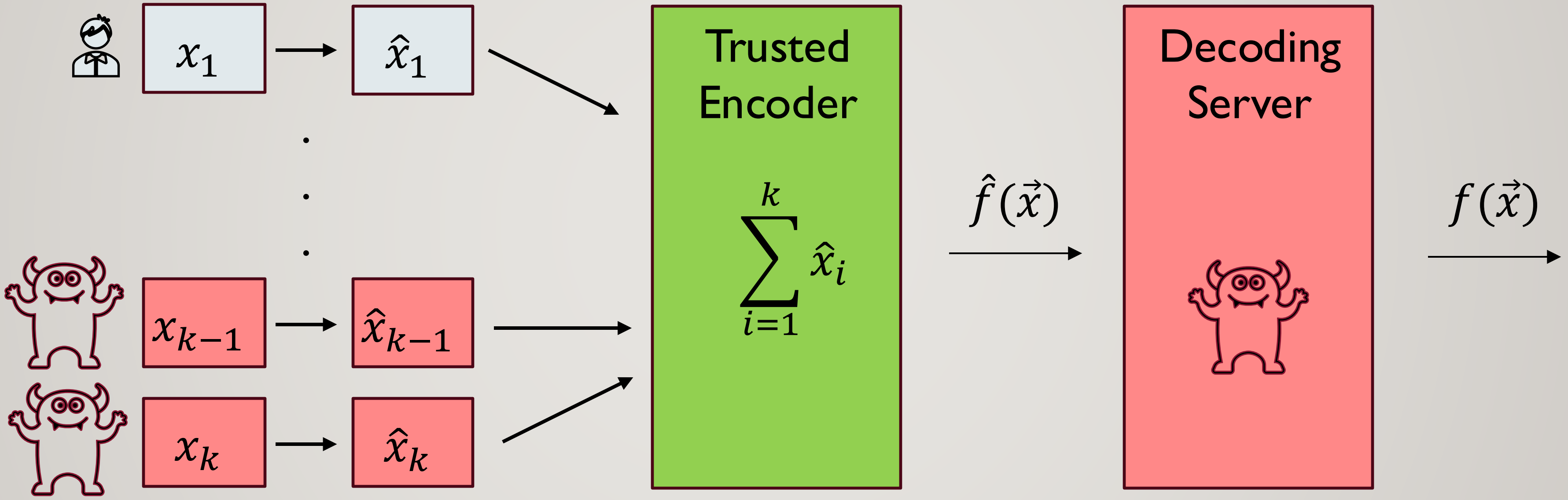
Our Focus: **Robust** ARE (RARE) [HIKR23]



- Security against **corrupted parties** C who collude with the corrupted server .
- Inevitable attack: **residual function** of honest parties H: $f_{x_H}(x_C) \coloneqq f(x_H, x_C)$.

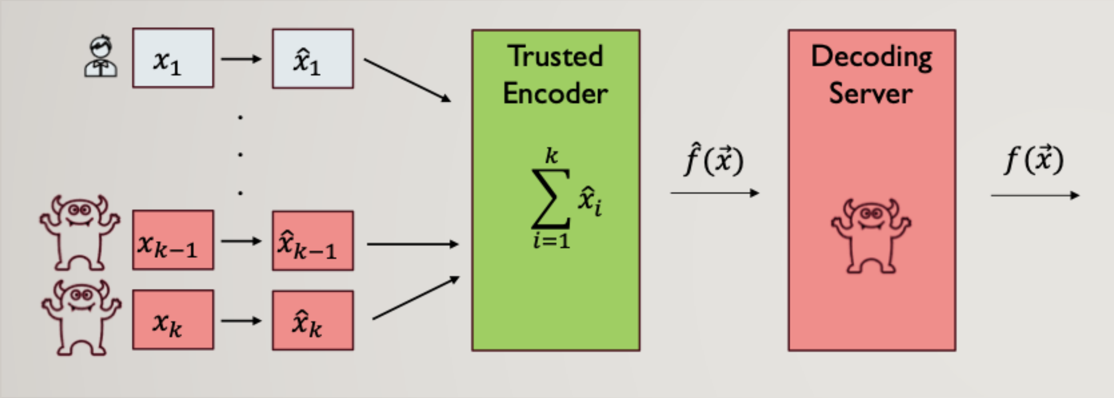
Our Focus: **Robust** ARE (RARE) [HIKR23]

$$f : (x_1, \ldots, x_k) \to \{0,1\}^*, pp$$



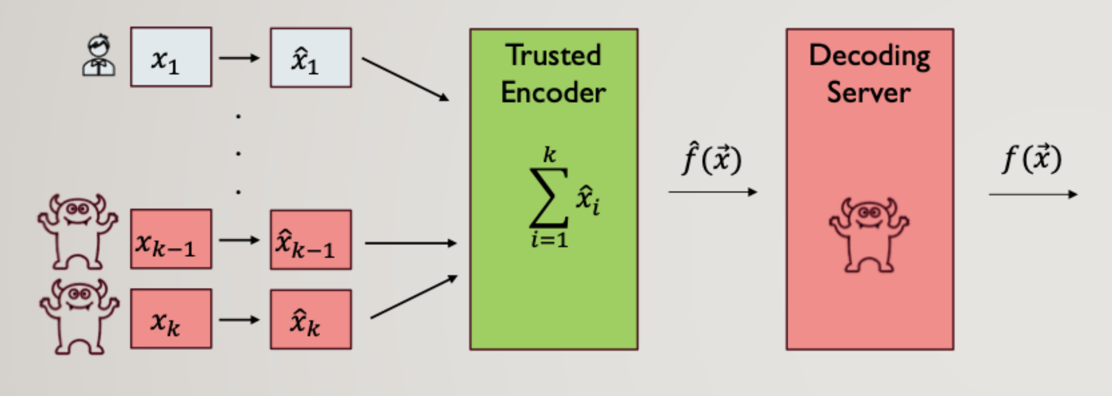
- Security against **corrupted parties** C who collude with the corrupted server .
- Inevitable attack: **residual function** of honest parties H: $f_{x_H}(x_C) := f(x_H, x_C)$.
- VBB simulation security, $Sim^{f_{x_H}}(f(x)) \approx (\hat{f}(x), pp)$

RARE implies Obfuscation



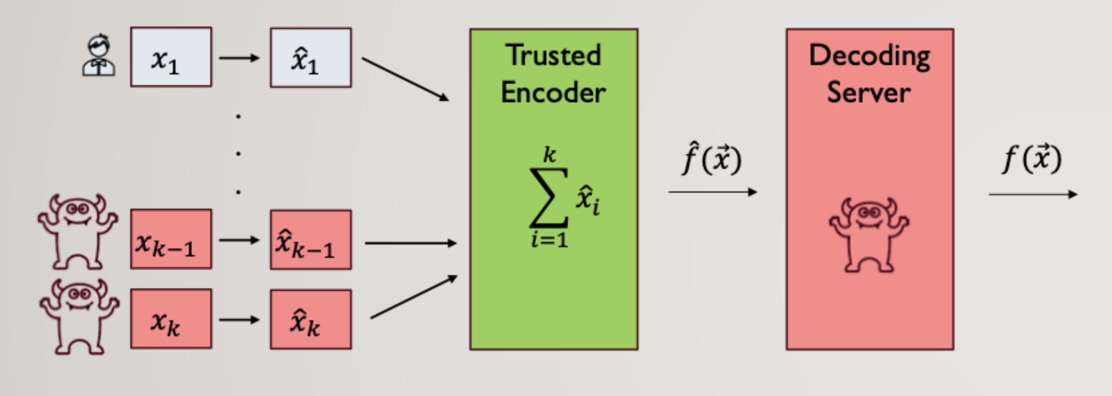
- 2-party simulation-secure RARE implies VBB Obfuscation ⇒ **Impossible**.

RARE implies Obfuscation



- 2-party simulation-secure RARE implies VBB Obfuscation ⇒ **Impossible**.
- Instead, indistinguishability security.

RARE implies Obfuscation



<u>Our focus</u>: **Indistinguishability** security:

For every $\vec{x}_H, \vec{w}_H$, with $f_{\vec{x}_H} \equiv f_{\vec{w}_H}$,

$$\left(pp, \hat{f}(\vec{x}_H)\right) \approx_c \left(pp, \hat{f}(\vec{w}_H)\right)$$

Implies iO

- 2-party simulation-secure RARE implies VBB Obfuscation ⇒ **Impossible**.
- Instead, indistinguishability security.

Known Results From [HIKR23]

Known Results From [HIKR23]

- Informtion-theoretic RARE for some functions (or, max, capped sum, etc.).

Known Results From [HIKR23]

- Informtion-theoretic RARE for some functions (or, max, capped sum, etc.).
- Computational (non-robust) ARE for all efficient functions.
  (from standard assumptions in bilinear groups)

Known Results From [HIKR23]

- Informtion-theoretic RARE for some functions (or, max, capped sum, etc.).
- Computational (non-robust) ARE for all efficient functions.
  (from standard assumptions in bilinear groups)
- Simulation-based RARE for all efficient functions in the ideal-obfuscation model.

Known Results From [HIKR23]

- Informtion-theoretic RARE for some functions (or, max, capped sum, etc.).
- Computational (non-robust) ARE for all efficient functions.
  (from standard assumptions in bilinear groups)
- Simulation-based RARE for all efficient functions in the ideal-obfuscation model.

Open question:

Can we construct **indistinguishability-based R**ARE for all efficient functions from IO and standard cryptographic assumptions? (in the plain model)

Our Results

1. **Indistinguishability-based RARE** from **IO** and (a new primitive we call) Pseudo Non Linear Codes (**PNLC**).

Our Results

1. **Indistinguishability-based RARE** from **IO** and (a new primitive we call) Pseudo Non Linear Codes (**PNLC**).

2. **PNLC** from either **LWE** or **DDH**.

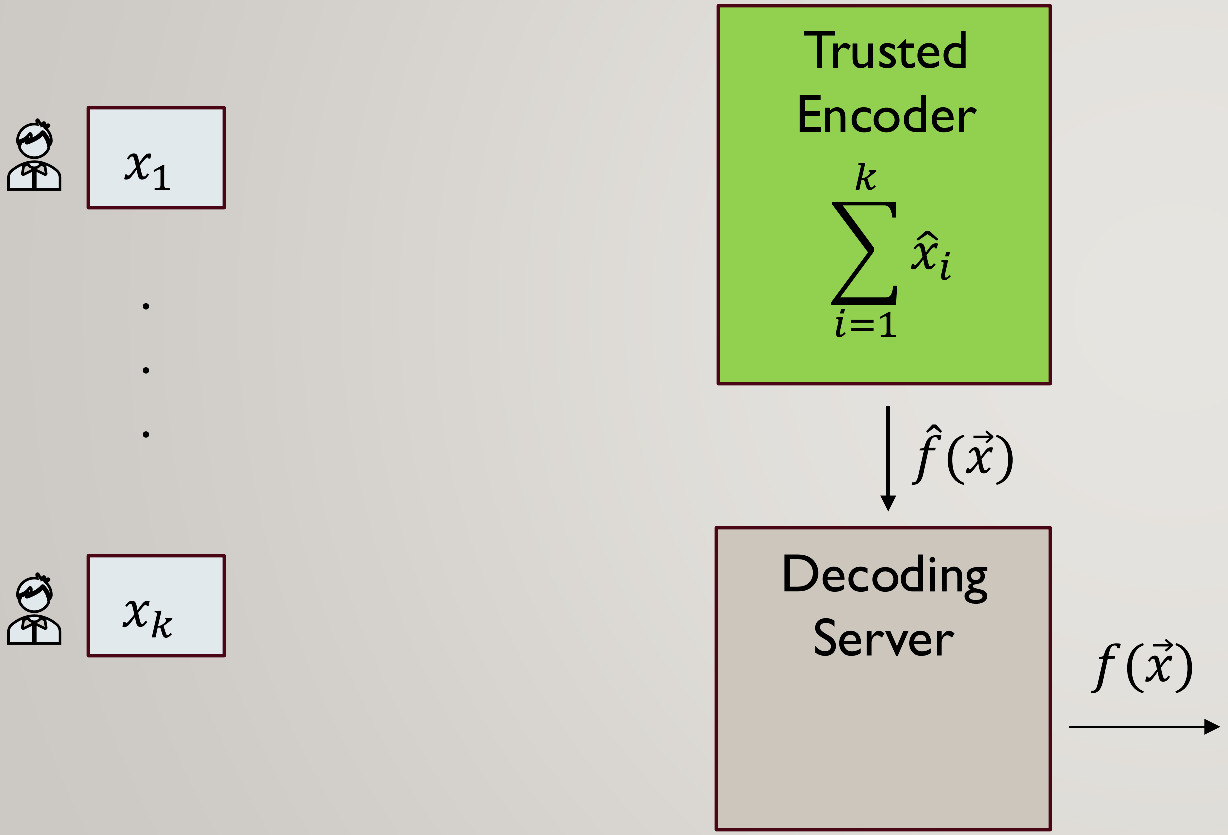
Our Results

1. **Indistinguishability-based RARE** from **IO** and (a new primitive we call) Pseudo Non Linear Codes (**PNLC**).

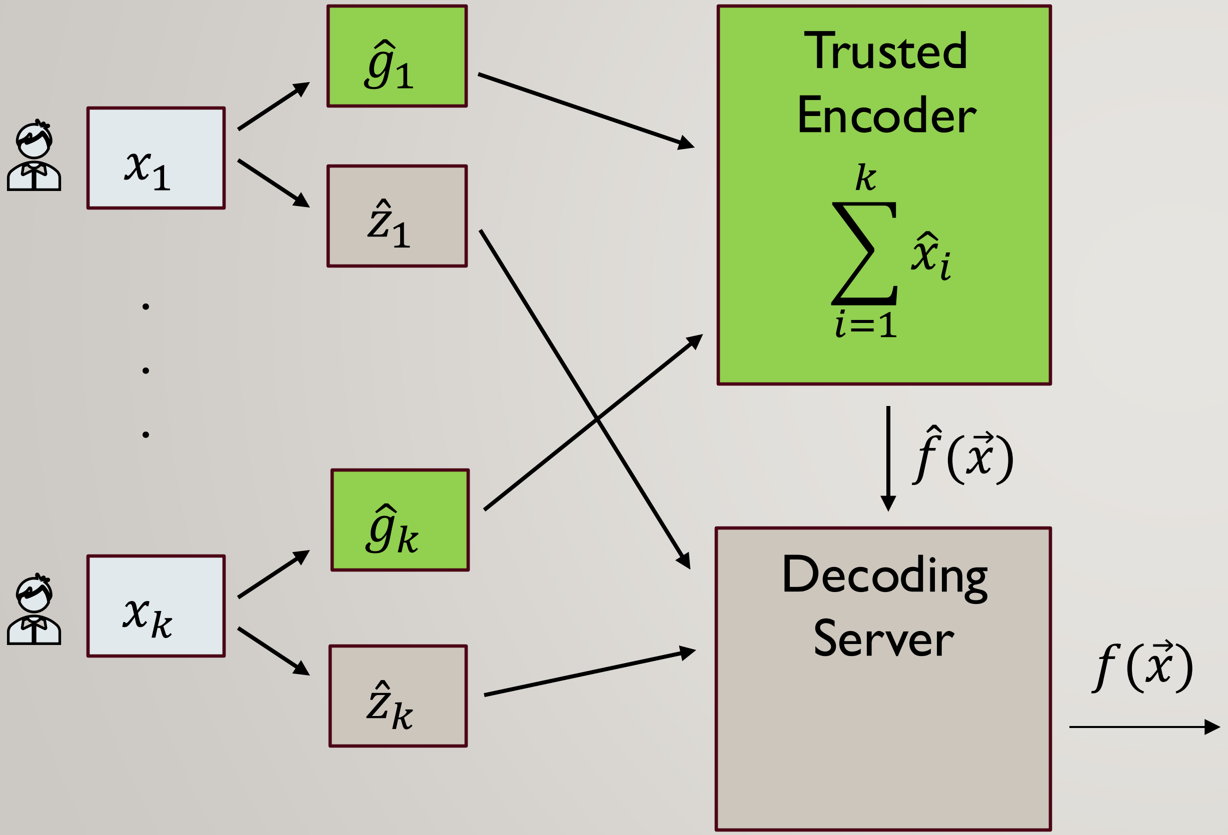
2. **PNLC** from either **LWE** or **DDH**.

3. Our **RARE** is **succinct** (more in next slide).

Succinct RARE



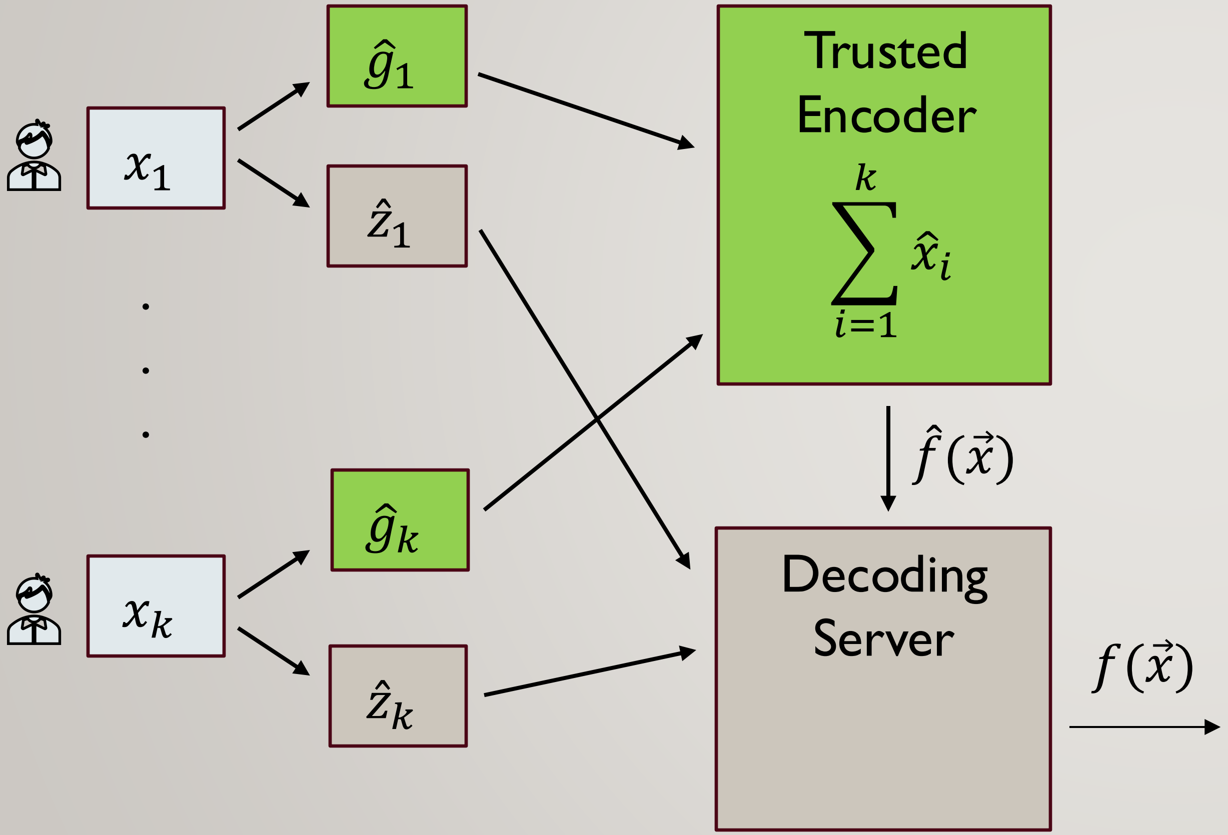
Succinct RARE



Split parties' encodings $\hat{x}_i = (\hat{z}_i, \hat{g}_i)$.

- $\hat{g}_i$ group element, $\hat{z}_i$ non-interactive part.

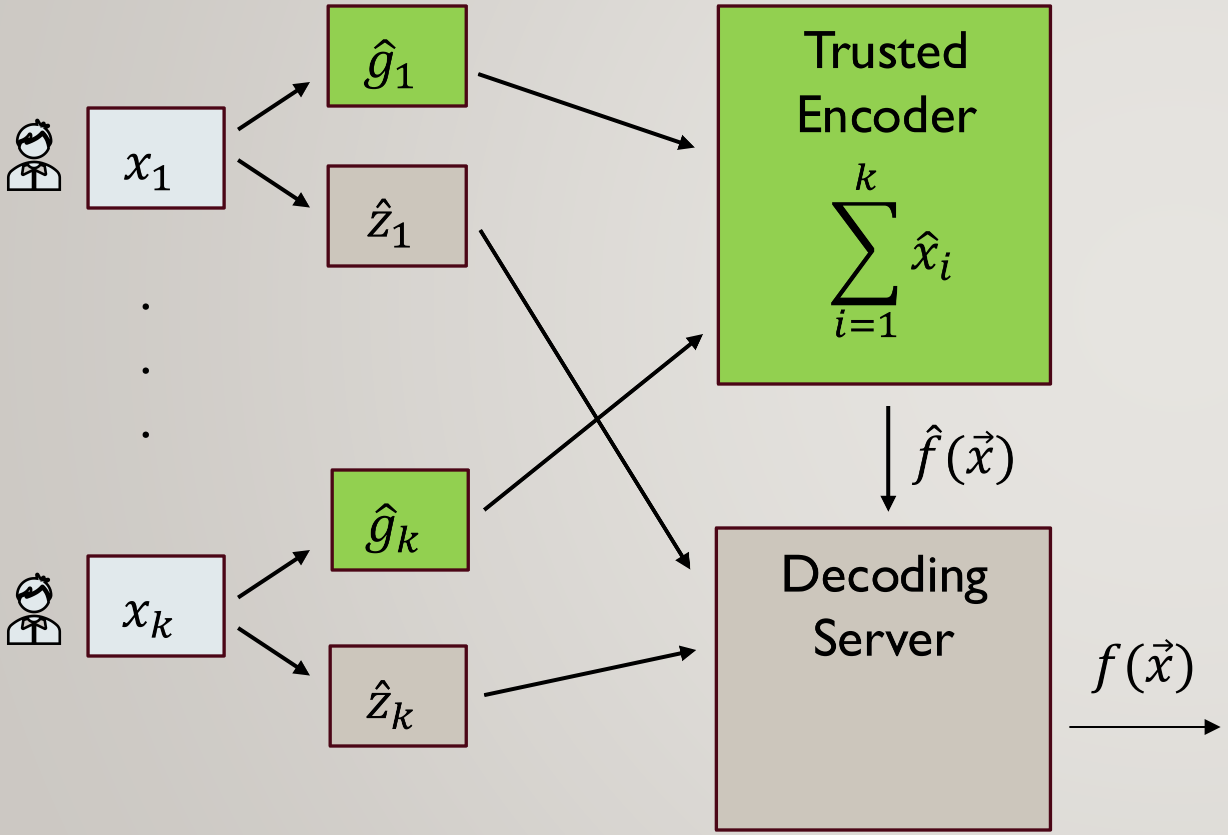
Succinct RARE



Split parties' encodings $\hat{x}_i = (\hat{z}_i, \hat{g}_i)$.

- $\hat{g}_i$ group element, $\hat{z}_i$ non-interactive part.
- Minimal communication complexity:
  - $|\hat{g}_i| \sim$ **security parameter**.
  - $|\hat{z}_i| \sim$ size of the **input** x_i.
- Independent of $k, |f|$

Succinct RARE

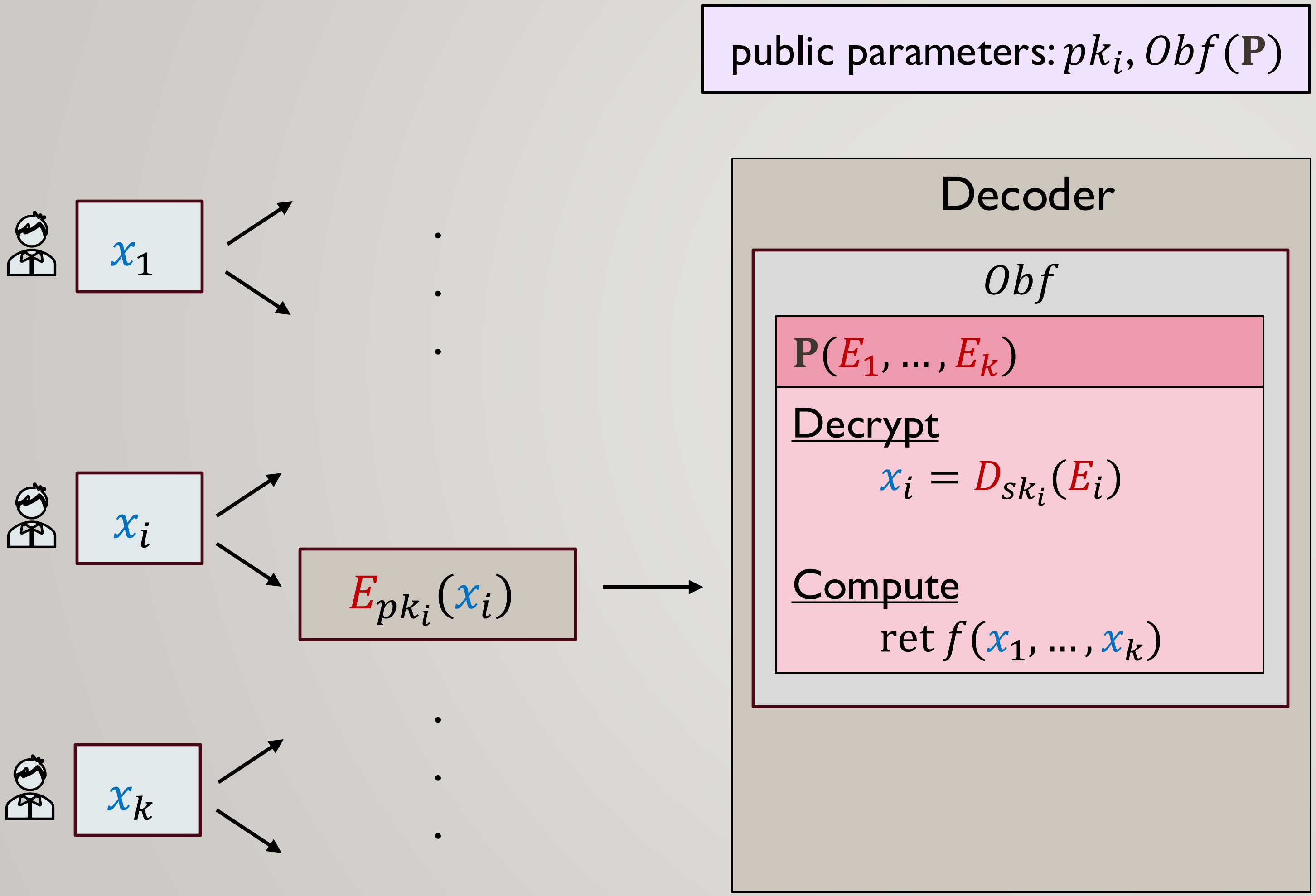


Split parties' encodings $\hat{x}_i = (\hat{z}_i, \hat{g}_i)$.

- $\hat{g}_i$ group element, $\hat{z}_i$ non-interactive part.
- Minimal communication complexity:
  - $|\hat{g}_i| \sim$ **security parameter**.
  - $|\hat{z}_i| \sim$ size of the **input** x_i.
- Independent of $k, |f|$
- Trusted computation is minimal.

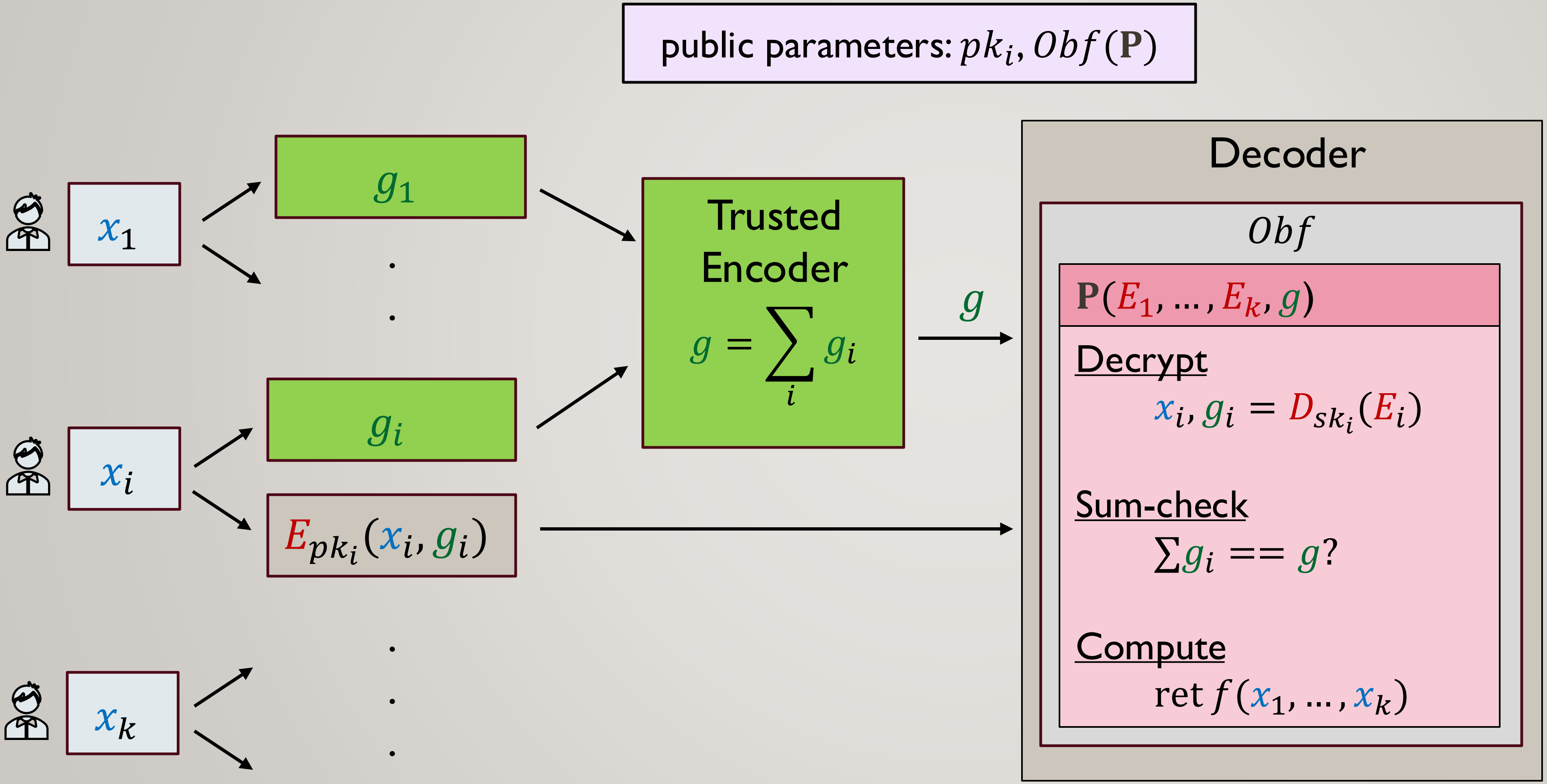
And now, the construction

Warmup: Ideal Obfuscation



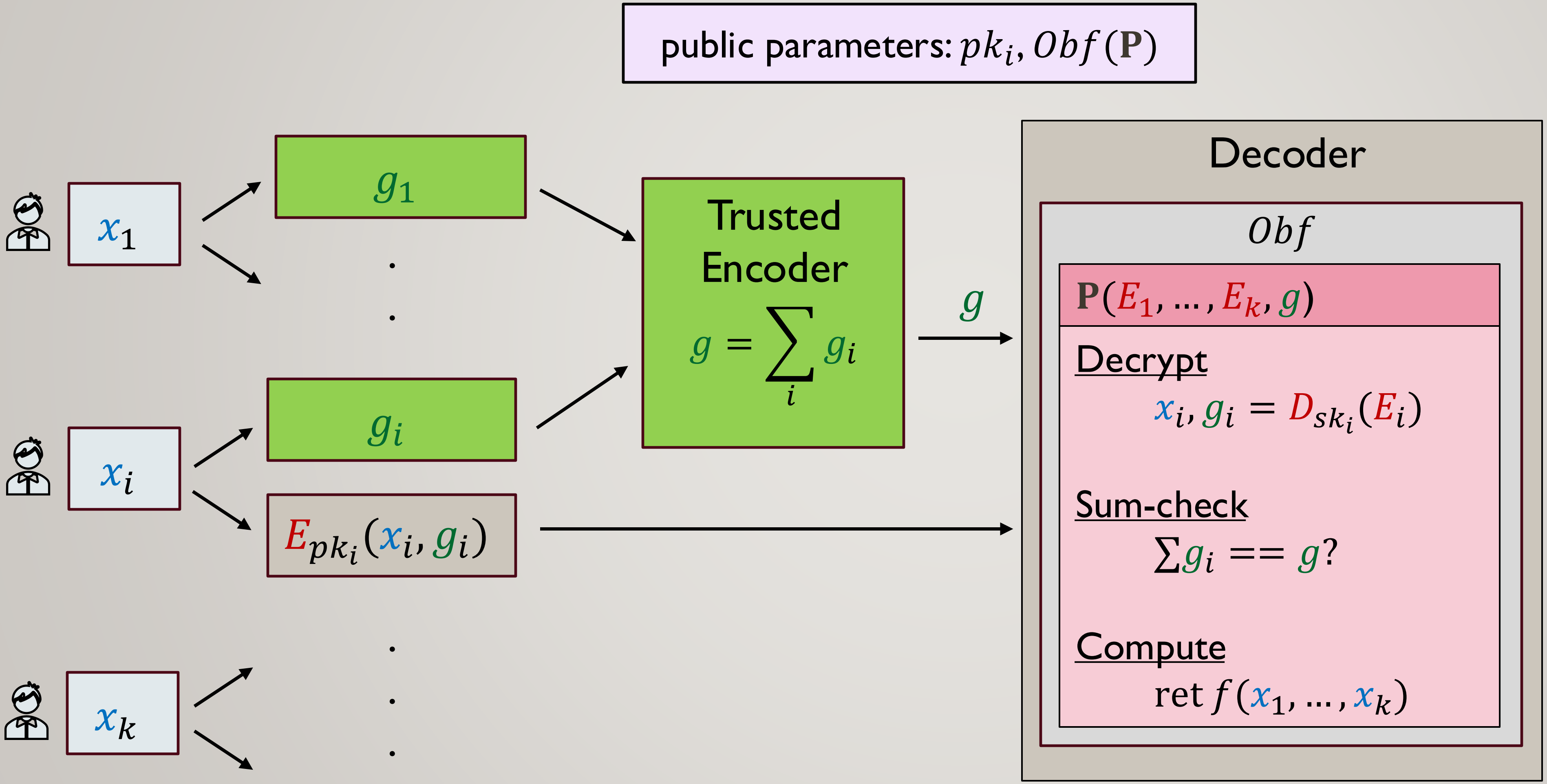
Problem:
Can change subsets of the honest parties' inputs.

Warmup: Ideal Obfuscation



- Idea: "glue" honest parties together.
- Add additive elements g_i.

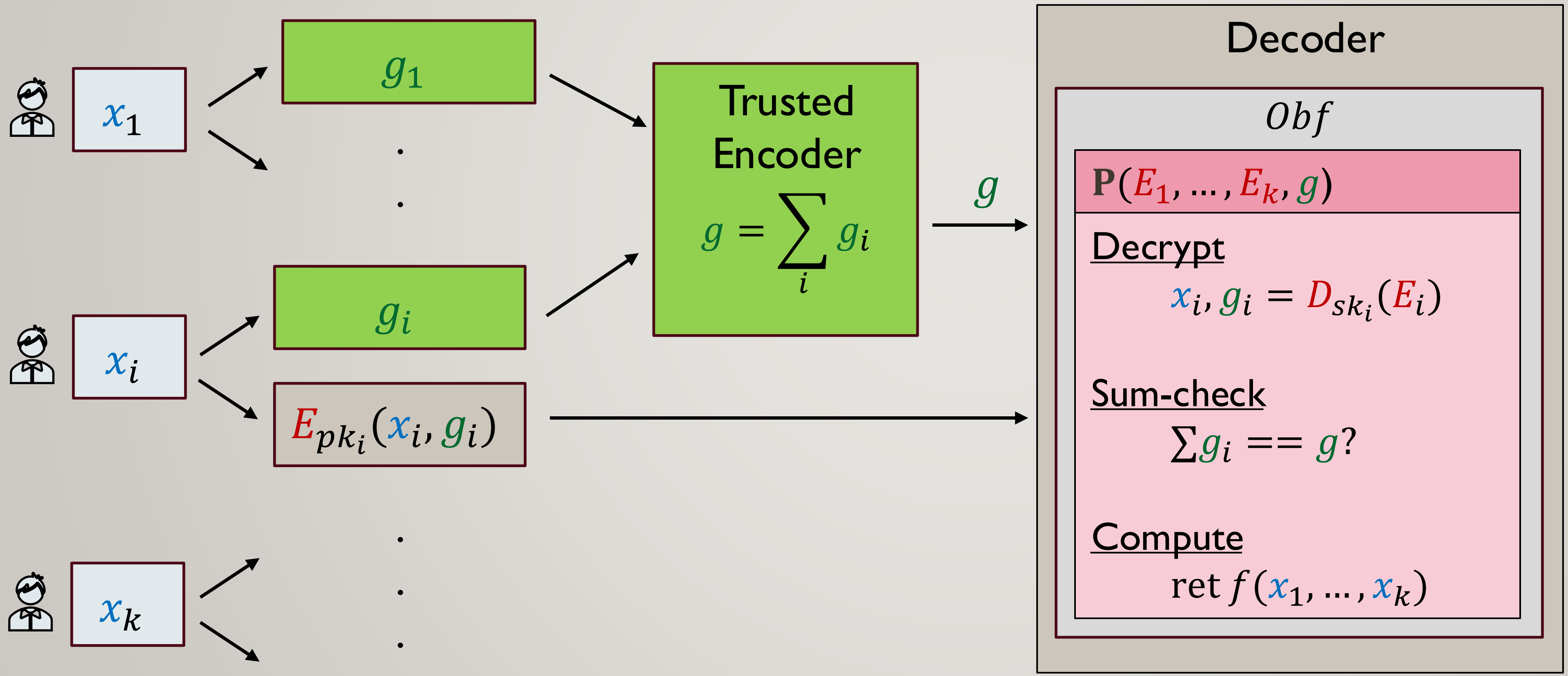
Warmup: Ideal Obfuscation



- Idea: "glue" honest parties together.
- Add additive elements g_i.
- Non-malleable encryption (CCA2-secure).

Warmup: Ideal Obfuscation

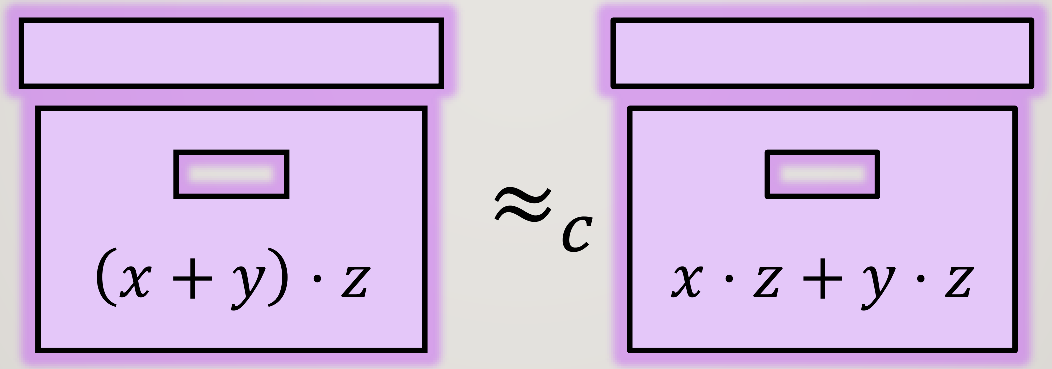
public parameters: $pk_i, Obf(\mathbf{P})$



- Idea: "glue" honest parties together.
- Add additive elements g_i.
- Non-malleable encryption (CCA2-secure).
- Secure in the ideal-obf model.

Moving To iO

$$f_1 \equiv f_2 \Longrightarrow iO(f_1) \approx_c iO(f_2)$$



Moving To $i\mathcal{O}$: First Problem

Moving To iO: First Problem

- CCA2- PKE is not necessarily iO friendly.

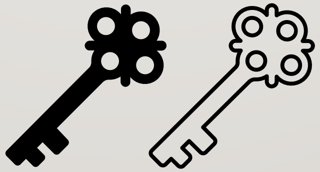
Moving To iO: First Problem

- CCA2- PKE is not necessarily iO friendly.
- Has already been handled before (e.g, in io-based constructions of FE [GGH+13]).

Moving To $i\mathcal{O}$: First Problem

- CCA2- PKE is not necessarily $i\mathcal{O}$ friendly.
- Has already been handled before (e.g, in io-based constructions of FE [GGH+13]).
- Solution: Naor-Yung **double** encryption with a Statistically-Simulation-Sound NIZK [Sah99].

$$E(m) = (E_{left}(m), E_{right}(m), \Pi_{NIZK})$$



Moving To $i\mathcal{O}$: Second Problem (simplified)

Moving To iO: Second Problem (simplified)

Let $f_{\vec{a}_H} \equiv f_{\vec{b}_H}$. Goal:

$$iO(\mathbf{P}),\ E_i^*(a_i, g_i^*),\ \Sigma_{i\in H}\, g_i^*$$
$$\approx_c$$
$$iO(\mathbf{P}),\ E_i^*(b_i, g_i^*),\ \Sigma_{i\in H}\, g_i^*$$

"need to replace encryptions of a with encryptions of b."

Moving To iO: Second Problem (simplified)

Let $f_{\vec{a}_H} \equiv f_{\vec{b}_H}$. Goal:

$$iO(\mathbf{P}),\ E_i^*(a_i, g_i^*),\ \sum_{i \in H} g_i^* \approx_c iO(\mathbf{P}),\ E_i^*(b_i, g_i^*),\ \sum_{i \in H} g_i^*$$

"need to replace encryptions of a with encryptions of b."

iO Hybrids

$\Rightarrow$

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$x_i, g_i = \boxed{a_i,} g_i^*$

Dec & Sum-check & Compute

$\equiv ?$

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$x_i, g_i = \boxed{b_i,} g_i^*$

Dec & Sum-check & Compute

Moving To iO: Second Problem (simplified)

Let $f_{\vec{a}_H} \equiv f_{\vec{b}_H}$. Goal:

$$iO(\mathbf{P}),\ E_i^*(a_i, g_i^*),\ \Sigma_{i\in H}\, g_i^*$$
$$\approx_c$$
$$iO(\mathbf{P}),\ E_i^*(b_i, g_i^*),\ \Sigma_{i\in H}\, g_i^*$$

"need to replace encryptions of a with encryptions of b."

iO Hybrids

$\Rightarrow$

$\mathbf{P}'(E_1, \ldots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$x_i, g_i = \boxed{a_i}, g_i^*$

Dec & Sum-check & Compute

$\equiv ?$

$\mathbf{P}'(E_1, \ldots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$x_i, g_i = \boxed{b_i}, g_i^*$

Dec & Sum-check & Compute

Problem:

If $\exists$ **subset** $F \subsetneq H$: $f(\vec{a}_F, \cdot) \not\equiv f(\vec{b}_F, \cdot)$

$\Rightarrow$

different functionalities (might be hard to find).

Solution: Pseudo-non-linear-codes

$\mathrm{P}'(E_1, \ldots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{a_i}, g_i^*$$

Dec & Sum-check & Compute

$\equiv ?$

$\mathrm{P}'(E_1, \ldots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{b_i}, g_i^*$$

Dec & Sum-check & Compute

Replace group elements g_i with PNLC encodings $\hat{g}_i$.

Pseudo Non Linear Codes

Solution: Pseudo-non-linear-codes

$\mathrm{P}'(E_1, \dots, E_k, g)$
For every $i \in H$ s.t. $E_i = E_i^*$:
$x_i, g_i = \boxed{a_i}, g_i^*$
Dec & Sum-check & Compute

$\equiv ?$

$\mathrm{P}'(E_1, \dots, E_k, g)$
For every $i \in H$ s.t. $E_i = E_i^*$:
$x_i, g_i = \boxed{b_i}, g_i^*$
Dec & Sum-check & Compute

Replace group elements g_i with PNLC encodings $\hat{g}_i$.

Pseudo Non Linear Codes

1. Homomorphicly additive $\Rightarrow$ can do sum-check stage.

Solution: Pseudo-non-linear-codes

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{a_i}, g_i^*$$

Dec & Sum-check & Compute

$\equiv ?$

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{b_i}, g_i^*$$

Dec & Sum-check & Compute

Replace group elements g_i with PNLC encodings $\hat{g}_i$.

Pseudo Non Linear Codes

1. Homomorphicly additive $\Rightarrow$ can do sum-check stage.
2. Admits fake-encodings, for which **subset sums** evade the code. The fake-encodings are $\approx_c$ from valid encodings.

Solution: Pseudo-non-linear-codes

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{a_i}, g_i^*$$

Dec & Sum-check & Compute

$\equiv ?$

$\mathbf{P}'(E_1, \dots, E_k, g)$

For every $i \in H$ s.t. $E_i = E_i^*$:

$$x_i, g_i = \boxed{b_i}, g_i^*$$

Dec & Sum-check & Compute

Replace group elements g_i with PNLC encodings $\hat{g}_i$.

Pseudo Non Linear Codes

1. Homomorphicly additive $\Rightarrow$ can do sum-check stage.
2. Admits fake-encodings, for which **subset sums** evade the code. The fake-encodings are $\approx_c$ from valid encodings.
3. Can construct from wither LWE or DDH

More details in the paper…

Future Direction - ARE

Improving **R**ARE
1. Simpler **public parameters**? no setup at all?
2. Assumptions **lighter then iO** for limited classes of functions.

(non-robust) ARE
1. Does **statistically** secure ARE for all efficient functions exist?
2. What assumptions imply **computational** ARE? Post-quantum? PK cryptography needed?

Future Direction - ARE

Improving **R**ARE
1. Simpler **public parameters**? no setup at all?
2. Assumptions **lighter then iO** for limited classes of functions.

(non-robust) ARE
1. Does **statistically** secure ARE for all efficient functions exist?
2. What assumptions imply **computational** ARE? Post-quantum? PK cryptography needed?

Thank You