Secure Computation on Encrypted Data in Multi-User Systems

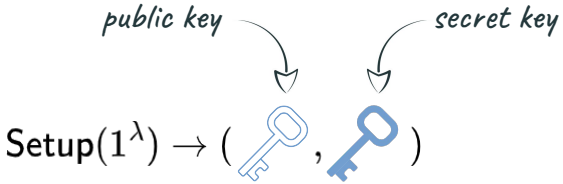
— PhD Thesis Defense —

Robert Schädlich

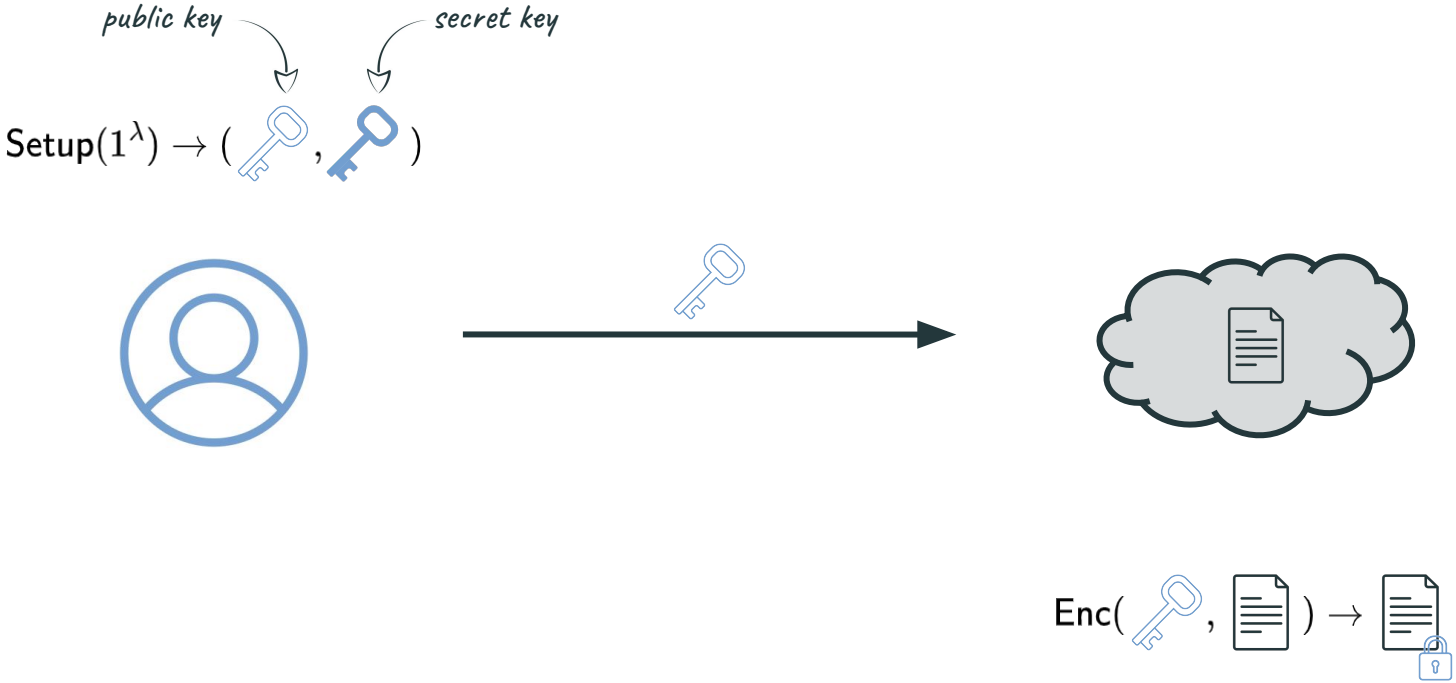
December 2, 2025

DIENS, École normale supérieure, PSL University, CNRS, Inria

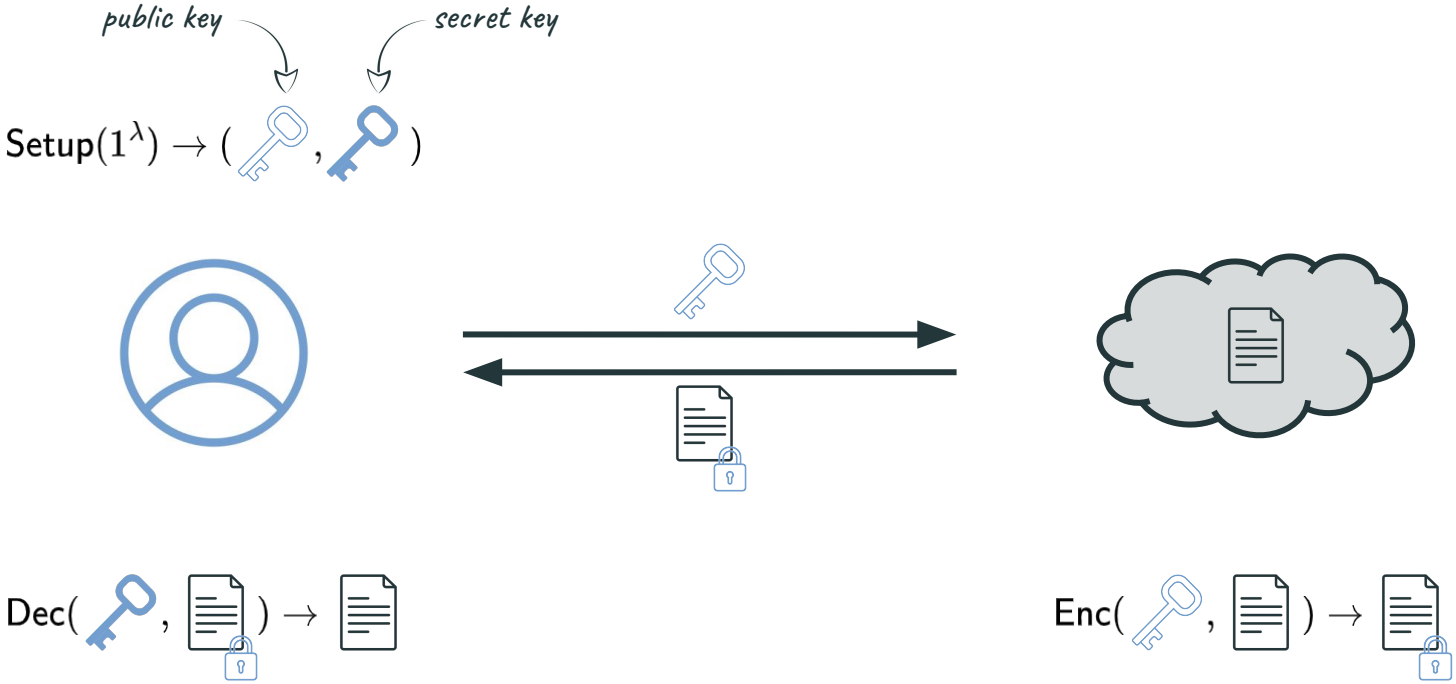
Public-Key Encryption



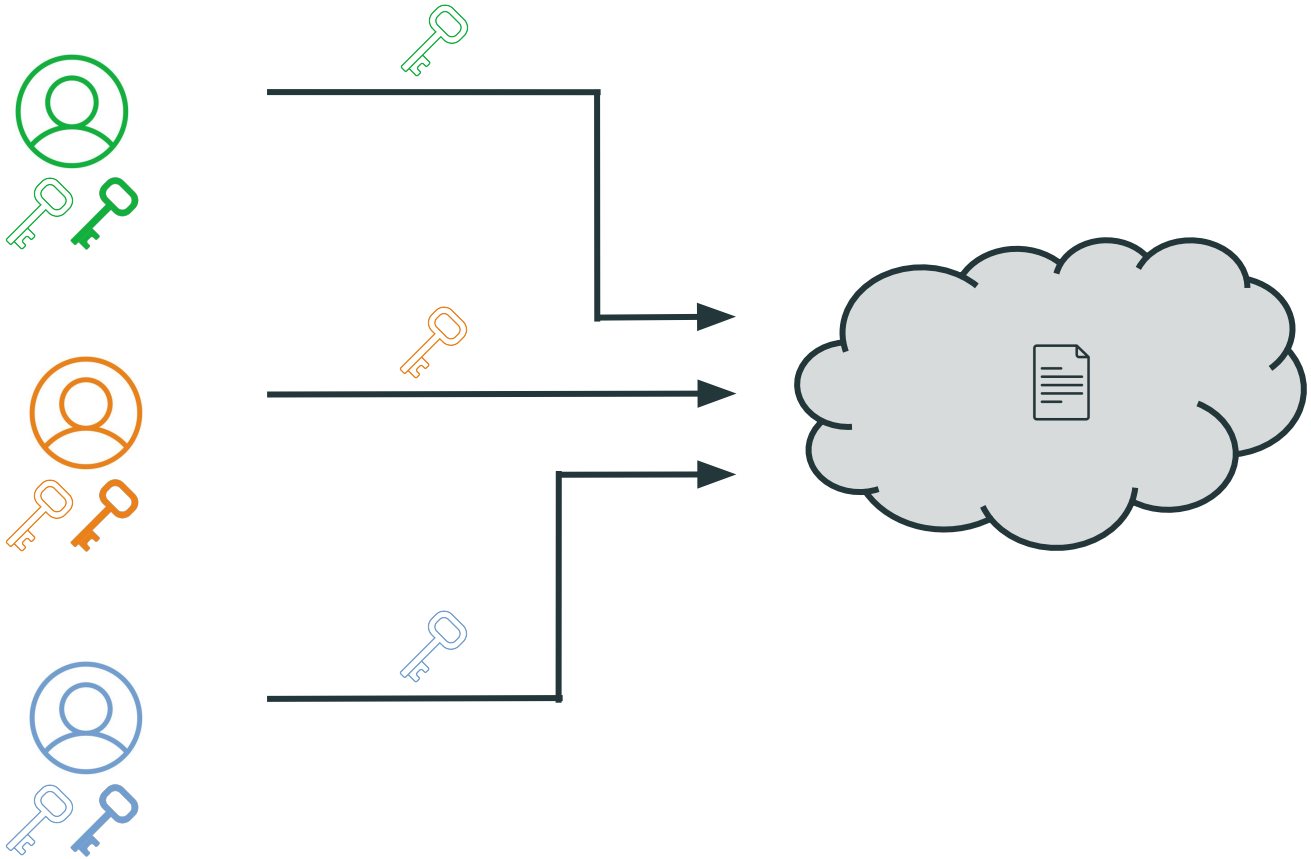
Public-Key Encryption



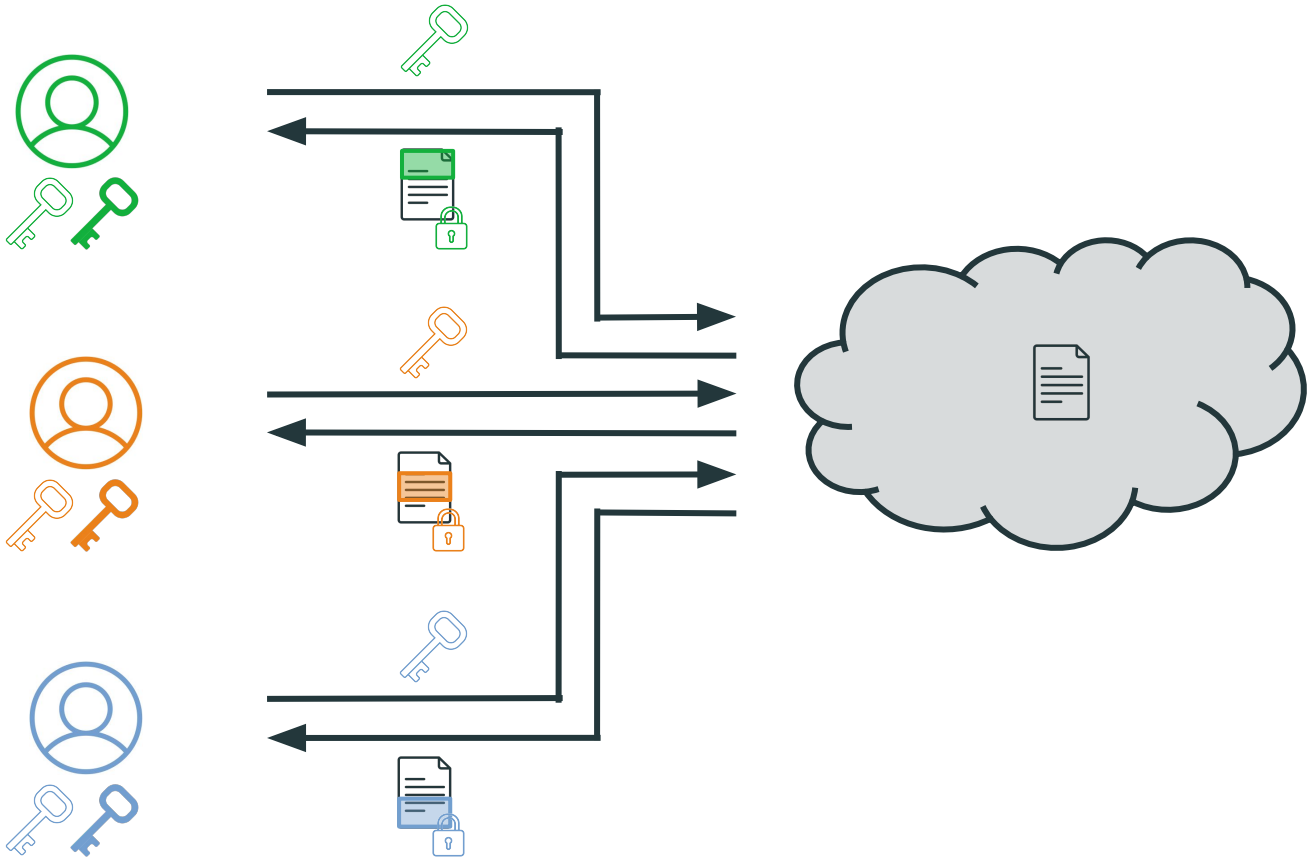
Public-Key Encryption



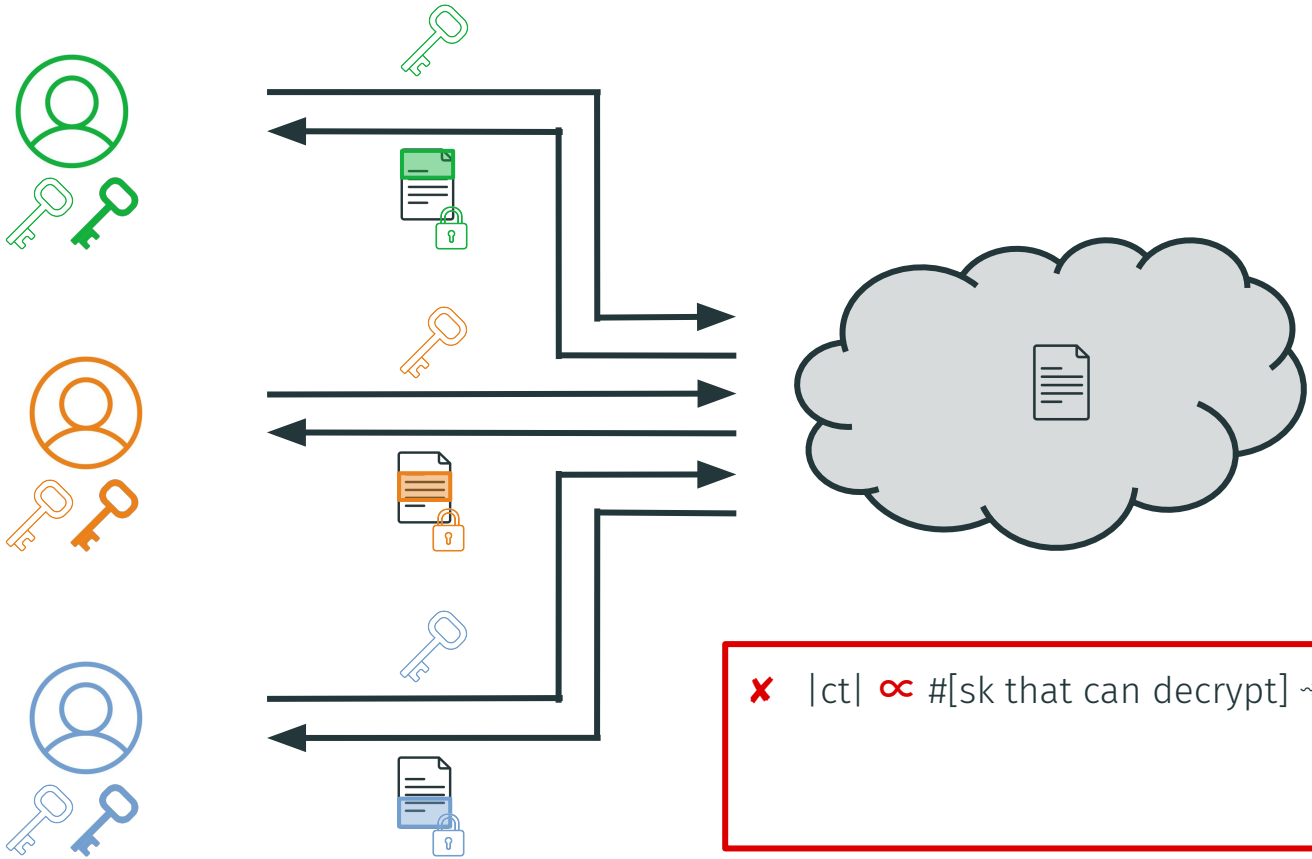
Public-Key Encryption



Public-Key Encryption

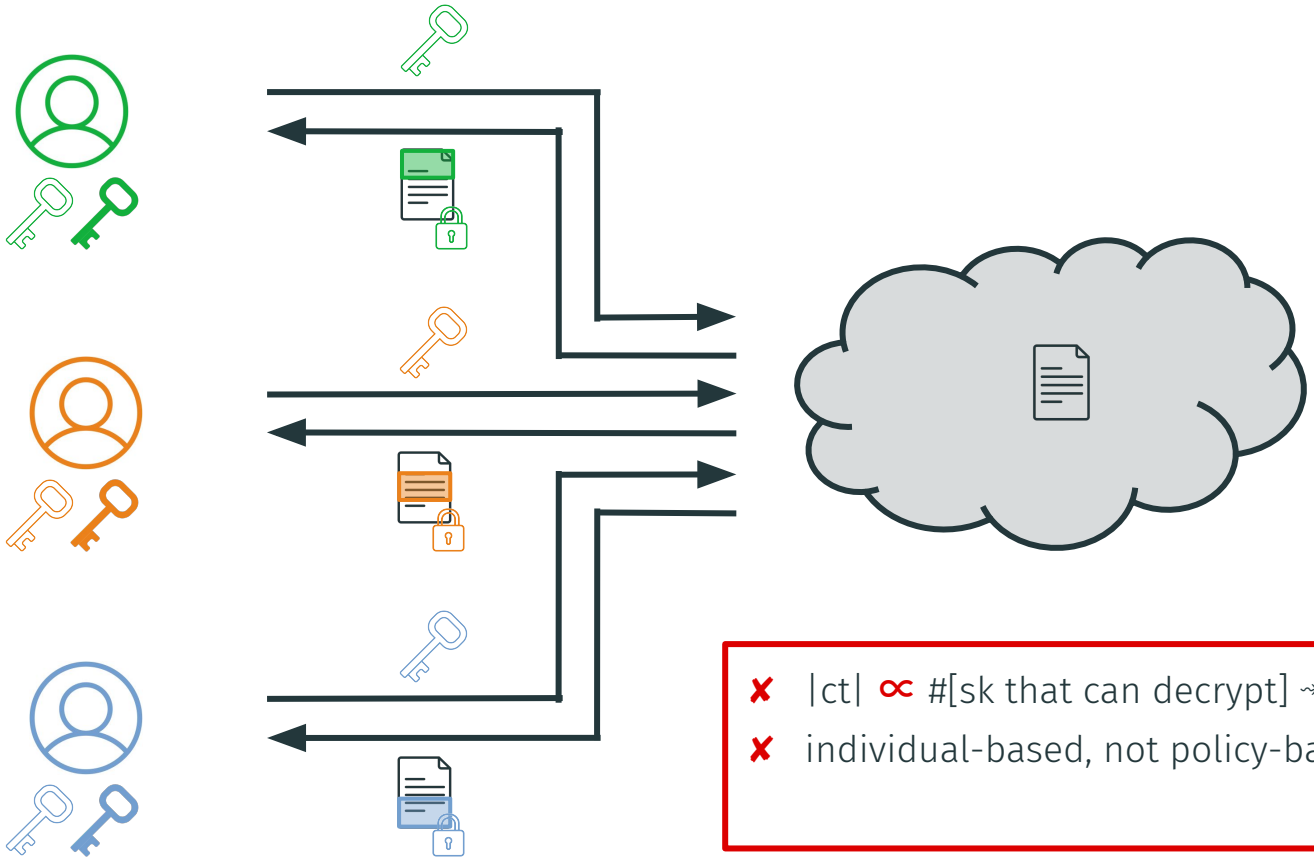


Public-Key Encryption



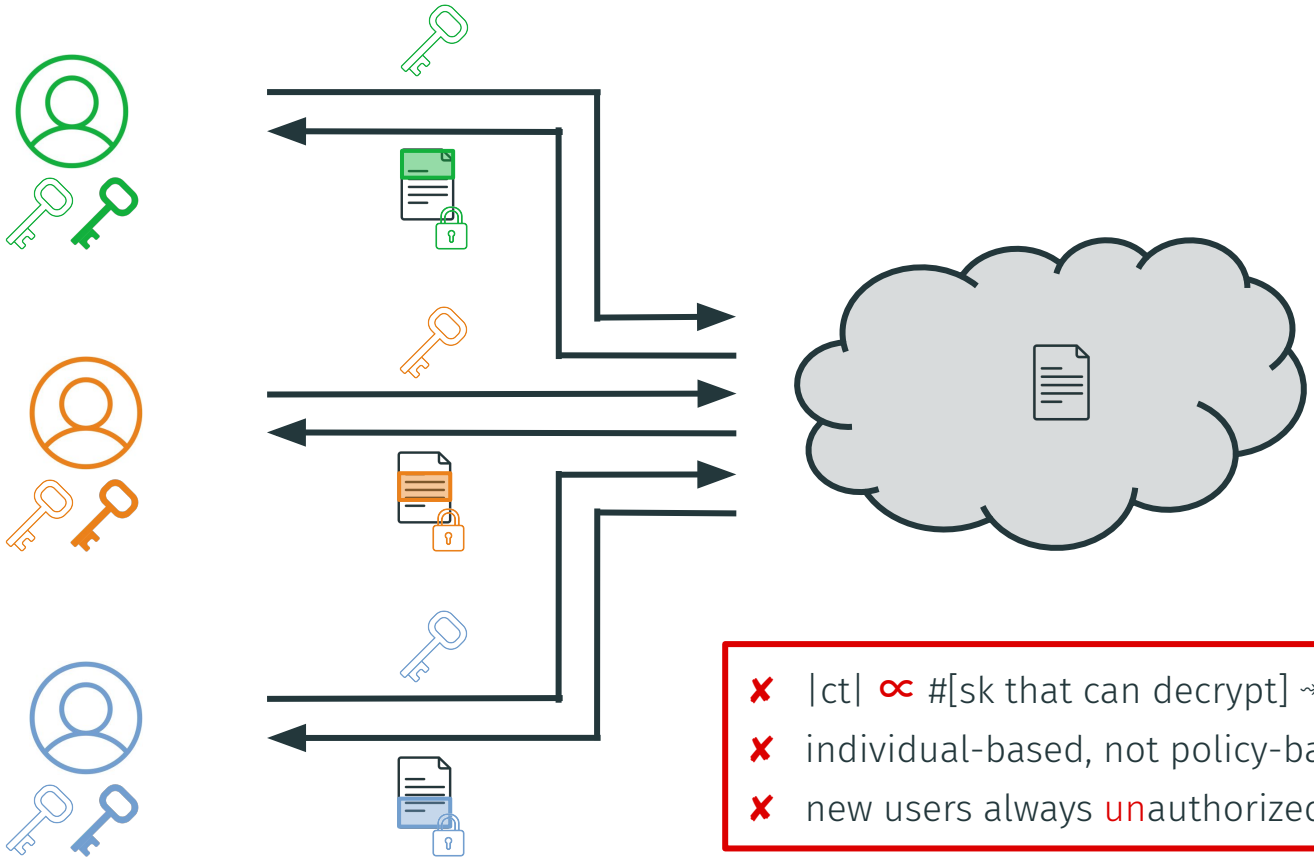
✘ |ct| ∝ #[sk that can decrypt] ↛ not scalable

Public-Key Encryption



- ✘ |ct| ∝ #[sk that can decrypt] ⇝ not scalable
- ✘ individual-based, not policy-based

Public-Key Encryption

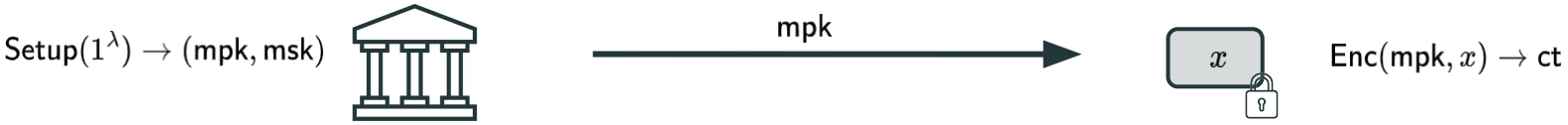


- ✘ |ct| ∝ #[sk that can decrypt] ↛ not scalable
- ✘ individual-based, not policy-based
- ✘ new users always unauthorized

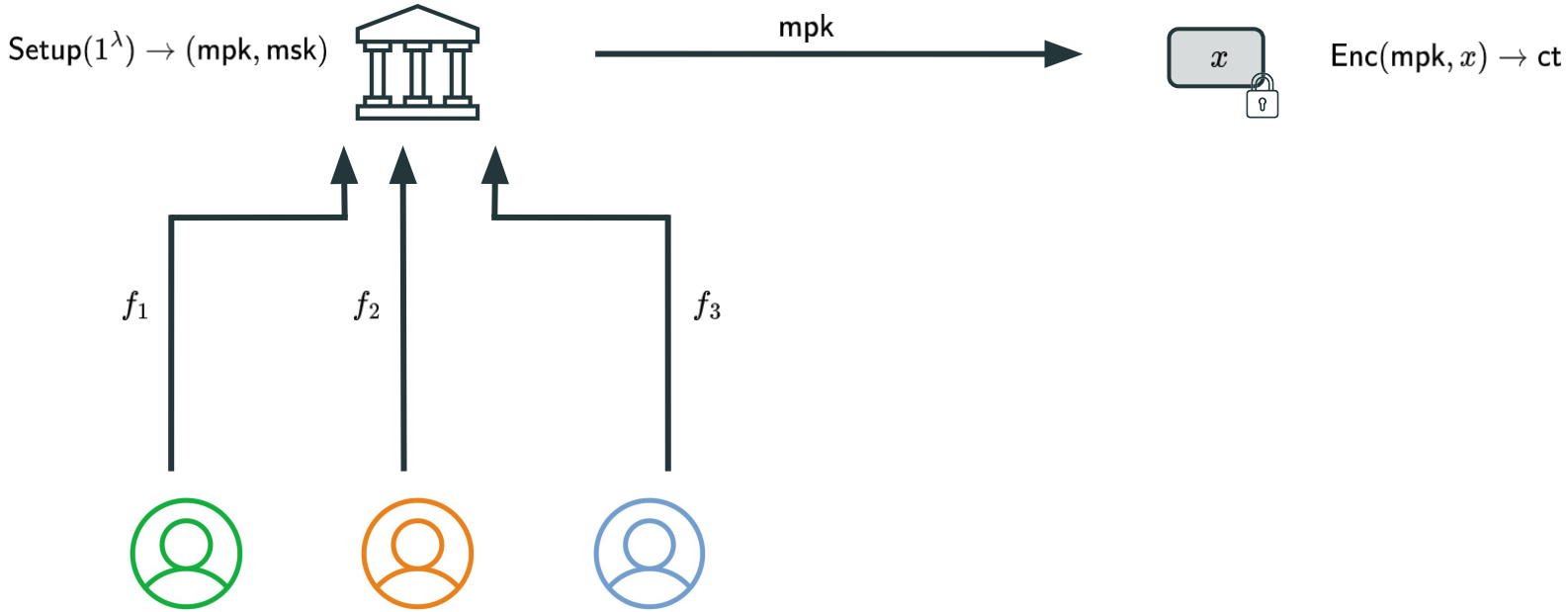
Functional Encryption [BSW11]

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

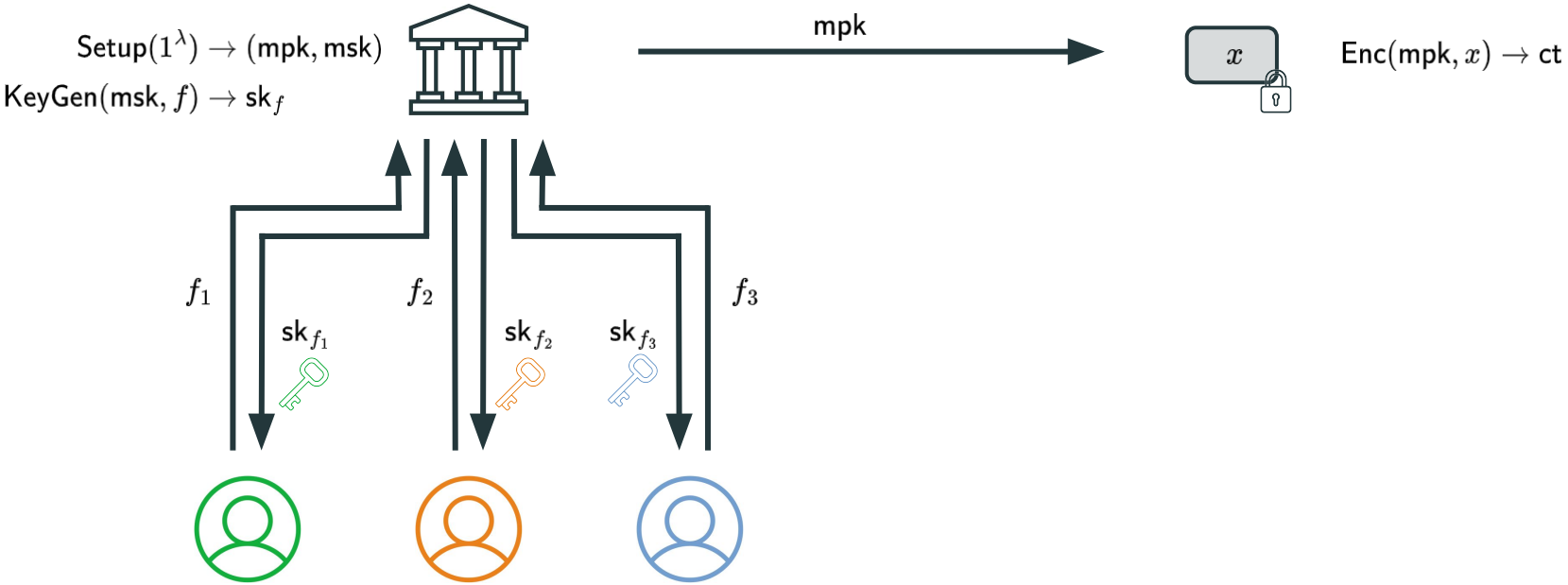
Functional Encryption [BSW11]



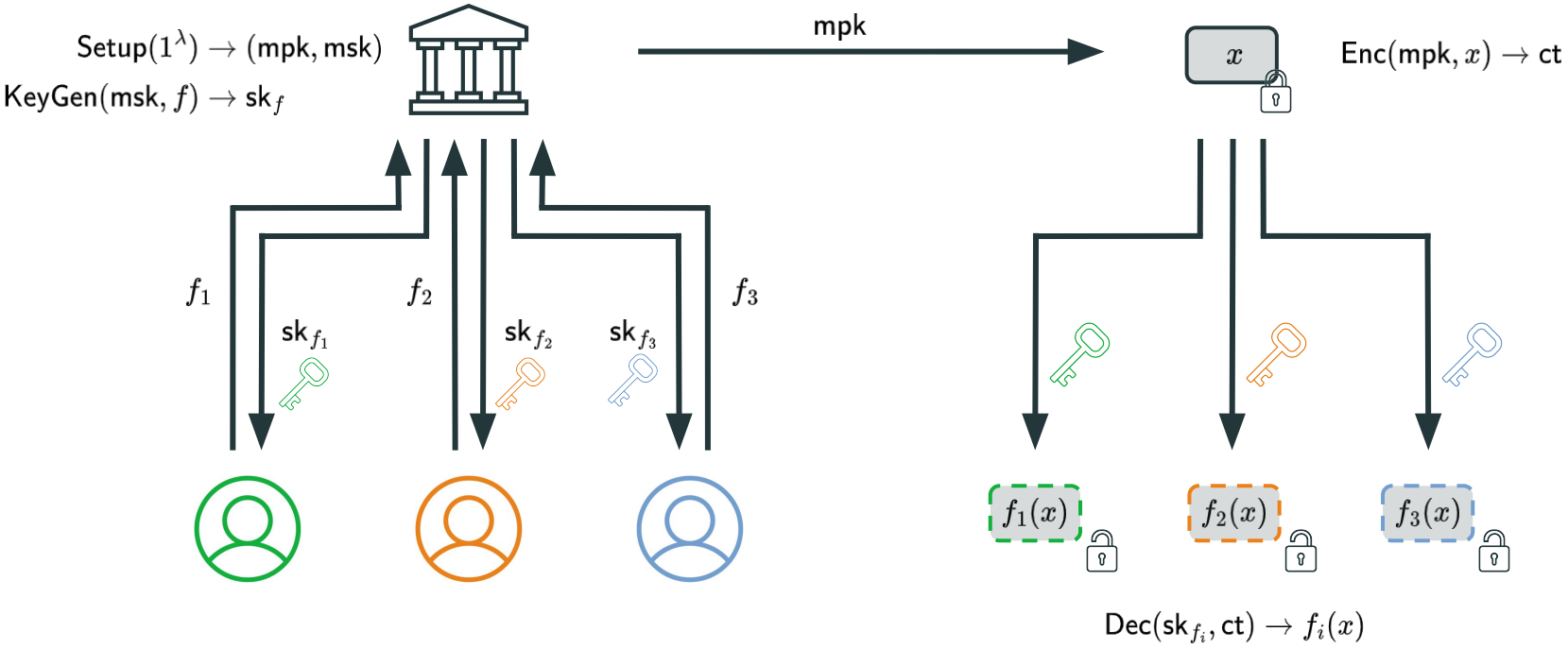
Functional Encryption [BSW11]



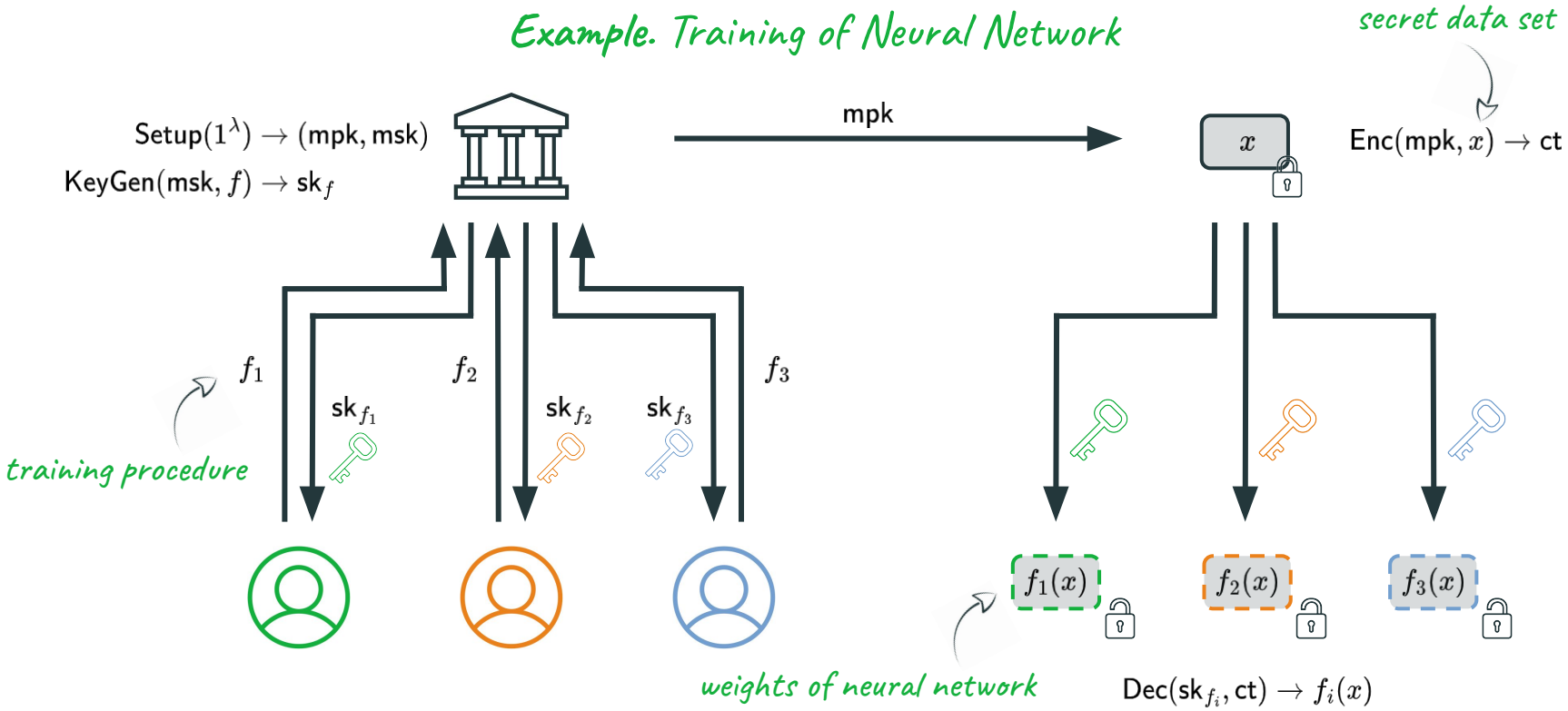
Functional Encryption [BSW11]



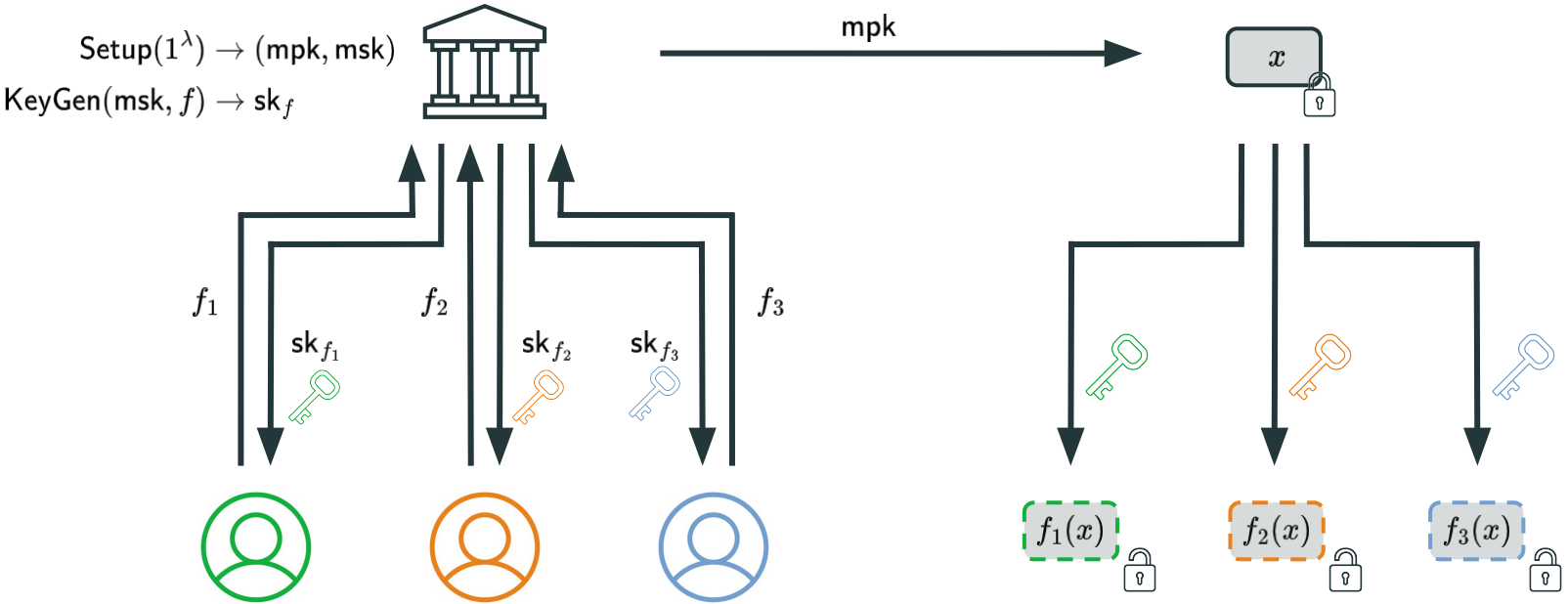
Functional Encryption [BSW11]



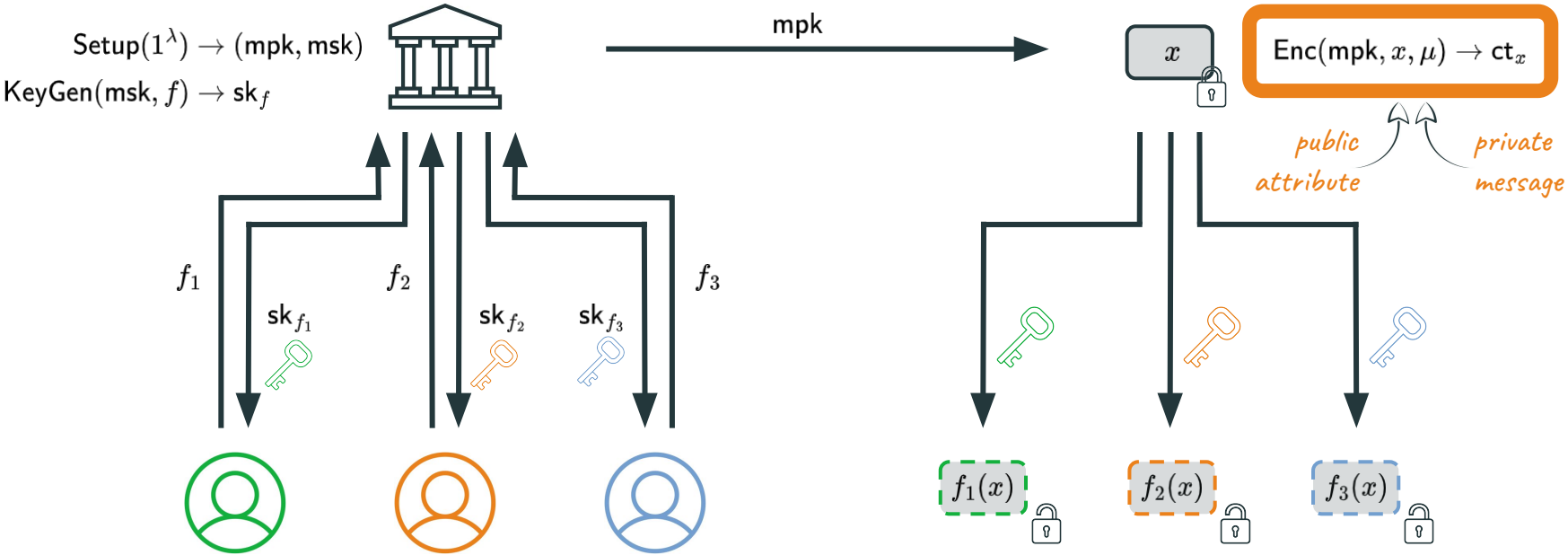
Functional Encryption [BSW11]



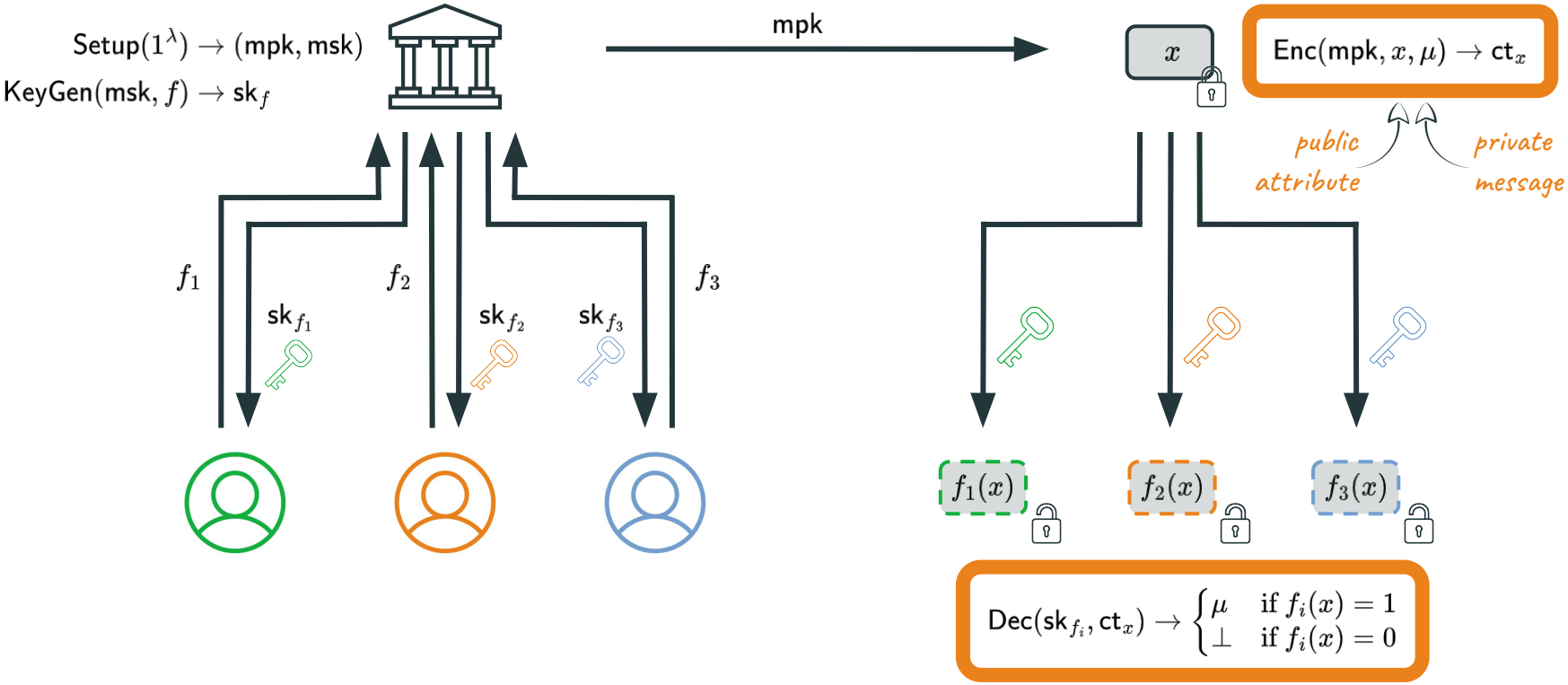
Special Case: Attribute-Based Encryption [SW05, GPSW06]



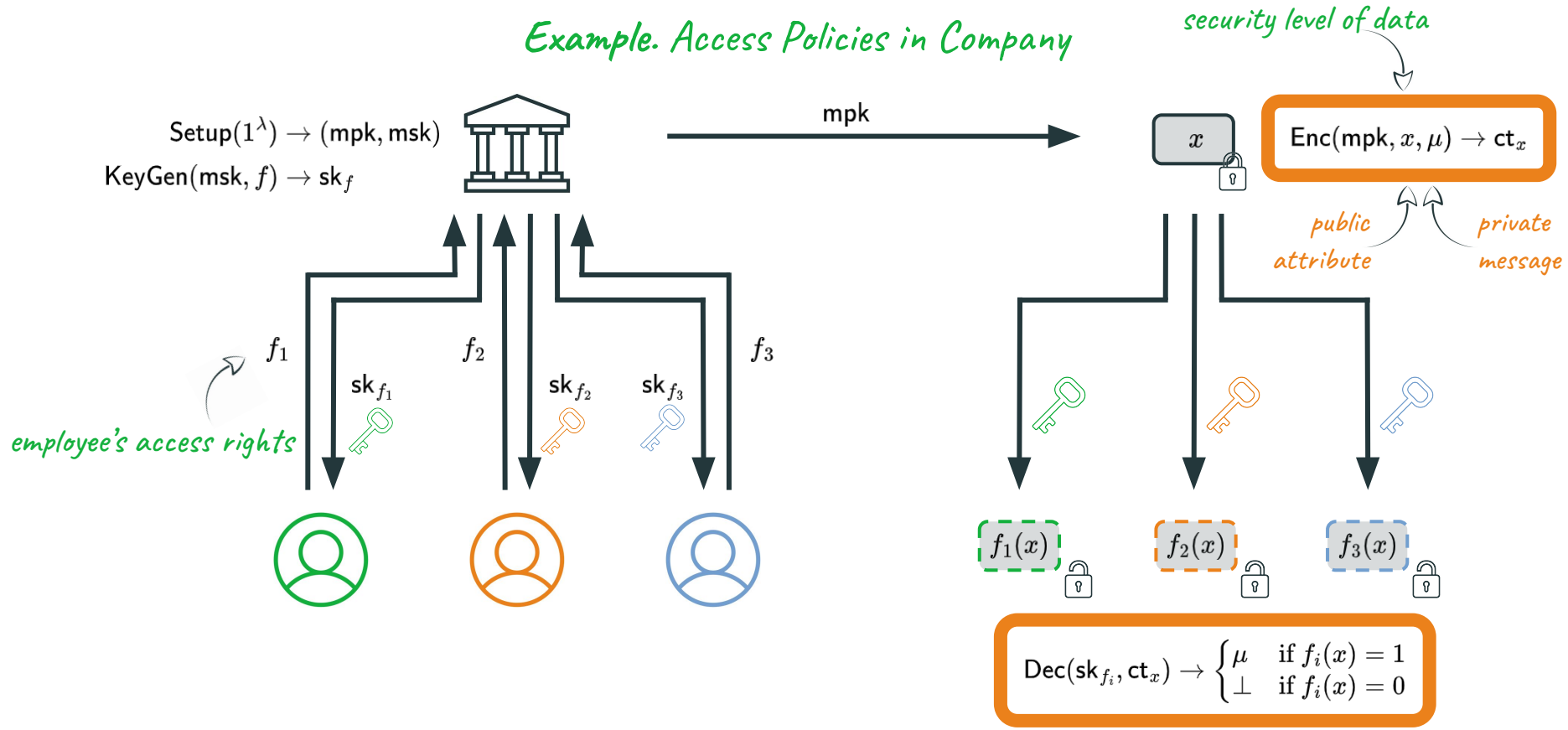
Special Case: Attribute-Based Encryption [SW05, GPSW06]



Special Case: Attribute-Based Encryption [SW05, GPSW06]



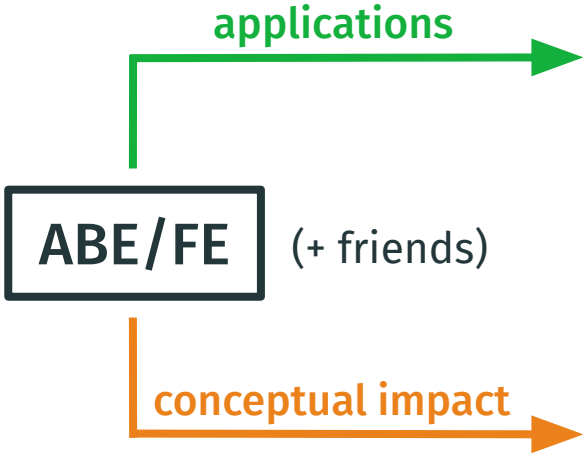
Special Case: Attribute-Based Encryption [SW05, GPSW06]



Why Study ABE and FE?

- audit logs [GPSW06]
- medical records [APGLPR11]
- private key distribution in cloud [Cloudflare17]
- money laundering detection [dPP22]

Why Study ABE and FE?



- audit logs [GPSW06]
- medical records [APGLPR11]
- private key distribution in cloud [Cloudflare17]
- money laundering detection [dPP22]

- implications to other primitives:
  - ABE $\Longrightarrow$ verifiable delegatable computation [PRV11]
  - FE $\Longrightarrow$ indistinguishability obfuscation [BV15, AJ15]
  - (MI-)ABE $\Longrightarrow$ non-trivial witness encryption [BJKPW17]
  - Reg-ABE $\Longrightarrow$ traitor tracing w/out authority [BLMMRW24]
- source of inspiration for new techniques:
  - succinct CP-ABE [W24, W25a] $\Longrightarrow$ SNARGs from SIS [W25b]

Organization

Part I. Removing the Trusted Authority (Registered ABE and FE)

- registered ABE via general paradigm [AC:P**S**25a]
- more [P**S**25b, P**S**T25]

Part II. Supporting Multiple Encryptors (Multi-Client ABE)

- formal definition, first instantiations [TCC:P**S**24]
- more functionalities, different assumptions [PKC:**S**25]

Organization

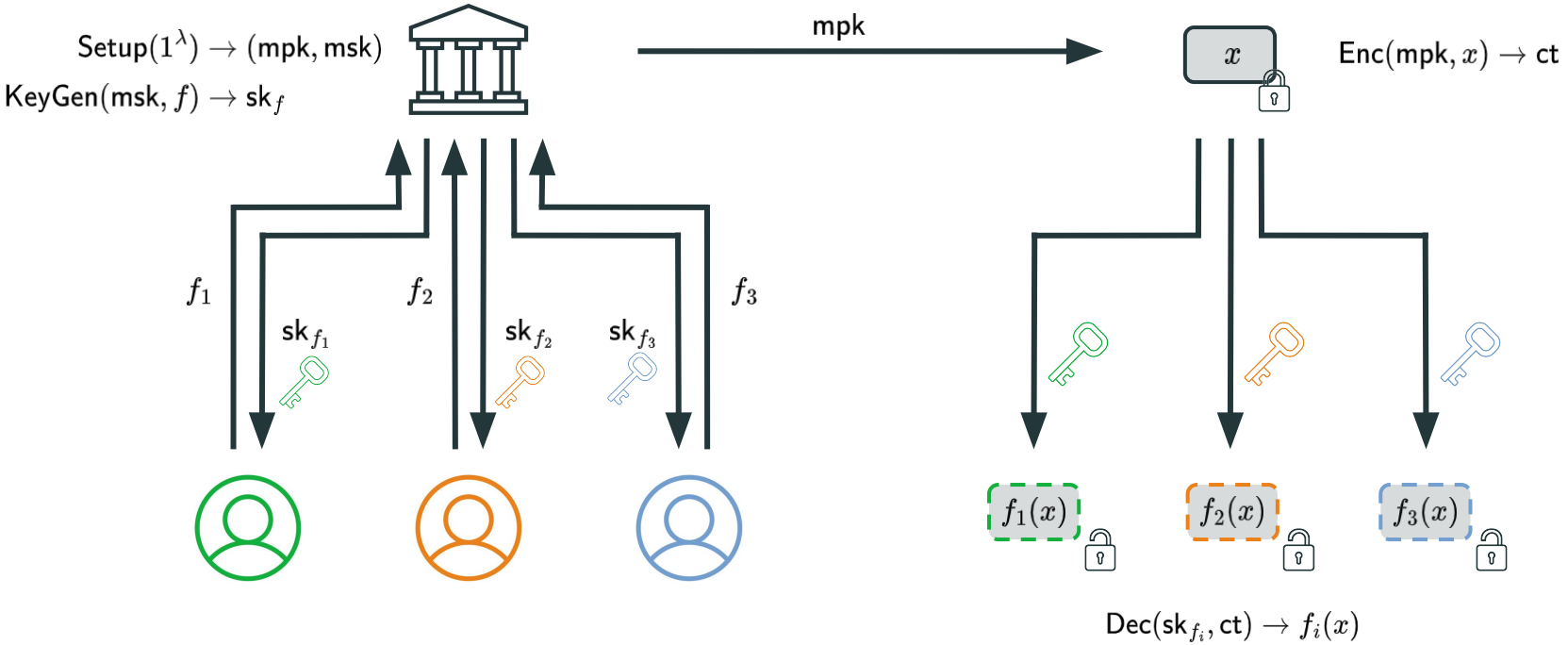
Part I. Removing the Trusted Authority (Registered ABE and FE)

- registered ABE via general paradigm [AC:PS25a]
- more [PS25b, PST25]

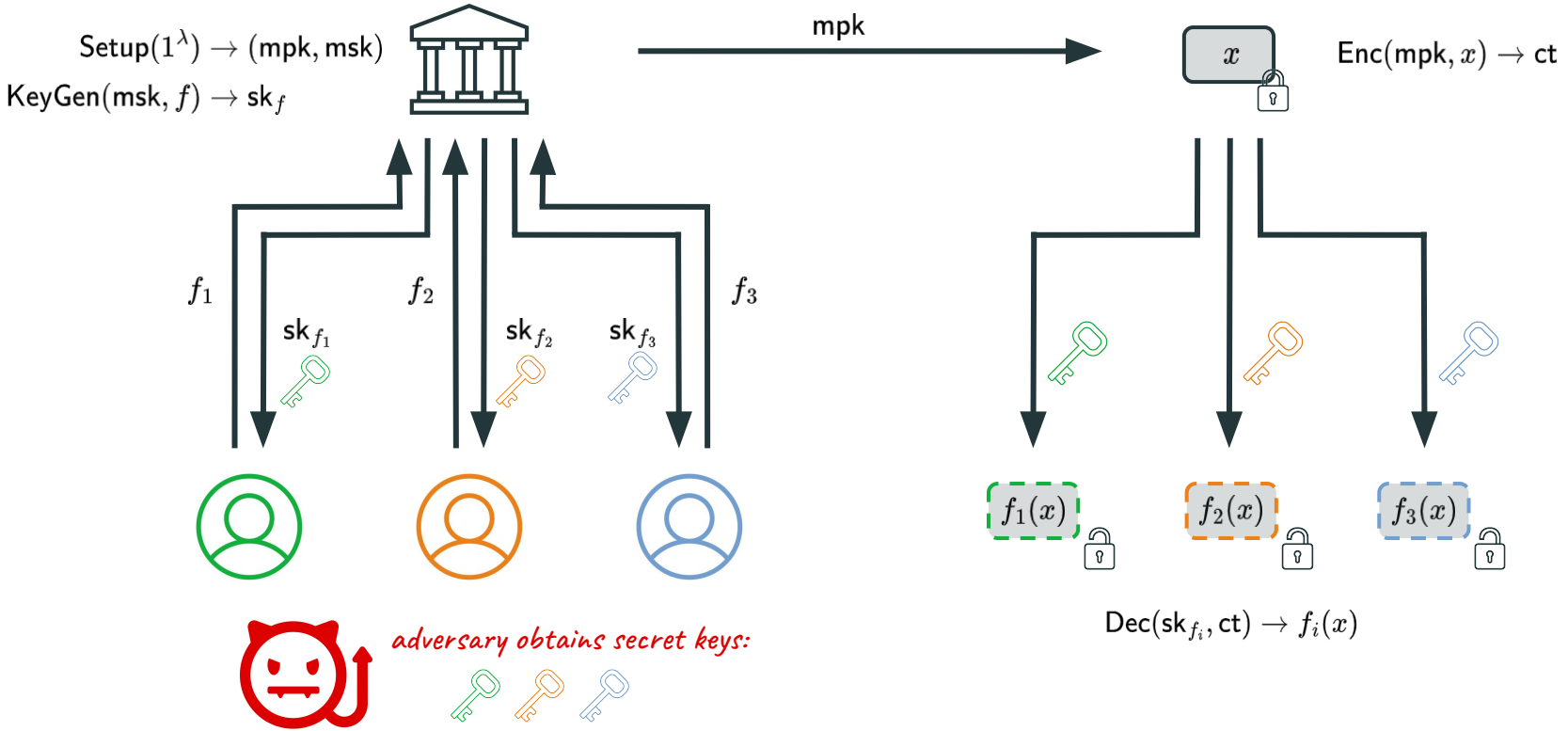
Part II. Supporting Multiple Encryptors (Multi-Client ABE)

- formal definition, first instantiations [TCC:PS24]
- more functionalities, different assumptions [PKC:S25]

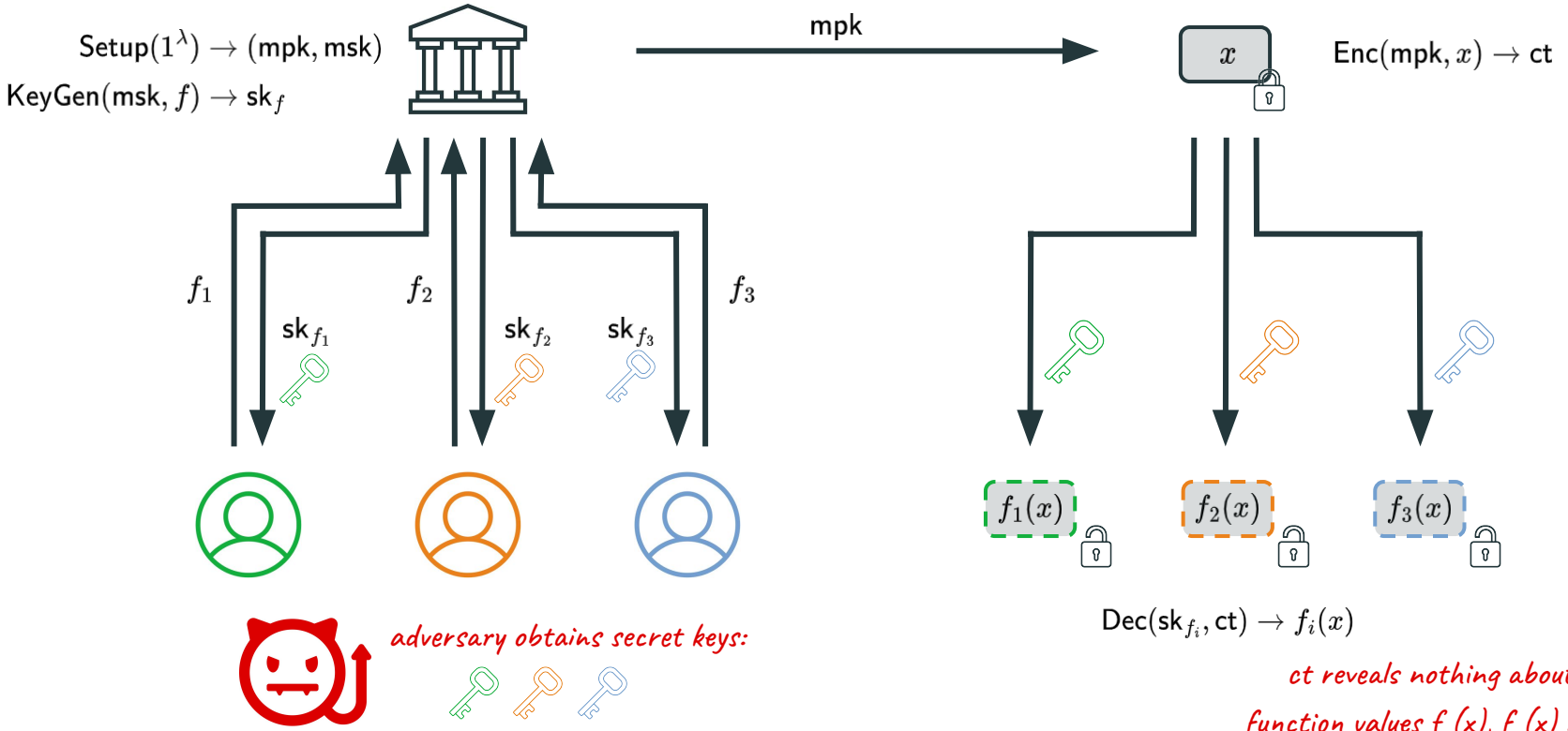
The Problem with FE Security



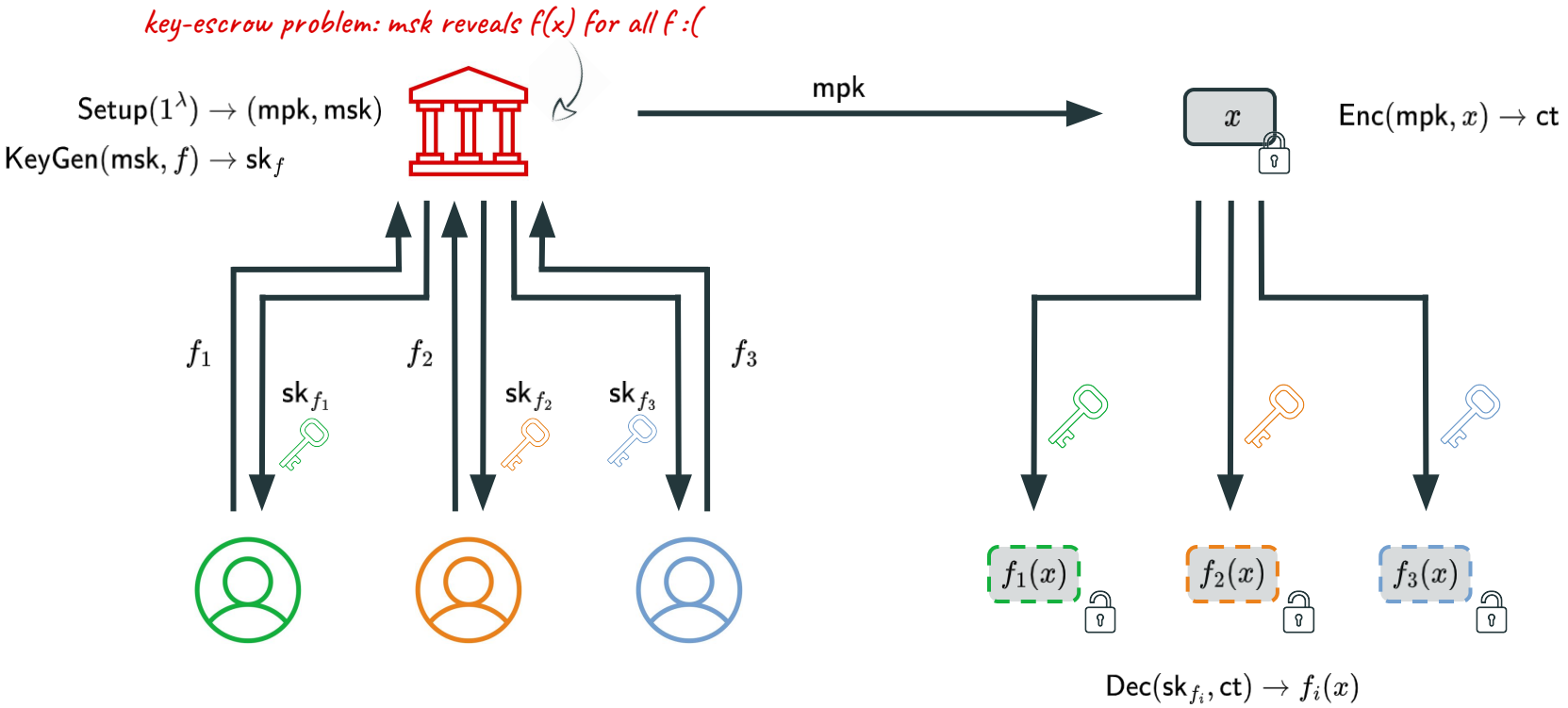
The Problem with FE Security



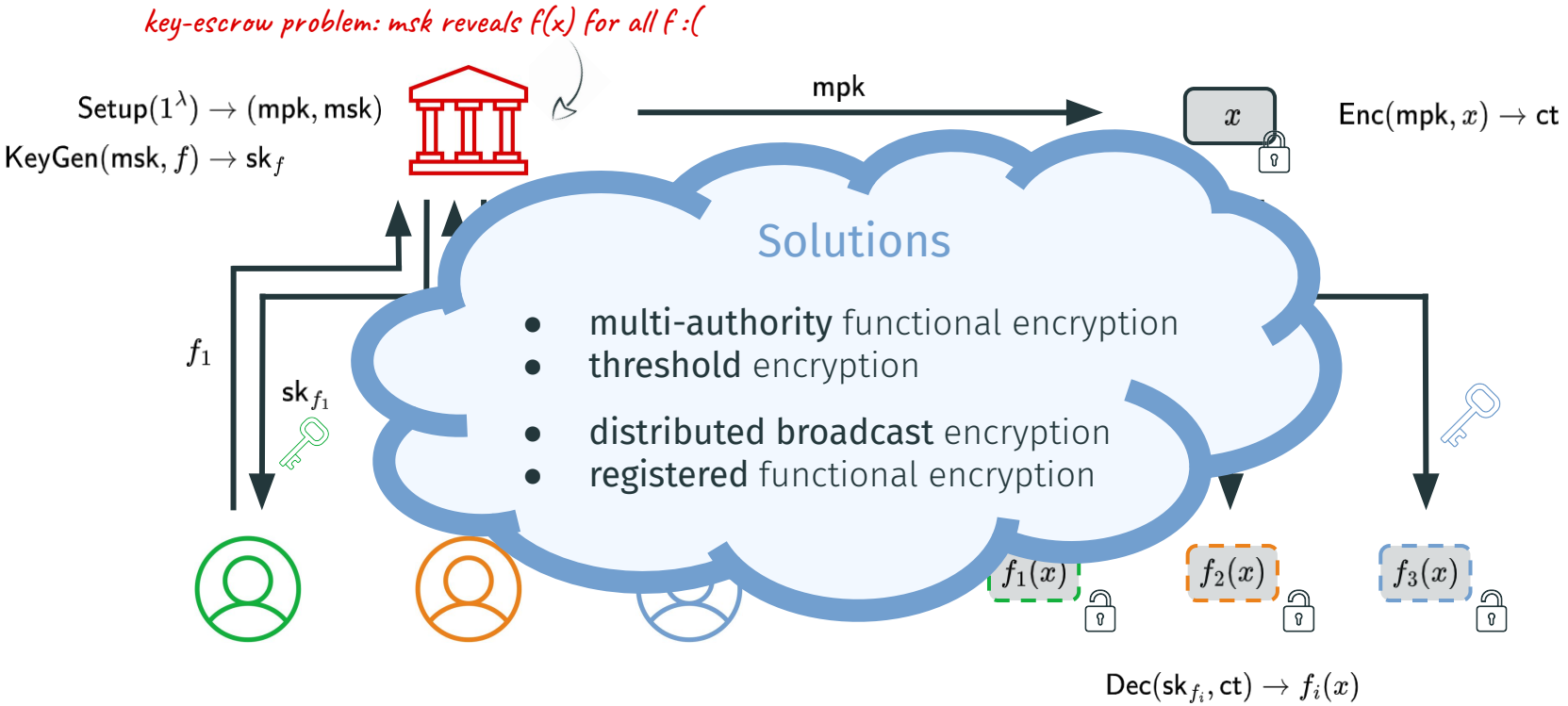
The Problem with FE Security



The Problem with FE Security



The Problem with FE Security



Registered Functional Encryption* [FFM+23]

$\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$

$\mathsf{pk}_1, \mathsf{sk}_1$

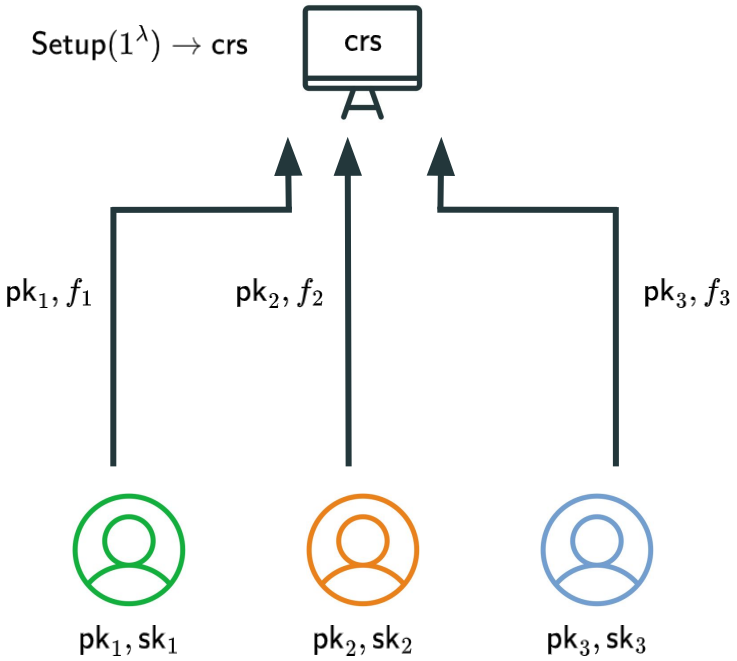
$\mathsf{pk}_2, \mathsf{sk}_2$

$\mathsf{pk}_3, \mathsf{sk}_3$

$\mathsf{KeyGen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$

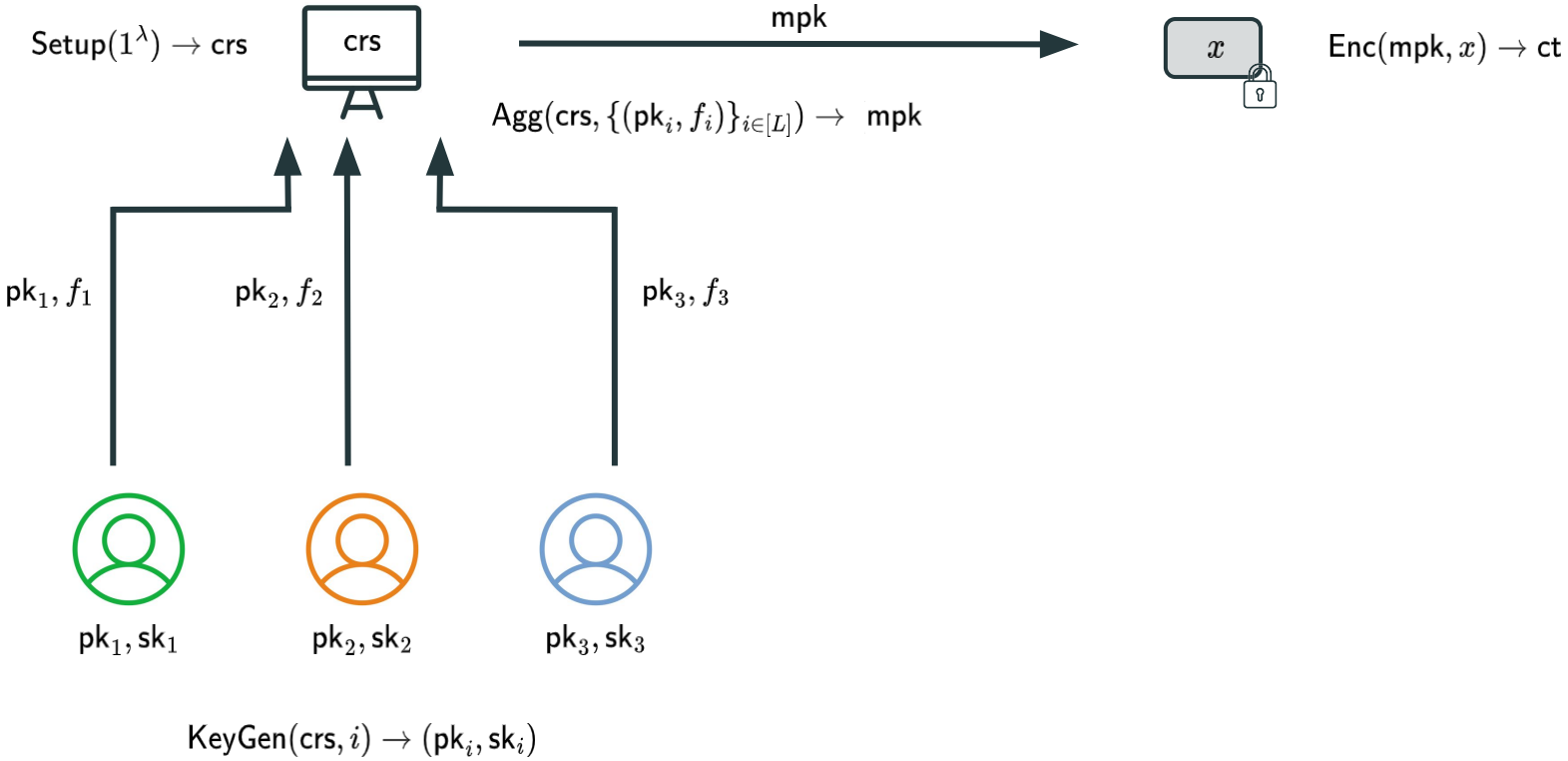
* not the full story, but good enough for now

Registered Functional Encryption* [FFM+23]



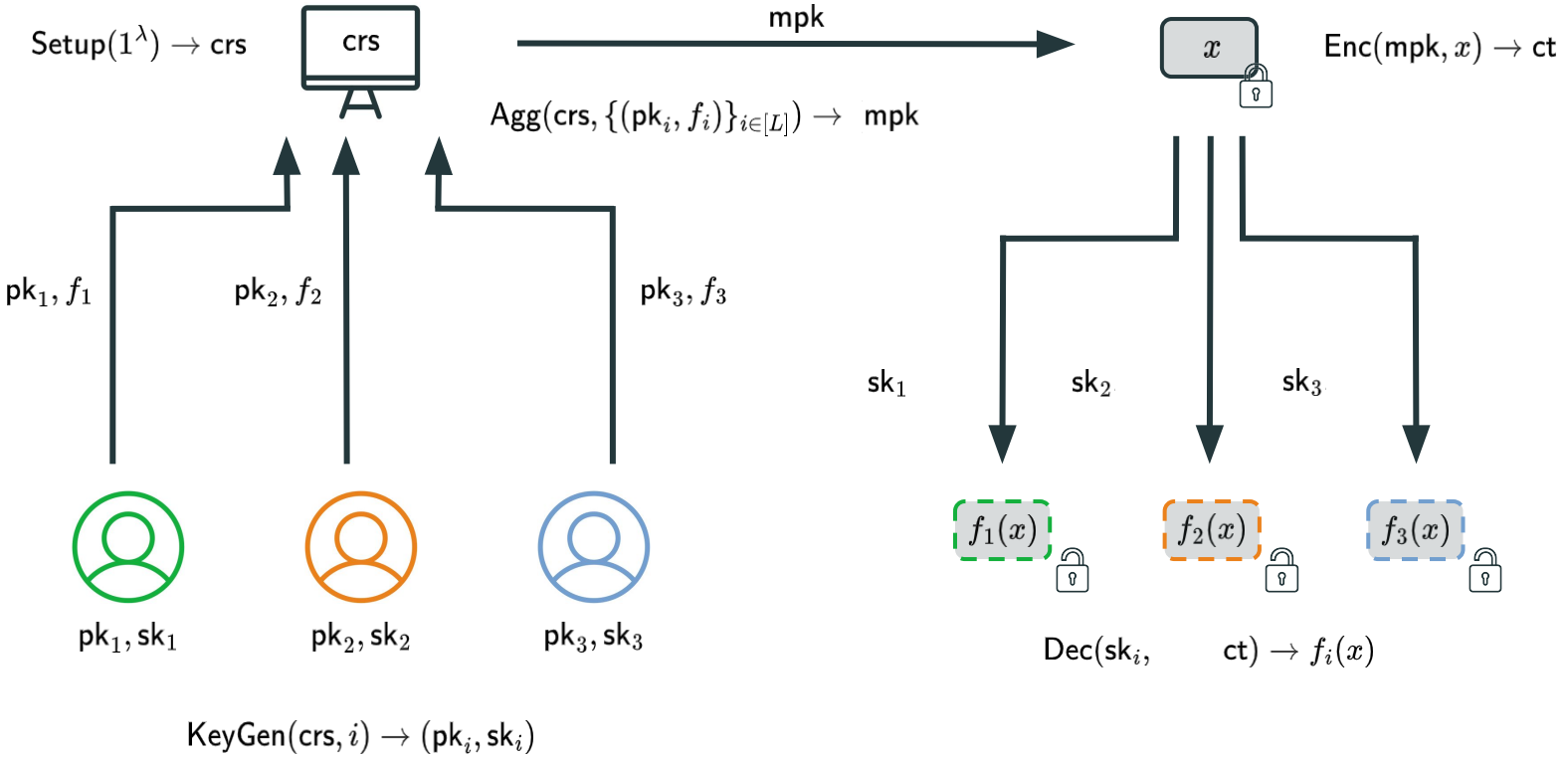
* not the full story, but good enough for now

Registered Functional Encryption* [FFM+23]



* not the full story, but good enough for now

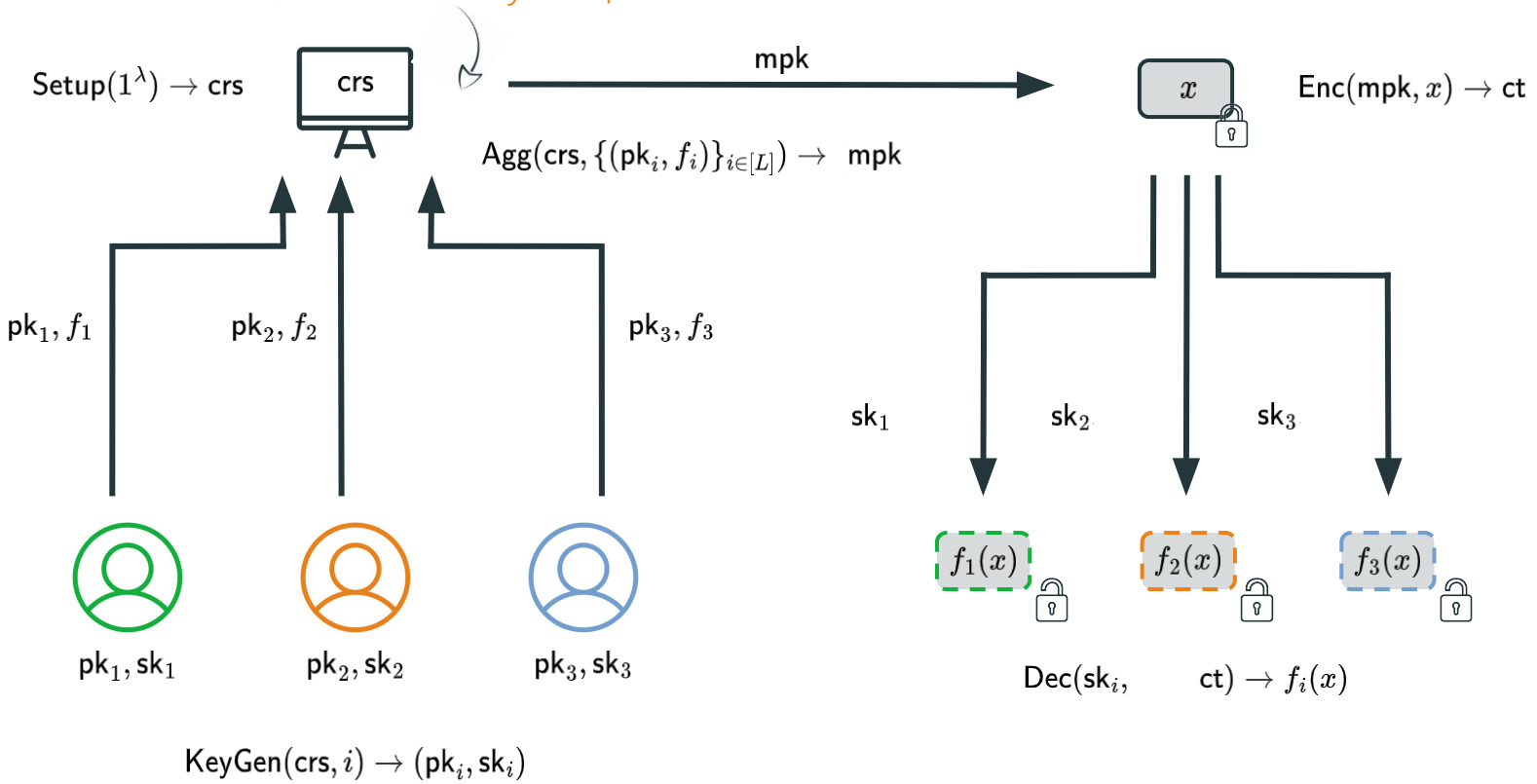
Registered Functional Encryption* [FFM+23]



* not the full story, but good enough for now

Registered Functional Encryption* [FFM+23]

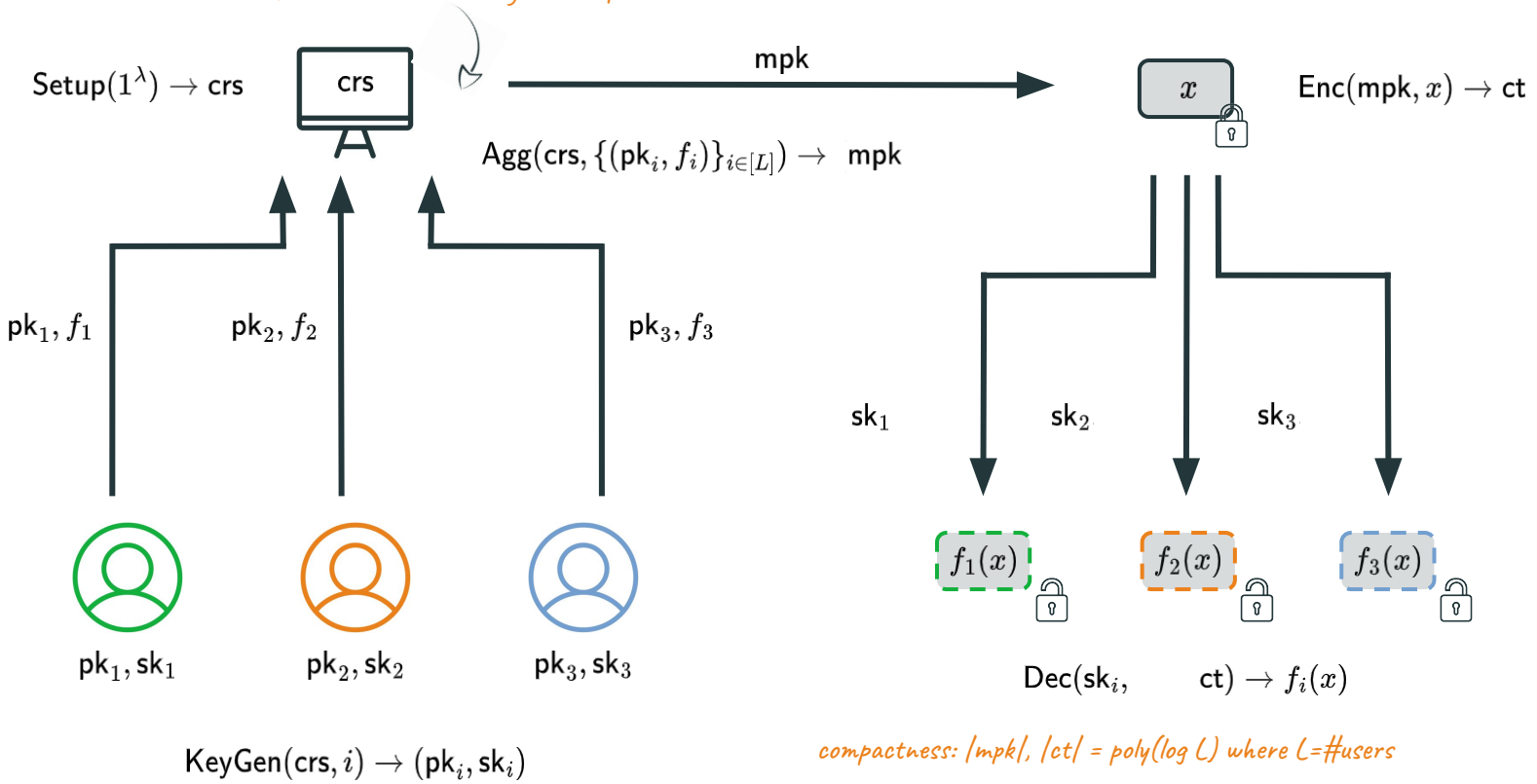
key curator is deterministic & holds no secret => key-escrow problem resolved!



* *not the full story, but good enough for now*

Registered Functional Encryption* [FFM+23]

key curator is deterministic & holds no secret => key-escrow problem resolved!

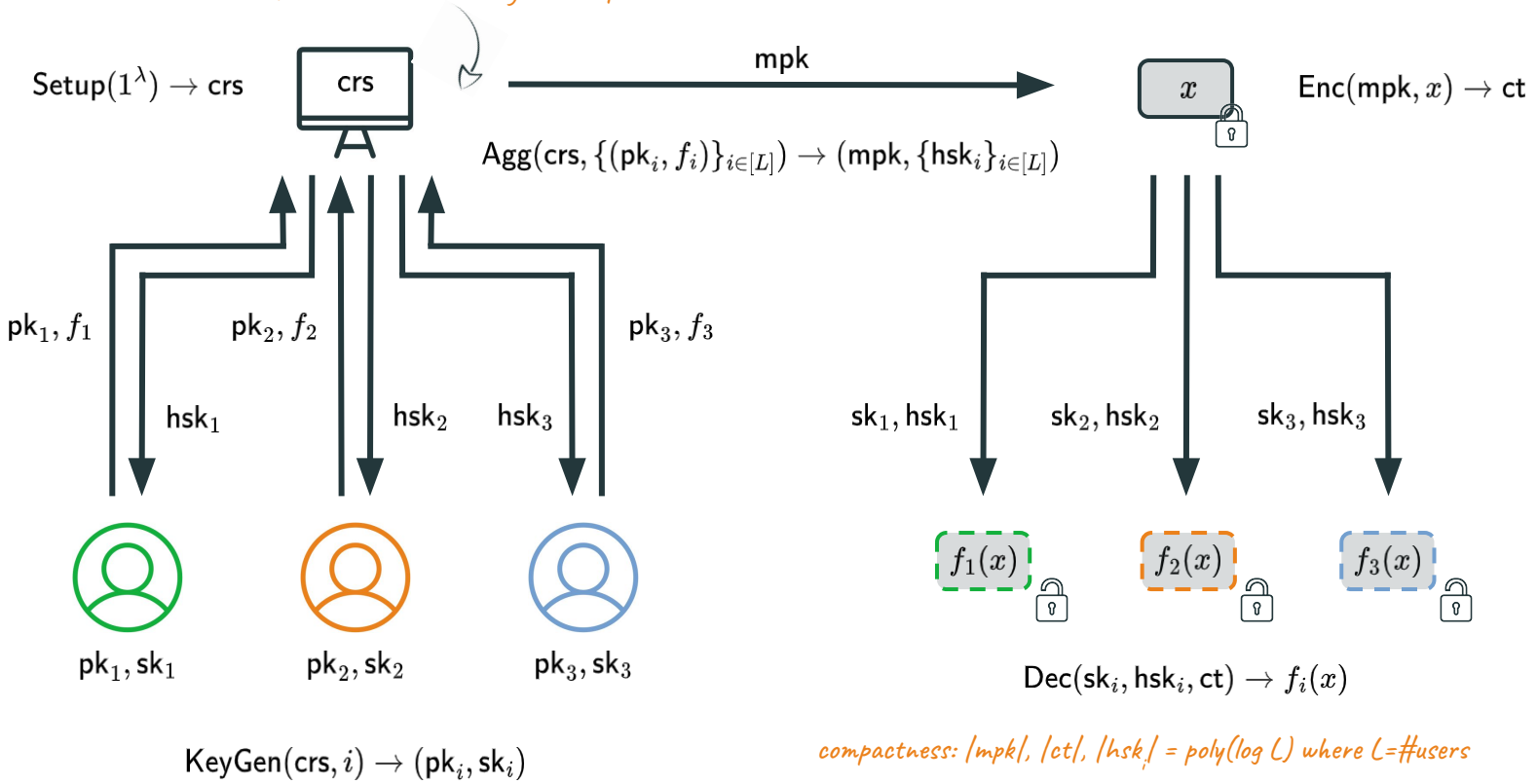


compactness: |mpk|, |ct| = poly(log L) where L=#users

* *not the full story, but good enough for now*

Registered Functional Encryption* [FFM+23]

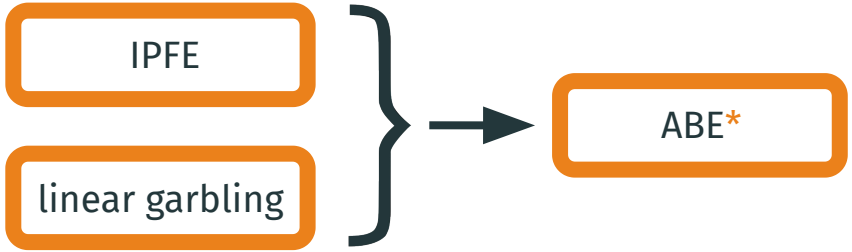
key curator is deterministic & holds no secret => key-escrow problem resolved!



compactness: |mpk|, |ct|, |hsk| = poly(log L) where L=#users

* *not the full story, but good enough for now*

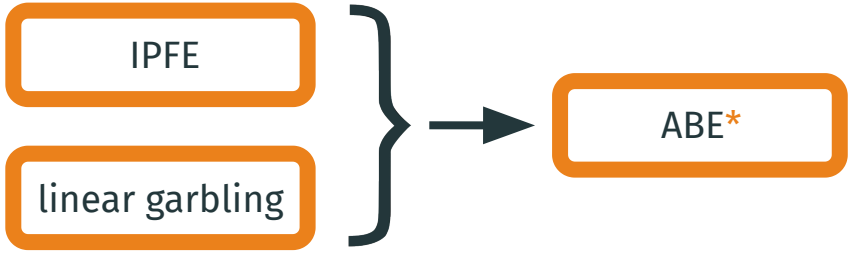
State of the Art. ABE ⇔ Registered ABE



* *natural generalization to FE*

- (Plain) ABE and FE.
  - ✔ **modular** – easy-to-verify building blocks
  - ✔ **powerful** – uniform models of computation, partially-hiding FE
  - ✔ **versatile** – flexible assumptions on different structures (pairings, lattices)

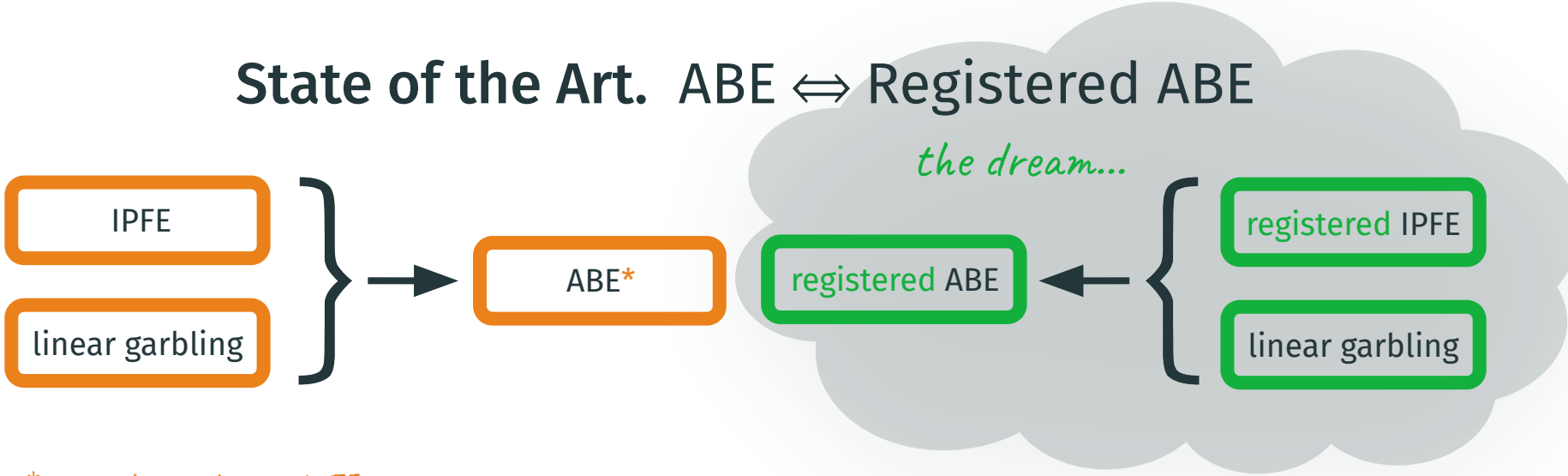
State of the Art. ABE ⇔ Registered ABE



* *natural generalization to FE*

- (Plain) ABE and FE.
  - ✔ **modular** – easy-to-verify building blocks
  - ✔ **powerful** – uniform models of computation, partially-hiding FE
  - ✔ **versatile** – flexible assumptions on different structures (pairings, lattices)
- Registered ABE and FE.
  - ✘ complex constructions, hard to verify
  - ✘ limited flexibility (few concrete functionalities and assumptions)

State of the Art. ABE ⇔ Registered ABE



* natural generalization to FE

- (Plain) ABE and FE.
  - ✔ **modular** – easy-to-verify building blocks
  - ✔ **powerful** – uniform models of computation, partially-hiding FE
  - ✔ **versatile** – flexible assumptions on different structures (pairings, lattices)
- Registered ABE and FE.
  - ✘ complex constructions, hard to verify
  - ✘ limited flexibility (few concrete functionalities and assumptions)

Linear Garbling [AIK11, IW14, LL20] *(Generalization of LSSS)*

1. $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
   - low-degree (affine) functions in *public* input $\mathbf{x}$ ("label functions")
   - coefficient vectors $(\mathbf{L}_1, \ldots, \mathbf{L}_m)$ encode *secret* input σ and randomness $\mathbf{r}$

2. $\ell_1 = L_1(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = L_m(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$
   - $\ell_1, \ldots, \ell_m$ ("labels")

Linear Garbling [AIK11, IW14, LL20] *(Generalization of LSSS)*

1. $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
   - low-degree (affine) functions in *public* input $\mathbf{x}$ ("label functions")
   - coefficient vectors $(\mathbf{L}_1, \ldots, \mathbf{L}_m)$ encode *secret* input σ and randomness $\mathbf{r}$

2. $\ell_1 = L_1(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = L_m(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$
   - $\ell_1, \ldots, \ell_m$ ("labels")

3. $\mathsf{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_m) \to \sigma \cdot f(\mathbf{x})$
   - high degree in $\mathbf{x}$
   - security: $\ell_1, \ldots, \ell_m$ reveal nothing about σ beyond $\sigma \cdot f(\mathbf{x})$

Linear Garbling [AIK11, IW14, LL20] *(Generalization of LSSS)*

1. $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
   - low-degree (affine) functions in *public* input $\mathbf{x}$ ("label functions")
   - coefficient vectors $(\mathbf{L}_1, \ldots, \mathbf{L}_m)$ encode *secret* input σ and randomness $\mathbf{r}$

Hide σ and r? → Hide these multiplications!

2. $\ell_1 = L_1(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = L_m(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$
   - $\ell_1, \ldots, \ell_m$ ("labels")

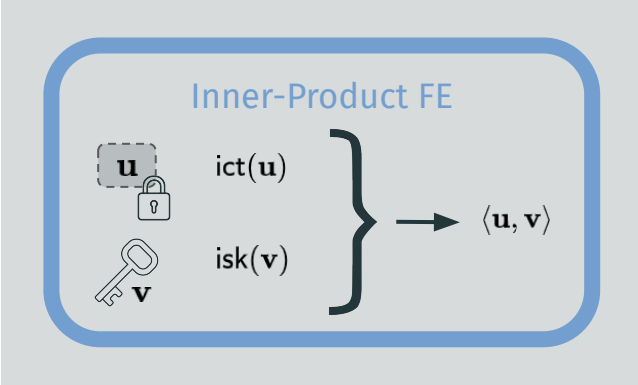
3. $\mathsf{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_m) \to \sigma \cdot f(\mathbf{x})$
   - high degree in $\mathbf{x}$
   - security: $\ell_1, \ldots, \ell_m$ reveal nothing about σ beyond $\sigma \cdot f(\mathbf{x})$

Linear Garbling [AIK11, IW14, LL20] *(Generalization of LSSS)*

1. $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
   - low-degree (affine) functions in *public* input $\mathbf{x}$ ("label functions")
   - coefficient vectors $(\mathbf{L}_1, \ldots, \mathbf{L}_m)$ encode *secret* input σ and randomness $\mathbf{r}$

Hide σ and r? → Hide these multiplications!

2. $\ell_1 = L_1(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = L_m(\mathbf{x}) = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$
   - $\ell_1, \ldots, \ell_m$ ("labels")



3. $\mathsf{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_m) \to \sigma \cdot f(\mathbf{x})$
   - high degree in $\mathbf{x}$
   - security: $\ell_1, \ldots, \ell_m$ reveal nothing about σ beyond $\sigma \cdot f(\mathbf{x})$

General Paradigm. ABE $\Leftarrow$ IPFE $\circ$ Garbling

sk_f  $\mathsf{isk}(\mathbf{L}_1), \ \ldots, \ \mathsf{isk}(\mathbf{L}_m)$ *(w.r.t. σ)*

$\mathsf{ct}_{\mathbf{x}}$  $\mathsf{ict}(1, \mathbf{x}), \ \sigma \oplus \mu$

General Paradigm. ABE $\Leftarrow$ IPFE $\circ$ Garbling

sk_f 

$\mathsf{isk}(\mathbf{L}_1), \ \ldots, \ \mathsf{isk}(\mathbf{L}_m)$ *(w.r.t. σ)*

$\mathsf{ct}_{\mathbf{x}}$ 

$\mathsf{ict}(1, \mathbf{x}), \ \ \sigma \oplus \mu$

$\longrightarrow \ \ell_1 = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$

General Paradigm. ABE ⇐ IPFE ∘ Garbling

sk_f 

$\mathsf{isk}(\mathbf{L}_1), \ \ldots, \ \mathsf{isk}(\mathbf{L}_m)$ *(w.r.t. σ)*

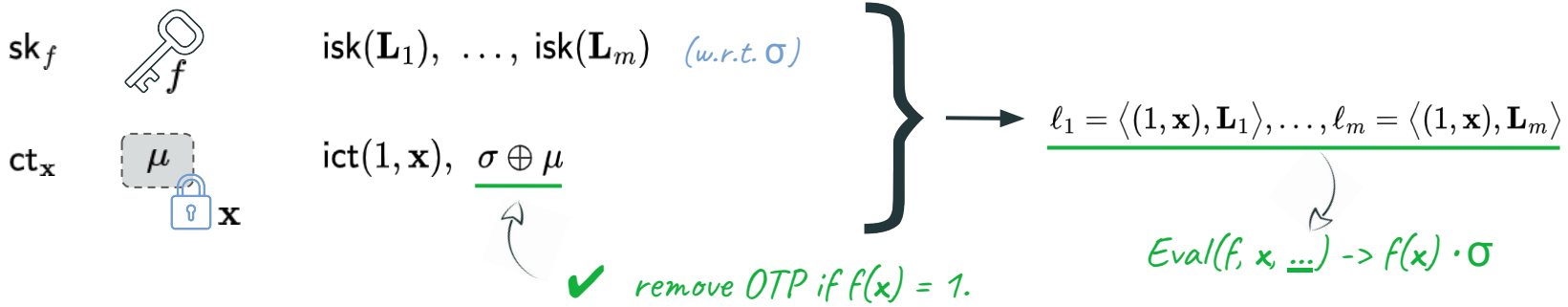
$\mathsf{ct}_{\mathbf{x}}$ 

$\mathsf{ict}(1, \mathbf{x}), \ \sigma \oplus \mu$

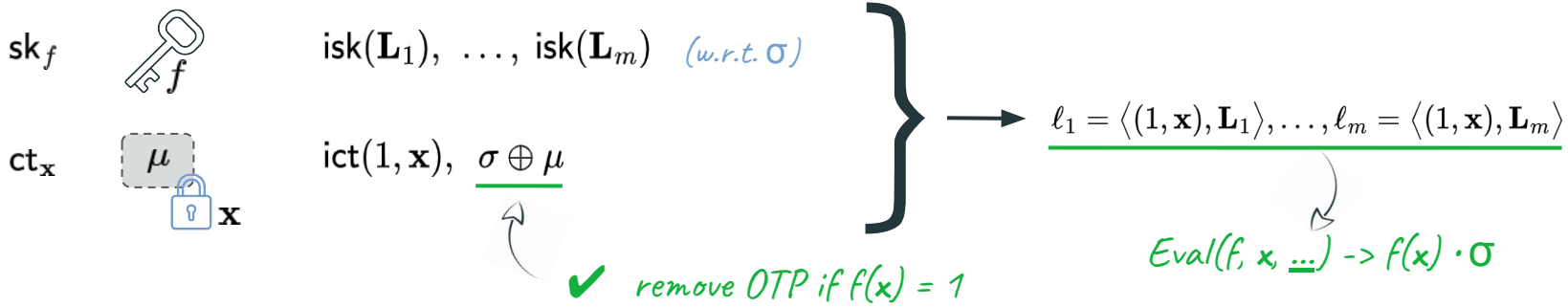
→ $\ell_1 = \langle (1, \mathbf{x}), \mathbf{L}_1 \rangle, \ldots, \ell_m = \langle (1, \mathbf{x}), \mathbf{L}_m \rangle$

Eval(f, x, ...) -> f(x) · σ

General Paradigm. ABE ⇐ IPFE ∘ Garbling



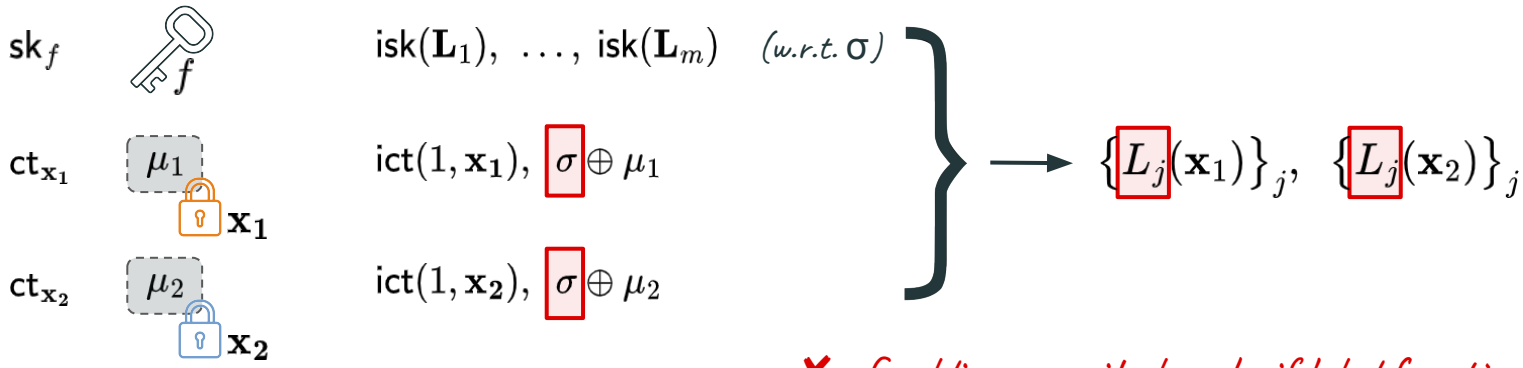
General Paradigm. ABE ⇐ IPFE ∘ Garbling



One-Time Security

1. IPFE ↛ only labels revealed
2. garbling ↛ only $\sigma \cdot f(\mathbf{x})$ revealed
3. σ is OTP for μ when $f(\mathbf{x}) = 0$

General Paradigm. ABE ⇐ IPFE ∘ Garbling

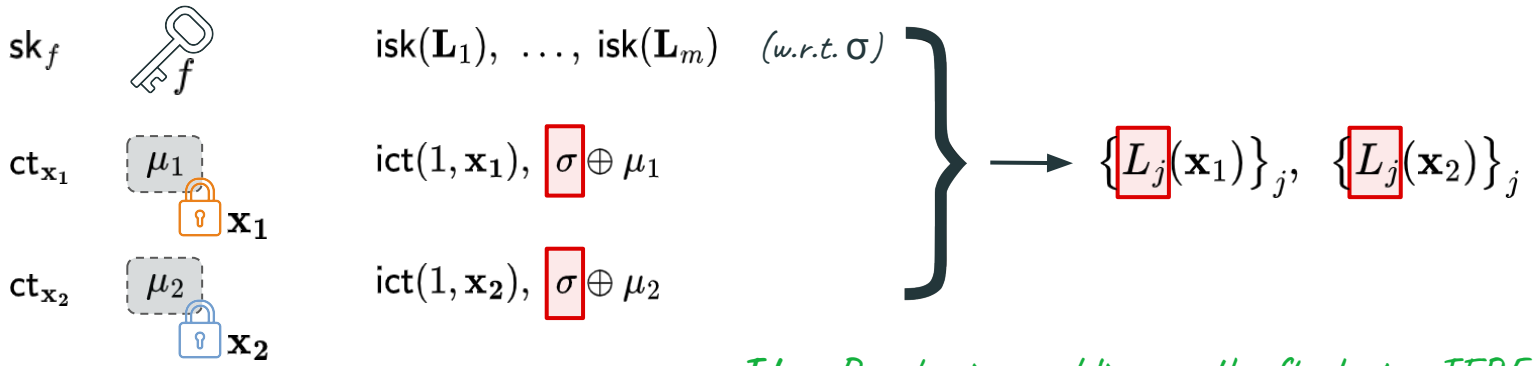


✘ *Garbling security breaks if label functions are reused!*

One-Time Security

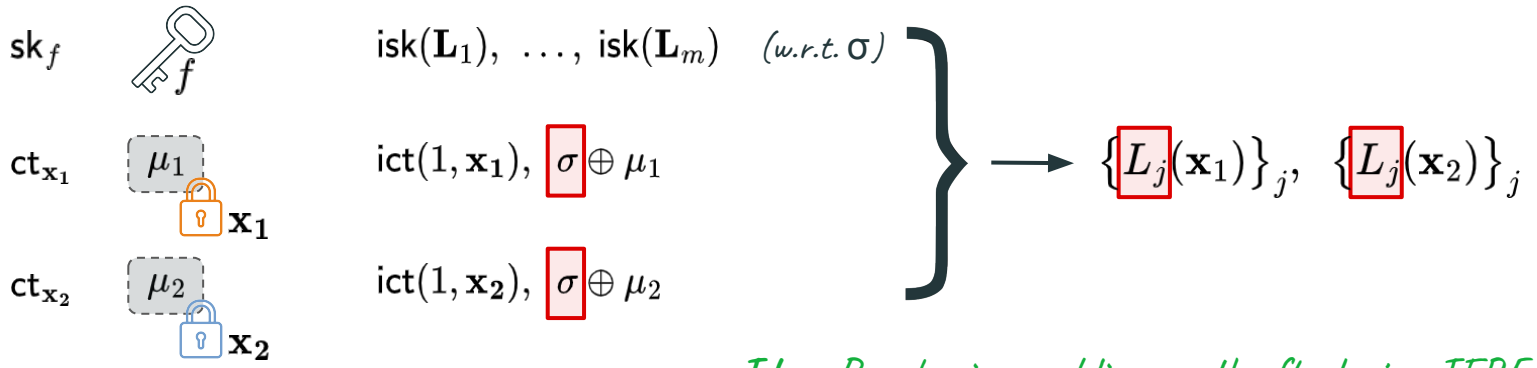
1. IPFE ↛ only labels revealed
2. garbling ↛ only $\sigma \cdot f(\mathbf{x})$ revealed
3. σ is OTP for μ when $f(\mathbf{x}) = 0$

General Paradigm. ABE ⇐ IPFE ∘ Garbling



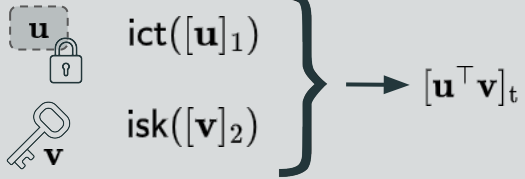
Idea. Randomize garbling on the fly during IFPE decryption.

General Paradigm. ABE ⇐ IPFE ∘ Garbling

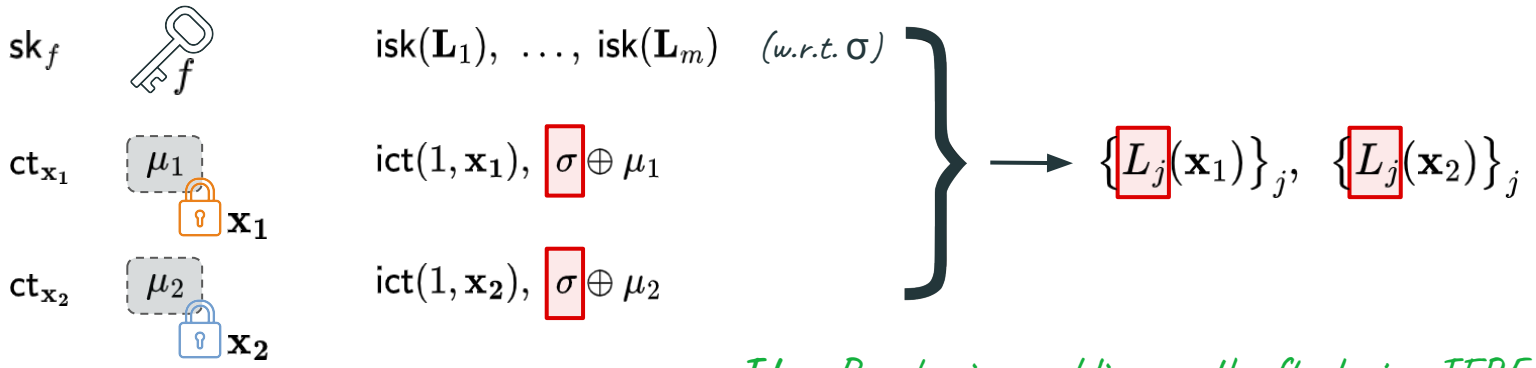


Idea. Randomize garbling on the fly during IFPE decryption.

Pairing-Based IPFE

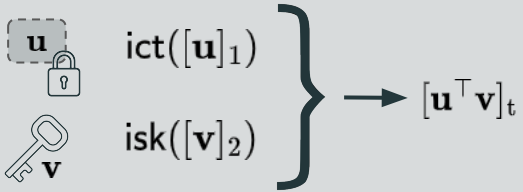


General Paradigm. ABE ⇐ IPFE ∘ Garbling



Idea. Randomize garbling on the fly during IFPE decryption.

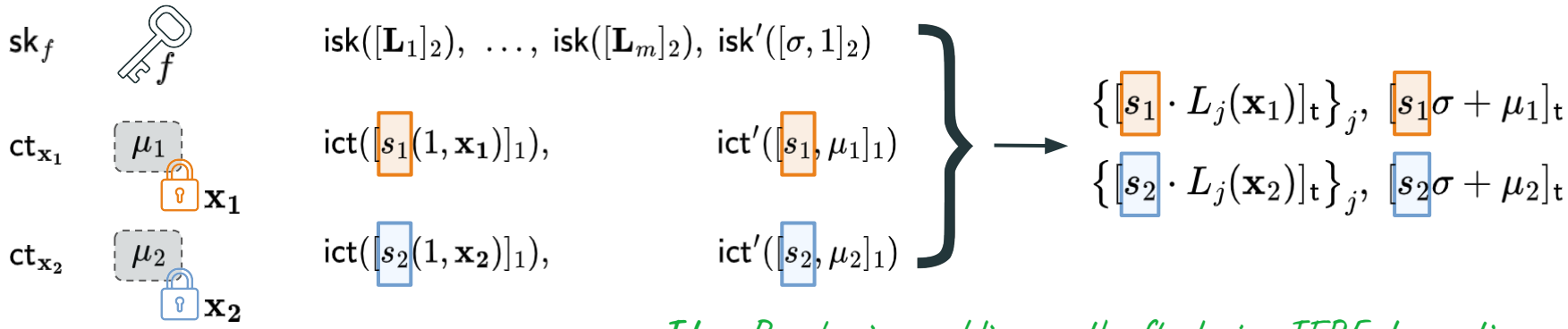
Pairing-Based IPFE



Linearity Properties of Linear Garbling

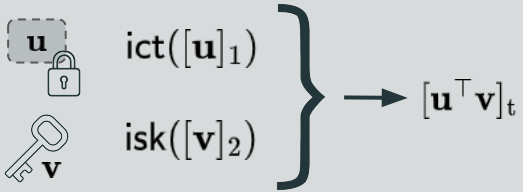
- $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
  linear in $(\sigma, \mathbf{r})$

General Paradigm. ABE $\Leftarrow$ IPFE $\circ$ Garbling



Idea. Randomize garbling on the fly during IFPE decryption.

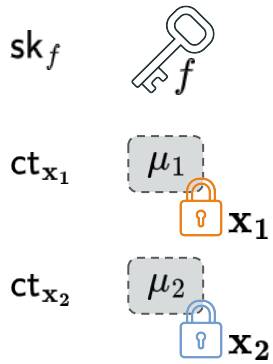
Pairing-Based IPFE



Linearity Properties of Linear Garbling

- $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
  linear in $(\sigma, \mathbf{r})$

General Paradigm. ABE ⇐ IPFE ∘ Garbling



$\mathsf{isk}([\mathbf{L}_1]_2),\ \ldots,\ \mathsf{isk}([\mathbf{L}_m]_2),\ \mathsf{isk}'([\sigma, 1]_2)$

$\mathsf{ict}([s_1(1, \mathbf{x}_1)]_1), \quad \mathsf{ict}'([s_1, \mu_1]_1)$

$\mathsf{ict}([s_2(1, \mathbf{x}_2)]_1), \quad \mathsf{ict}'([s_2, \mu_2]_1)$

→

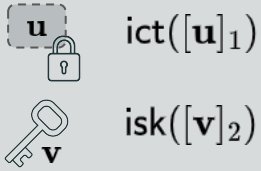
Eval can be run in the exponent.

$$\left\{[s_1 \cdot L_j(\mathbf{x}_1)]_t\right\}_j,\ [s_1\sigma + \mu_1]_t$$

$$\left\{[s_2 \cdot L_j(\mathbf{x}_2)]_t\right\}_j,\ [s_2\sigma + \mu_2]_t$$

Idea. Randomize garbling on the fly during IFPE decryption.

Pairing-Based IPFE



$\mathsf{ict}([\mathbf{u}]_1)$

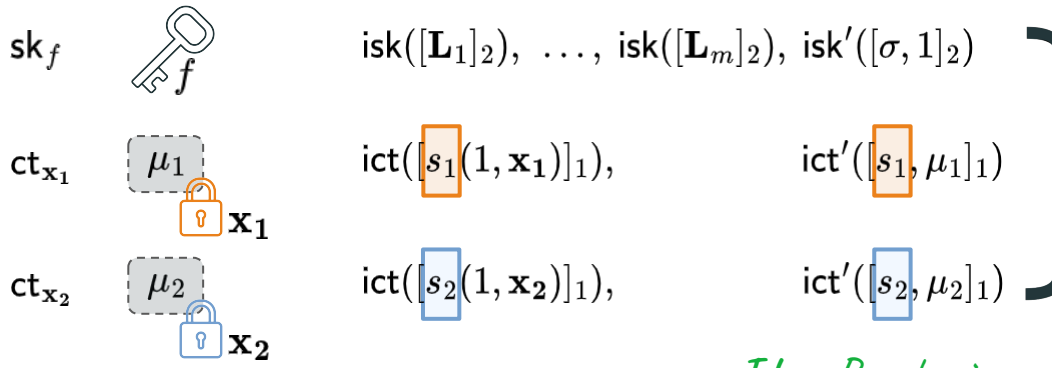
$\mathsf{isk}([\mathbf{v}]_2)$

→ $[\mathbf{u}^\top \mathbf{v}]_t$

Linearity Properties of Linear Garbling

- $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
  linear in $(\sigma, \mathbf{r})$
- $\mathsf{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_m) \to \sigma \cdot f(\mathbf{x})$
  linear in $\ell_1, \ldots, \ell_m$

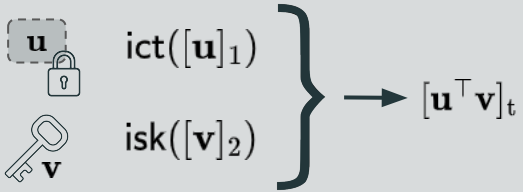
General Paradigm. ABE ⇐ IPFE ∘ Garbling



Garbling looks fresh under DDH.

Idea. Randomize garbling on the fly during IFPE decryption.

Pairing-Based IPFE



Linearity Properties of Linear Garbling

- $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to (L_1, \ldots, L_m)$
  linear in $(\sigma, \mathbf{r})$
- $\mathsf{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_m) \to \sigma \cdot f(\mathbf{x})$
  linear in $\ell_1, \ldots, \ell_m$

General Paradigm. Reg-ABE $\Leftarrow$ Reg-IPFE $\circ$ Garbling

General Paradigm. Reg-ABE ⇐ Reg-IPFE ∘ Garbling

Challenges in the registered setting.

(1) Registered IPFE supporting registration of vectors over group unknown

General Paradigm. Reg-ABE ⇐ Reg-IPFE ∘ Garbling

Challenges in the registered setting.

(1) Registered IPFE supporting registration of vectors over group unknown

(2) Sampling of user-specific randomness

- key generation performed by (potentially **malicious**) users
- aggregation is **deterministic**
- encryption time is polylogarithmic in number of users (**compactness**)

General Paradigm. Reg-ABE ⟸ Reg-IPFE ∘ Garbling

Challenges in the registered setting.

(1) Registered IPFE supporting registration of vectors over group unknown

(2) Sampling of user-specific randomness

- key generation performed by (potentially **malicious**) users
- aggregation is **deterministic**
- encryption time is polylogarithmic in number of users (**compactness**)
- setup performed before user functions are known ⇝ decompose garbling procedure

Linearity to the Rescue

- divide garbling algorithm into two phases
  a. probabilistic offline phase: sample $(\sigma, \mathbf{r})$
  b. deterministic online phase: compute matrix $\widehat{\mathbf{L}} = (\widehat{\mathbf{L}}_1 \| \dots \| \widehat{\mathbf{L}}_m)$ s.t. $\mathbf{L}_i = \left(\mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}_i$

Linearity Properties of Linear Garbling

- $\mathsf{Garble}(f, \sigma; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \dots \| \mathbf{L}_m)$ linear in $(\sigma, \mathbf{r})$
- $\boldsymbol{\ell} = (\ell_1, \dots, \ell_m) = (1, \mathbf{x}) \cdot \mathbf{L}$ affine in $\mathbf{x}$

Linearity to the Rescue

- divide garbling algorithm into two phases
  a. probabilistic offline phase: sample$(\sigma, \mathbf{r})$
  b. deterministic online phase: compute matrix $\widehat{\mathbf{L}} = \left(\widehat{\mathbf{L}}_1 \| \ldots \| \widehat{\mathbf{L}}_m\right)$ s.t. $\mathbf{L}_i = \left(\mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}_i$

$$\boldsymbol{\ell} = (1, \mathbf{x}) \cdot \mathbf{L} = (1, \mathbf{x}) \cdot \left(\mathbf{I}_{1+\mathbf{x}} \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}} = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$$

Linearity Properties of Linear Garbling

- Garble$(f, \sigma; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \ldots \| \mathbf{L}_m)$ linear in $(\sigma, \mathbf{r})$
- $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m) = (1, \mathbf{x}) \cdot \mathbf{L}$ affine in $\mathbf{x}$

Linearity to the Rescue

- divide garbling algorithm into two phases
  a. probabilistic offline phase: sample$(\sigma, \mathbf{r})$
  b. deterministic online phase: compute matrix $\widehat{\mathbf{L}} = \left(\widehat{\mathbf{L}}_1 \| \dots \| \widehat{\mathbf{L}}_m\right)$ s.t. $\mathbf{L}_i = \left(\mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}_i$

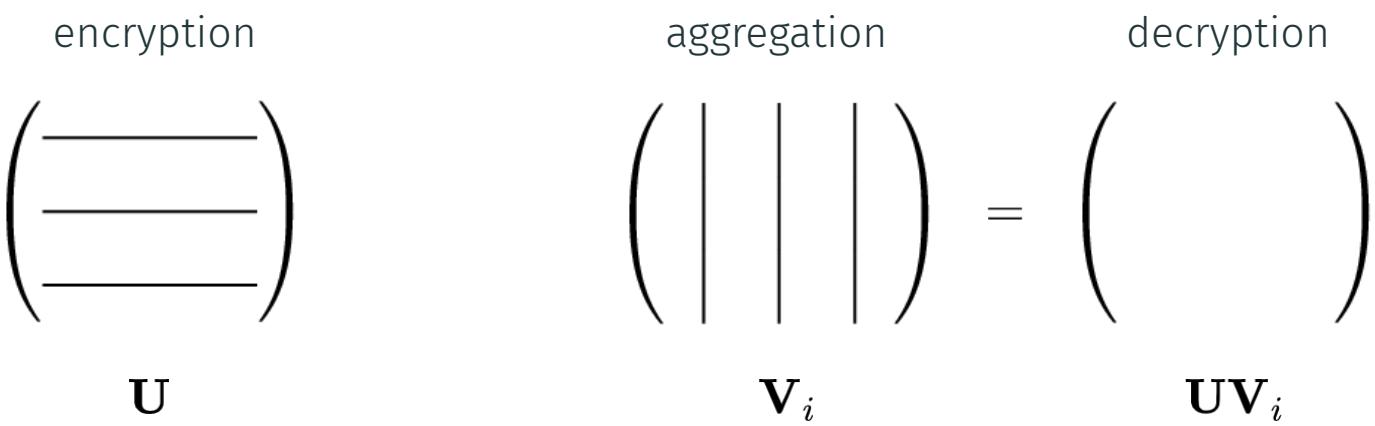
$$\boldsymbol{\ell} = (1, \mathbf{x}) \cdot \mathbf{L} = (1, \mathbf{x}) \cdot \left(\mathbf{I}_{1+\mathbf{x}} \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}} = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$$

- run offline phase during setup, online phase during aggregation
  ⇝ we need **generalization of inner product functionality**

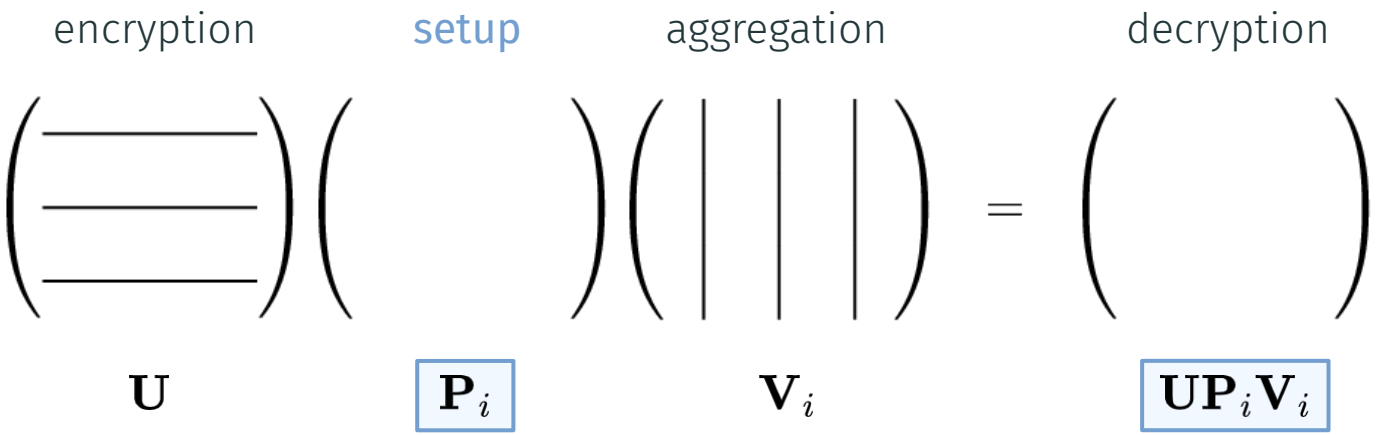
Linearity Properties of Linear Garbling

- Garble$(f, \sigma; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \dots \| \mathbf{L}_m)$ linear in $(\sigma, \mathbf{r})$
- $\boldsymbol{\ell} = (\ell_1, \dots, \ell_m) = (1, \mathbf{x}) \cdot \mathbf{L}$ affine in $\mathbf{x}$

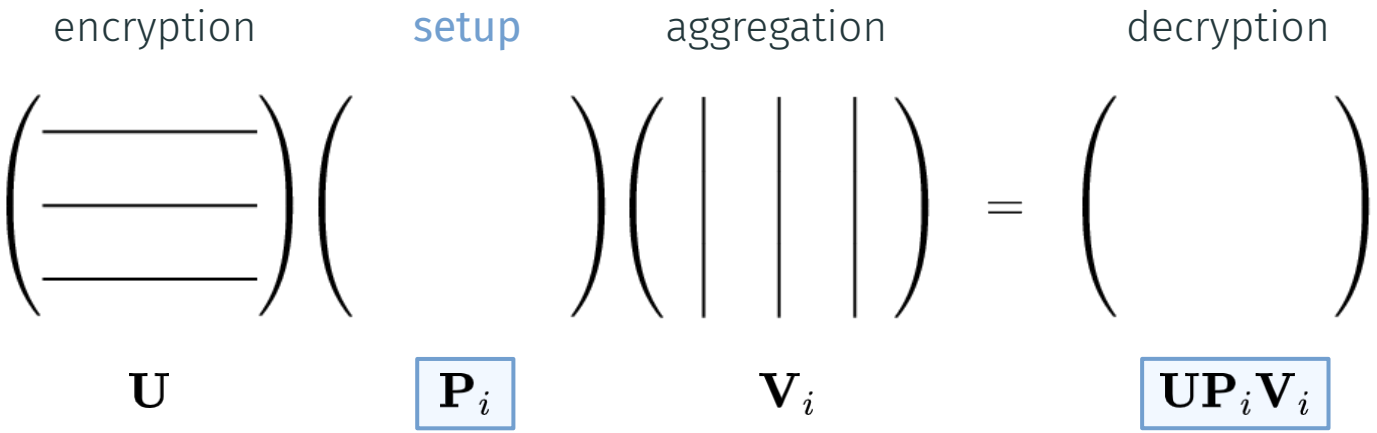
Reg-FE for IP (Batch Variant)



Reg-FE for Pre-IP (Batch Variant)



Reg-FE for Pre-IP (Batch Variant)



Theorem. Reg-FE for Pre-IP can be built from (bilateral) MDDH. [ZLGQ24, AC:PS25]

How to Pick the Matrices?

encryption setup aggregation decryption

$$\left(\mu, s, \left((1, \mathbf{x}) \otimes s\mathbf{I}_{1+|r|}\right)\right) \begin{pmatrix} 1 & \\ \sigma_i & \\ & \mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma_i, \mathbf{r}_i) \end{pmatrix} \begin{pmatrix} 1 & \\ & \widehat{\mathbf{L}}_i \end{pmatrix} = \left(\mu + s\sigma_i, \left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right)$$

$[\mathbf{U}]_1$ $[\mathbf{P}_i]_2$ $\mathbf{V}_i$ $[\mathbf{U}\mathbf{P}_i\mathbf{V}_i]_t$

Formula for Garbling Labels. $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m) = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$

How to Pick the Matrices?

encryption setup aggregation decryption

$$\left(\mu, s, \left((1, \mathbf{x}) \otimes s\mathbf{I}_{1+|r|}\right)\right) \begin{pmatrix} 1 & \\ \sigma_i & \\ & \mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma_i, \mathbf{r}_i) \end{pmatrix} \begin{pmatrix} 1 & \\ & \widehat{\mathbf{L}}_i \end{pmatrix} = \left(\mu + s\sigma_i, \left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right)$$

$[\mathbf{U}]_1$ $[\mathbf{P}_i]_2$ $\mathbf{V}_i$ $[\mathbf{U}\mathbf{P}_i\mathbf{V}_i]_t$

Correctness. *RIPFE decryption yields* $[\mu + s\sigma_i]_t$, $\left[\left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right]_t$

↳ *Eval(...) -> $f_i(x) \cdot s\sigma_i$*

Formula for Garbling Labels. $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m) = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$

How to Pick the Matrices?

encryption setup aggregation decryption

$$\left(\mu, s, \left((1, \mathbf{x}) \otimes s\mathbf{I}_{1+|r|}\right)\right) \begin{pmatrix} 1 & \\ \sigma_i & \\ & \mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma_i, \mathbf{r}_i) \end{pmatrix} \begin{pmatrix} 1 & \\ & \widehat{\mathbf{L}}_i \end{pmatrix} = \left(\mu + s\sigma_i, \left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right)$$

$[\mathbf{U}]_1$ $[\mathbf{P}_i]_2$ $\mathbf{V}_i$ $[\mathbf{U}\mathbf{P}_i\mathbf{V}_i]_t$

Security. *RIPFE leakage is* $[\mu + s\sigma_i]_t$, $\left[\left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right]_t$ (s ~ $)

↳ *indistinguishable from Sim(f, x, d <-$)*

Formula for Garbling Labels. $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m) = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$

How to Pick the Matrices?

encryption setup aggregation decryption

$$\left(\mu, s, \left((1, \mathbf{x}) \otimes s\mathbf{I}_{1+|r|}\right)\right) \begin{pmatrix} 1 & \\ \sigma_i & \\ & \mathbf{I}_{1+|\mathbf{x}|} \otimes (\sigma_i, \mathbf{r}_i) \end{pmatrix} \begin{pmatrix} 1 & \\ & \widehat{\mathbf{L}}_i \end{pmatrix} = \left(\mu + s\sigma_i, \left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right)$$

$[\mathbf{U}]_1$ $[\mathbf{P}_i]_2$ $\mathbf{V}_i$ $[\mathbf{U}\mathbf{P}_i\mathbf{V}_i]_t$

$[\mu + s\sigma_i]_t$, $\left[\left((1, \mathbf{x}) \otimes (s\sigma_i, s\mathbf{r}_i)\right) \cdot \widehat{\mathbf{L}}_i\right]_t$

$\$$ $\$$ $\$$

indistinguishable from Sim(f, x, d <-$)

What about Turing machines?

Problem: shape of L and r depends on input length, runtime and space

-> study concrete garbling schemes

Formula for Garbling Labels. $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m) = \left((1, \mathbf{x}) \otimes (\sigma, \mathbf{r})\right) \cdot \widehat{\mathbf{L}}$

Generalization to Registered FE

- so far, we used σ_1 = pad for fixed message μ (and σ_0 not used at all)

Linear Garbling

$\mathsf{Garble}(f, \sigma_0, \sigma_1; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \dots \| \mathbf{L}_m)$

$\mathsf{Eval}(f, \mathbf{x}, \ell := (1, \mathbf{x}) \cdot \mathbf{L}) \to d$ s.t. $d = \sigma_1 f(\mathbf{x}) + \sigma_0$

Generalization to Registered FE

- so far, we used σ_1 = pad for fixed message μ (and σ_0 not used at all)
- more general we can
  - encode data in σ_1
    ⇝ attribute-weighted sums functionalities [AGW20]
  - use σ_0 as masking term for other Reg-FE functionalities
    ⇝ attribute-based functionalities (AB-AWS, AB-QF)

Linear Garbling

$\mathsf{Garble}(f, \sigma_0, \sigma_1; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \dots \| \mathbf{L}_m)$

$\mathsf{Eval}(f, \mathbf{x}, \ell := (1, \mathbf{x}) \cdot \mathbf{L}) \to d$ s.t. $d = \sigma_1 f(\mathbf{x}) + \sigma_0$

Generalization to Registered FE

- so far, we used σ_1 = pad for fixed message μ (and σ_0 not used at all)
- more general we can
  - encode data in σ_1
    ⇝ attribute-weighted sums functionalities [AGW20]
  - use σ_0 as masking term for other Reg-FE functionalities
    ⇝ attribute-based functionalities (AB-AWS, AB-QF)
- this yields Reg-FE instantiations for almost all functionalities known for pairing-based FEs (exception: *unbounded* linear and quadratic functions [T23])

Linear Garbling

$\mathsf{Garble}(f, \sigma_0, \sigma_1; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \dots \| \mathbf{L}_m)$

$\mathsf{Eval}(f, \mathbf{x}, \ell := (1, \mathbf{x}) \cdot \mathbf{L}) \to d$ s.t. $d = \sigma_1 f(\mathbf{x}) + \sigma_0$

Part I. More Results

[PS25a]. (Generic) Construction from **Pairings**

- registered analogs of many pairing-based ABEs and FEs, e.g.
  - Reg-ABE for ABPs and logspace TMs (↔ [LL20])
  - Reg-FE for attribute-based quadratic functions (↔ [W20])
  - Reg-FE for (attribute-based) attribute-weighted sums (↔ [DP21, DPT22, ATY23])

Part I. More Results

[PS25a]. (Generic) Construction from **Pairings**

- registered analogs of many pairing-based ABEs and FEs, e.g.
  - Reg-ABE for ABPs and logspace TMs (↔ [LL20])
  - Reg-FE for attribute-based quadratic functions (↔ [W20])
  - Reg-FE for (attribute-based) attribute-weighted sums (↔ [DP21, DPT22, ATY23])

[PS25b]. (Concrete) Construction from **Pairings**

- adaptive security
- succinct ciphertexts, i.e. *ciphertext size* does not grow with *attribute length*

Part I. More Results

[PS25a]. (Generic) Construction from **Pairings**

- registered analogs of many pairing-based ABEs and FEs, e.g.
  - Reg-ABE for ABPs and logspace TMs (↔ [LL20])
  - Reg-FE for attribute-based quadratic functions (↔ [W20])
  - Reg-FE for (attribute-based) attribute-weighted sums (↔ [DP21, DPT22, ATY23])

[PS25b]. (Concrete) Construction from **Pairings**

- adaptive security
- succinct ciphertexts, i.e. *ciphertext size* does not grow with *attribute length*

[PST25]. Construction from **Lattices**

- (KP|CP) Reg-ABE and predicate encryption
  - ✔ unbounded depth circuits
  - ✔ general Turing machines
  - ✔ (almost) optimal asymptotic parameters
  - ✘ uses *pseudorandom* FE for specific samplers
    (can be instantiated from private-coin evasive LWE)

Part I. Open Questions

Pairing-Based. **Unbounded Reg-FE** (with Constant-Size CRS)

- **standard** assumptions: $|\mathsf{crs}| = O(L^2)$ (recall: L = #users)
- **non-standard** assumptions:

[GLWW24]	$\lvert\mathsf{crs}\rvert = O(L^{1.58})$	composite-order, q-type
[GHKKP25]	$\lvert\mathsf{crs}\rvert = O(L)$	prime-order, GGM

Part I. Open Questions

Pairing-Based. Unbounded Reg-FE (with Constant-Size CRS)

- **standard** assumptions: $|\mathsf{crs}| = O(L^2)$ (recall: L = #users)
- **non-standard** assumptions:
  - [GLWW24] $|\mathsf{crs}| = O(L^{1.58})$ composite-order, q-type
  - [GHKKP25] $|\mathsf{crs}| = O(L)$ prime-order, GGM

Lattice-Based. Constructions from Falsifiable Assumptions

- state of the art: **key-policy** Reg-ABE for **bounded** depth circuits
- future targets:
  ciphertext-policy Reg-ABE, **unbounded** depth circuits, **uniform** models of computations

Organization

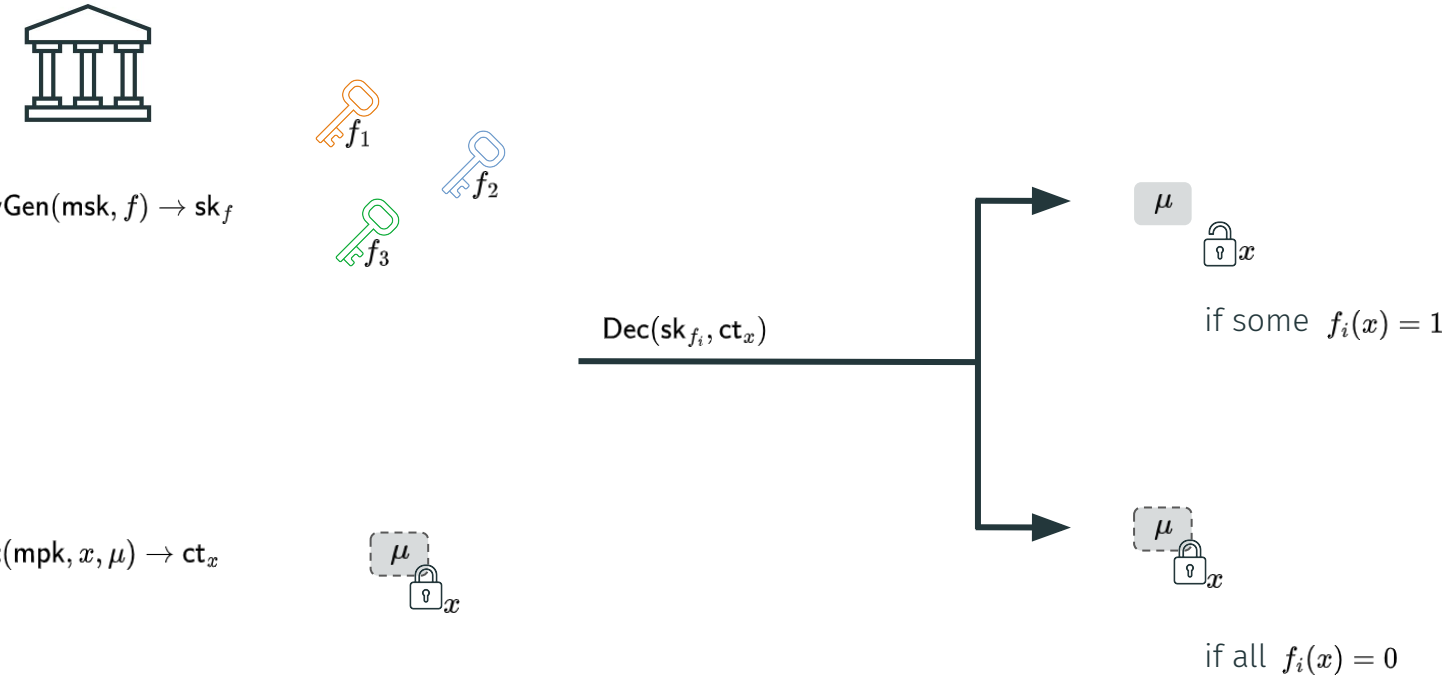
Part I. **Removing the Trusted Authority** (Registered ABE and FE)

- registered ABE via general paradigm [AC:P**S**25a]
- more [P**S**25b, P**S**T25]

Part II. **Supporting Multiple Encryptors** (Multi-Client ABE)

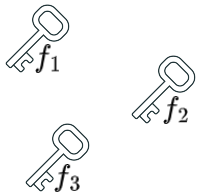
- formal definition, first instantiations [TCC:P**S**24]
- more functionalities, different assumptions [PKC:**S**25]

Recap: Attribute-Based Encryption



Multi-Input Attribute-Based Encryption [BJK+18]

$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$



$\mathsf{Enc}_1(\mathsf{msk}, x_1, \mu_1) \to \mathsf{ct}_{x_1}$

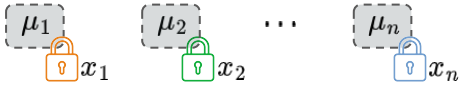
$\mathsf{Enc}_2(\mathsf{msk}, x_2, \mu_2) \to \mathsf{ct}_{x_2}$

$\vdots$

$\mathsf{Enc}_n(\mathsf{msk}, x_n, \mu_n) \to \mathsf{ct}_{x_n}$

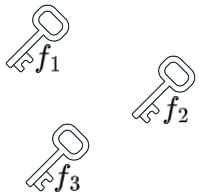
$\mathsf{Dec}(\mathsf{sk}_{f_i}, \mathsf{ct}_{x_1}, \ldots, \mathsf{ct}_{x_n})$

if some $f_i(x_1, \ldots, x_n) = 1$



if all $f_i(x_1, \ldots, x_n) = 0$

Multi-Client Attribute-Based Encryption [PS24]



$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$

separation & corruption of encryption keys

$\mathsf{Enc}(\mathsf{ek}_1, \mathsf{lab}, x_1, \mu_1) \to \mathsf{ct}_{\mathsf{lab},x_1}$ $\quad \mu_1 \; x_1$

$\mathsf{Enc}(\mathsf{ek}_2, \mathsf{lab}, x_2, \mu_2) \to \mathsf{ct}_{\mathsf{lab},x_2}$ $\quad \mu_2 \; x_2$

$\vdots$

$\mathsf{Enc}(\mathsf{ek}_n, \mathsf{lab}, x_n, \mu_n) \to \mathsf{ct}_{\mathsf{lab},x_n}$ $\quad \mu_n \; x_n$

$\mathsf{Dec}(\mathsf{sk}_{f_i}, \mathsf{ct}_{\mathsf{lab},x_1}, \ldots, \mathsf{ct}_{\mathsf{lab},x_n})$

$\mu_1 \; x_1 \quad \mu_2 \; x_2 \quad \cdots \quad \mu_n \; x_n$

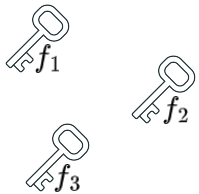
if some $f_i(x_1, \ldots, x_n) = 1$

$\mu_1 \; x_1 \quad \mu_2 \; x_2 \quad \cdots \quad \mu_n \; x_n$

if all $f_i(x_1, \ldots, x_n) = 0$

Multi-Client Attribute-Based Encryption [PS24]

$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$



separation & corruption of encryption keys

$\mathsf{Enc}(\mathsf{ek}_1, \mathsf{lab}, x_1, \mu_1) \to \mathsf{ct}_{\mathsf{lab},x_1}$ μ_1 x_1

$\mathsf{Enc}(\mathsf{ek}_2, \mathsf{lab}, x_2, \mu_2) \to \mathsf{ct}_{\mathsf{lab},x_2}$ μ_2 x_2

$\vdots$

$\mathsf{Enc}(\mathsf{ek}_n, \mathsf{lab}, x_n, \mu_n) \to \mathsf{ct}_{\mathsf{lab},x_n}$ μ_n x_n

encryption w.r.t. labels

$\mathsf{Dec}(\mathsf{sk}_{f_i}, \mathsf{ct}_{\mathsf{lab},x_1}, \ldots, \mathsf{ct}_{\mathsf{lab},x_n})$

μ_1 μ_2 $\cdots$ μ_n

x_1 x_2 x_n

if some $f_i(x_1, \ldots, x_n) = 1$

μ_1 μ_2 $\cdots$ μ_n

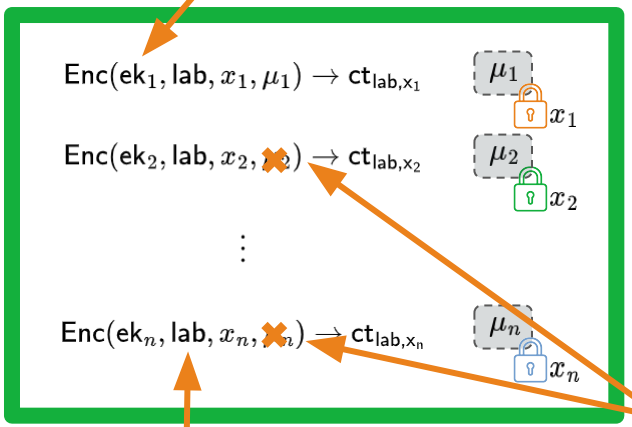
x_1 x_2 x_n

if all $f_i(x_1, \ldots, x_n) = 0$

Multi-Client Attribute-Based Encryption [PS24]

$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$

separation & corruption of encryption keys

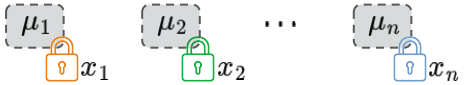


encryption w.r.t. labels

w.l.o.g., only 1st client encrypts message

$\mathsf{Dec}(\mathsf{sk}_{f_i}, \mathsf{ct}_{\mathsf{lab},x_1}, \ldots, \mathsf{ct}_{\mathsf{lab},x_n})$

if some $f_i(x_1, \ldots, x_n) = 1$



if all $f_i(x_1, \ldots, x_n) = 0$

Remember the General Paradigm?

sk_f 

$\mathsf{ct}_{\mathbf{x}}$ 

$$\left.\begin{array}{l} \mathsf{isk}([\mathbf{L}_1]_2),\ \ldots,\ \mathsf{isk}([\mathbf{L}_m]_2),\ \mathsf{isk}'([\sigma, 1]_2) \\ \mathsf{ict}([s(1, \mathbf{x})]_1), \qquad\qquad \mathsf{ict}'([s, \mu]_1) \end{array}\right\} \longrightarrow \left\{[s \cdot L_j(\mathbf{x})]_\mathsf{t}\right\}_j,\ [s\sigma + \mu]_\mathsf{t}$$

Pairing-Based IPFE

$\mathbf{u}$ 

$\mathsf{ict}([\mathbf{u}]_1)$

$\mathsf{isk}([\mathbf{v}]_2)$

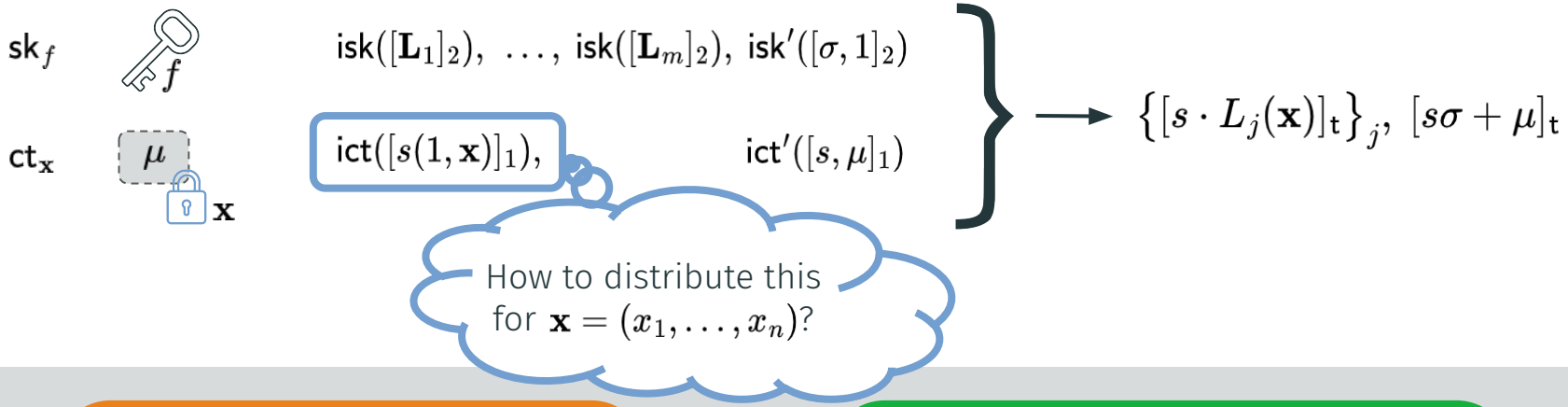
$$\longrightarrow [\mathbf{u}^\top \mathbf{v}]_\mathsf{t}$$

Linear Garbling

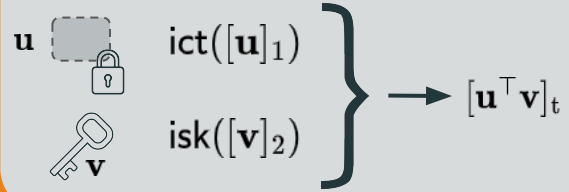
$$\mathsf{Garble}(f, \sigma; \mathbf{r}) \to \mathbf{L} = (\mathbf{L}_1 \| \ldots \| \mathbf{L}_m)$$

$$\mathsf{Eval}(f, \mathbf{x}, \boldsymbol{\ell} := (1, \mathbf{x}) \cdot \mathbf{L}) \to \sigma \cdot f(\mathbf{x})$$

Remember the General Paradigm?



Pairing-Based IPFE

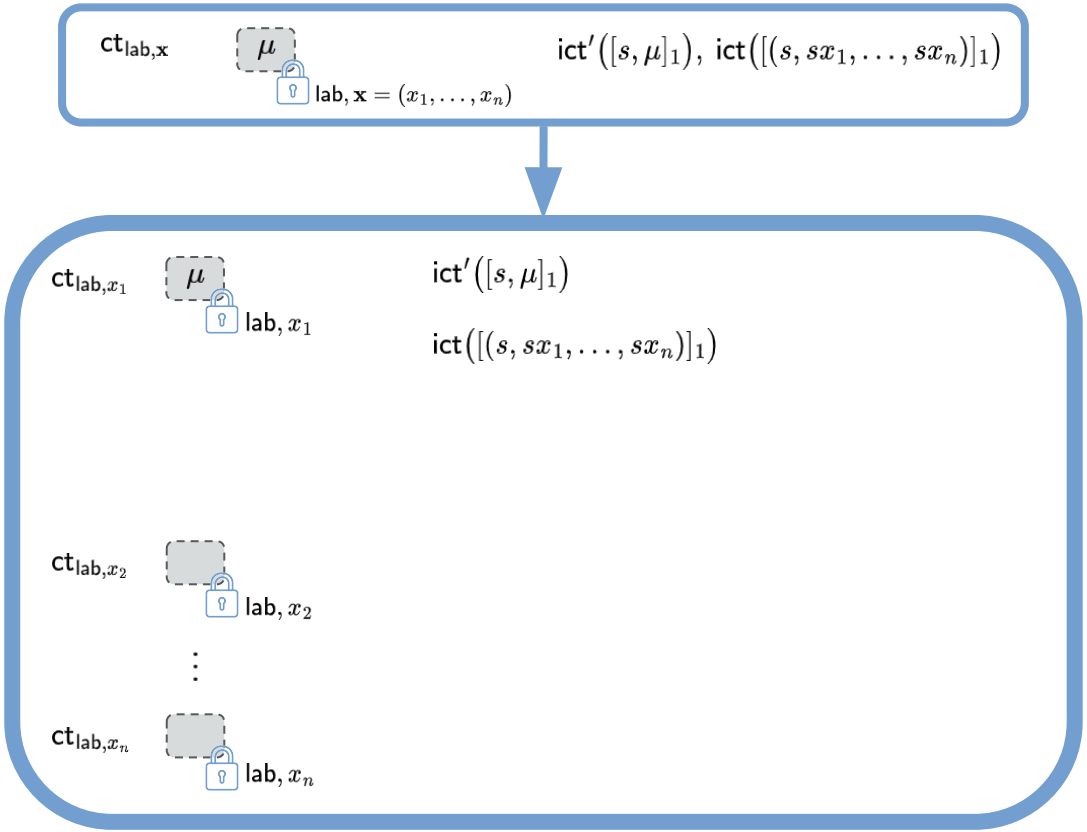


Linear Garbling

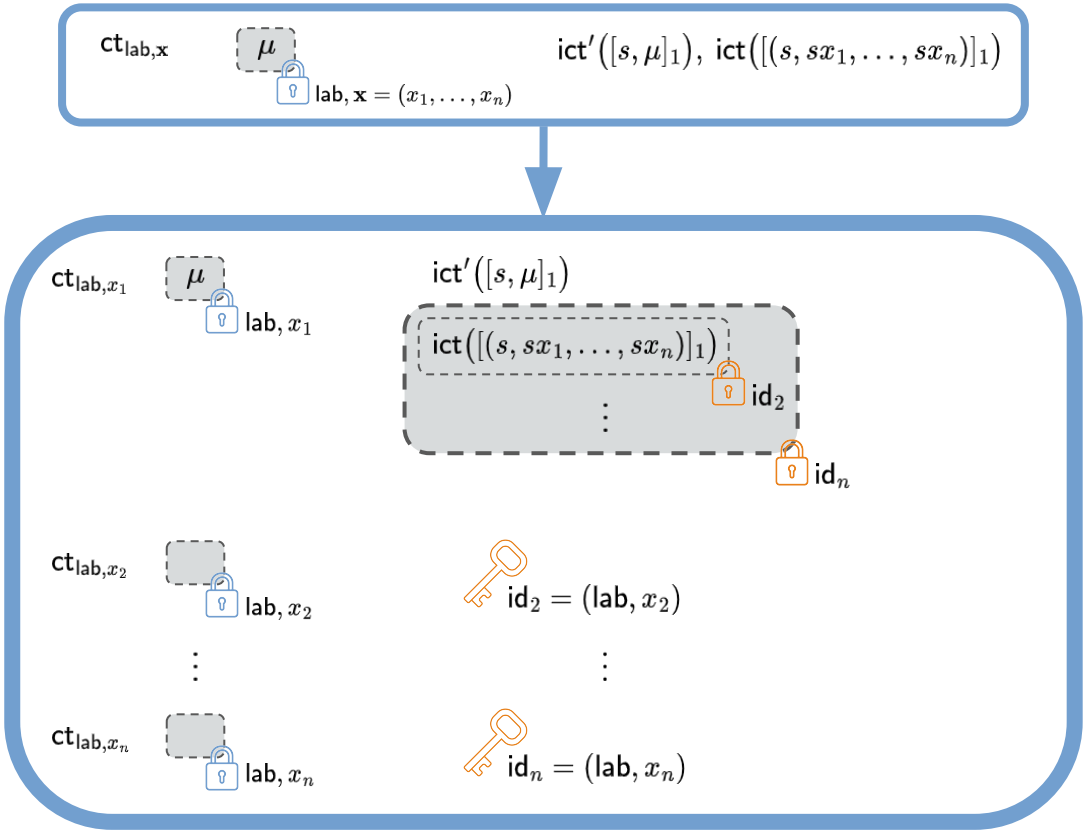
$$\mathsf{Garble}(f,\sigma;\mathbf{r})\to\mathbf{L}=(\mathbf{L}_1\|\ldots\|\mathbf{L}_m)$$

$$\mathsf{Eval}(f,\mathbf{x},\boldsymbol{\ell}:=(1,\mathbf{x})\cdot\mathbf{L})\to\sigma\cdot f(\mathbf{x})$$

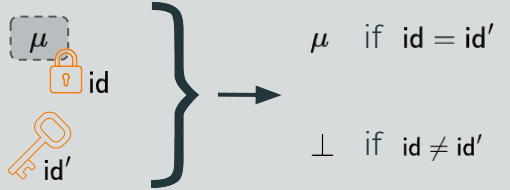
Distributed Encryption



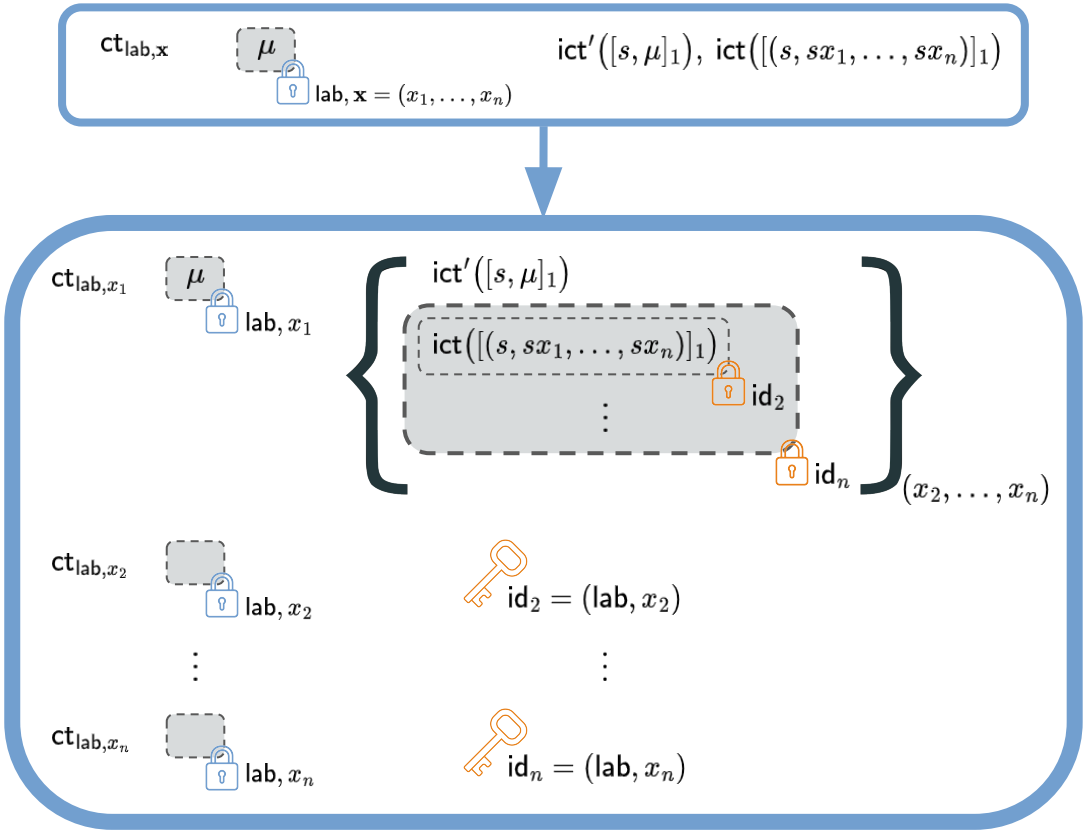
Distributed Encryption

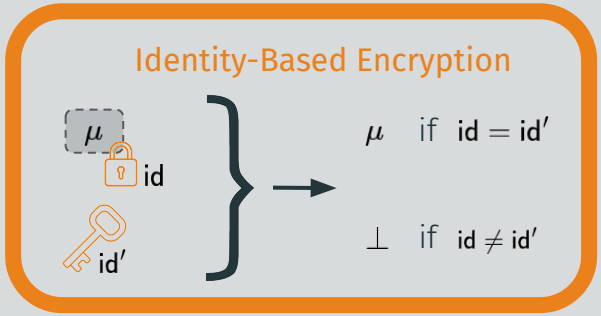


Identity-Based Encryption

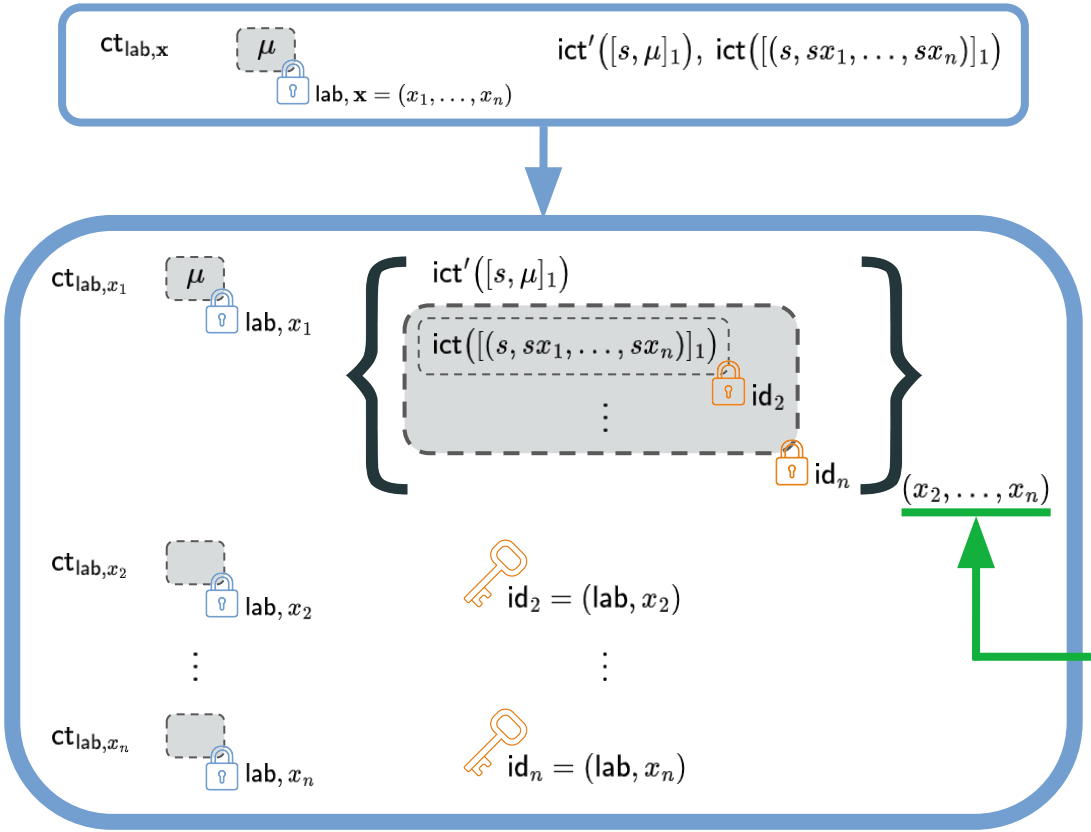


Distributed Encryption

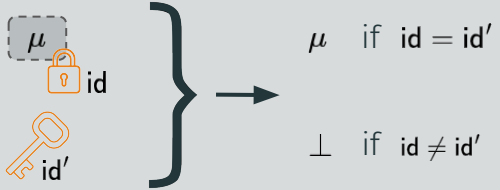




Distributed Encryption



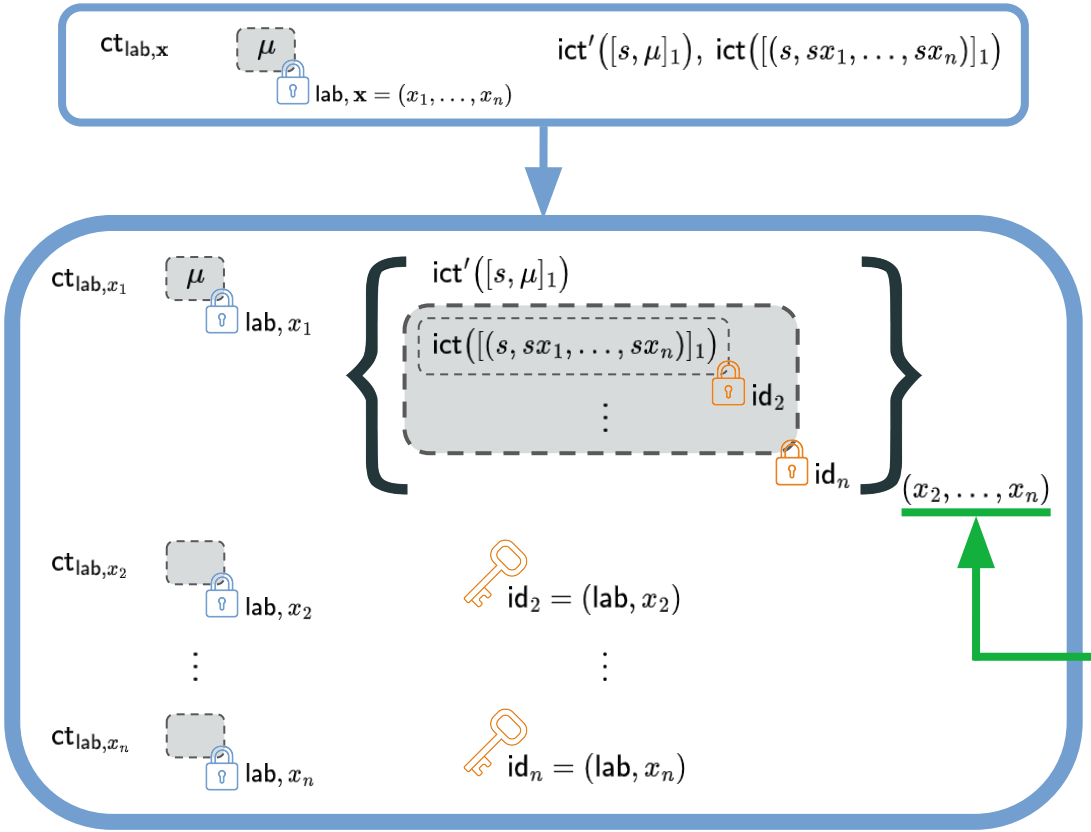
Identity-Based Encryption



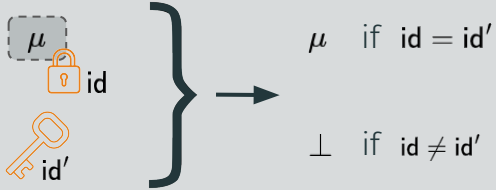
Which $(x_2, \ldots, x_n) \in \{0,1\}^{n-1}$ do we need?

- all $\rightarrow$ NC^1 or ABP policies for $O(\log \lambda)$ inputs

Distributed Encryption



Identity-Based Encryption



Which $(x_2, \ldots, x_n) \in \{0,1\}^{n-1}$ do we need?

- **all** ↛ NC^1 or ABP policies for $O(\log \lambda)$ inputs
- **constant-size subsets** ↛ NC^0 and constant-threshold policies

A Different Perspective

Fact. MI-ABE for polynomial arity and NC^1 policies $\Longrightarrow$ witness encryption for NP

A Different Perspective

Fact. MI-ABE for polynomial arity and NC^1 policies $\Longrightarrow$ witness encryption for NP

We consider settings that *circumvent* this implication.

1) **Weaker Policies** ($\nrightarrow$ cannot verify NP relation)
   - MC-ABE for NC^0 policies
   - MC-ABE for constant-threshold policies

A Different Perspective

Fact. MI-ABE for polynomial arity and NC^1 policies $\Longrightarrow$ witness encryption for NP

We consider settings that *circumvent* this implication.

1) **Weaker Policies** ($\nrightarrow$ cannot verify NP relation)
   - MC-ABE for NC^0 policies
   - MC-ABE for constant-threshold policies

2) **Short Inputs** ($\nrightarrow$ WE with exp-size ciphertexts)
   - MC-ABE for NC^1 for parameters s.t. $|x_1| + \cdots + |x_n| = O(\log \lambda)$

A Different Perspective

Fact. MI-ABE for polynomial arity and NC^1 policies $\Longrightarrow$ witness encryption for NP

We consider settings that *circumvent* this implication.

1) **Weaker Policies** ($\nrightarrow$ cannot verify NP relation)
   - MC-ABE for NC^0 policies
   - MC-ABE for constant-threshold policies

2) **Short Inputs** ($\nrightarrow$ WE with exp-size ciphertexts)
   - MC-ABE for NC^1 for parameters s.t. $|x_1| + \cdots + |x_n| = O(\log \lambda)$

3) **Weaker Security Model** ($\nrightarrow$ MC-ABE with OT labels $\not\Rightarrow$ MI-ABE)
   - MC-ABE for NC^1 under one-time label restriction

Part II. More Results and Open Questions

More Results.

- [PS24] generic compiler MC-ABE + lockable obfuscation $\Longrightarrow$ MC-PE
- [S25] new function classes (succinctly enumerable, conjunction policies), lattice instantiations

$$f(x_1, \ldots, x_n) = f_1(x_1) \wedge \cdots \wedge f_n(x_n)$$

Part II. More Results and Open Questions

More Results.

- [PS24] generic compiler MC-ABE + lockable obfuscation $\Longrightarrow$ MC-PE
- [S25] new function classes (succinctly enumerable, conjunction policies), lattice instantiations

$$f(x_1, \ldots, x_n) = f_1(x_1) \wedge \cdots \wedge f_n(x_n)$$

Open Question. Relation between MC-ABE and Witness Encryption

- goal: narrow down the gap
  instantiations of MC-ABE from standard assumptions $\Leftrightarrow$ implications to WE
- e.g., new constructions of MC-ABE supporting

more than 2 slots	(ideally: poly)
more than NC^0 policies	(ideally: at least NC^1)
based on falsifiable lattice assumption	(ideally: standard)

Conclusion

Part I. Removing the Trusted Authority (Registered ABE and FE)

- registered ABE and FE via general paradigm [AC:PS25a]
- adaptive security, stronger functionalities, lattice-based assumptions [PS25b, PST25]

Part II. Supporting Multiple Encryptors (Multi-Client ABE)

- formal definition, first instantiations from pairings [TCC:PS24]
- more functionalities, instantiations from lattices [PKC:S25]
- generic compiler MC-ABE $\Longrightarrow$ MC-PE [TCC:PS24]

Conclusion

Part I. Removing the Trusted Authority (Registered ABE and FE)

- registered ABE and FE via general paradigm [AC:PS25a]
- adaptive security, stronger functionalities, lattice-based assumptions [PS25b, PST25]

Part II. Supporting Multiple Encryptors (Multi-Client ABE)

- formal definition, first instantiations from pairings [TCC:PS24]
- more functionalities, instantiations from lattices [PKC:S25]
- generic compiler MC-ABE $\Longrightarrow$ MC-PE [TCC:PS24]

Thank you! :-)

Publications

* discussed in the thesis
** discussed in the defense

- Tracing a Linear Subspace: Application to Linearly-Homomorphic Group Signatures
  Chloé Hébant, David Pointcheval, and Robert Schädlich | PKC 2023
- Decentralized Multi-Client Functional Encryption with Strong Security
  Ky Nguyen, David Pointcheval, and Robert Schädlich | IACR CiC 2024
- Multi-Client Attribute-Based and Predicate Encryption from Standard Assumptions**
  David Pointcheval and Robert Schädlich | TCC 2024
- Dynamic Decentralized Functional Encryption: Generic Constructions with Strong Security
  Ky Nguyen, David Pointcheval, and Robert Schädlich | PKC 2025
- Multi-Client Attribute-Based and Predicate Encryption, Revisited*
  Robert Schädlich | PKC 2025
- A General Framework for Registered Functional Encryption via User-Specific Pre-Constraining**
  Tapal Pal and Robert Schädlich | Asiacrypt 2025 (to appear)
- Ciphertext-Updatable Attribute-Based and Predicate Encryption from Lattices
  Robert Schädlich, Linda Scheu-Hachtel, Erkan Tairi, and Yuejun Wang | Preprint 2025 (under submission)
- Registered Functional Encryption for Attribute-Weighted Sums with Access Control*
  Tapas Pal and Robert Schädlich | Preprint 2025 (under submission)
- Registered Functional Encryption for Pseudorandom Functionalities from Lattices: Registered ABE for Unbounded Depth Circuits and Turing Machines, and More*
  Tapas Pal, Robert Schädlich, and Erkan Tairi | Preprint 2025 (under submission)

Arithmetization of TM Computations

- consider TM $M = (Q, \mathbf{y}_{\text{acc}}, \delta)$ and denote by (N, S, T) the input length, space and runtime
- set of internal configurations $\mathcal{C} = [N] \times [S] \times \{0,1\}^S \times [Q]$ with initial configuration $c_0 = (1, 1, \mathbf{0}_S, 1)$

(input tape head, working tape head, working tape, state)

- define transition matrix $\mathbf{T}(\mathbf{x}) \in \mathbb{Z}_p^{\mathcal{C} \times \mathcal{C}}$ as

$$\mathbf{T}(\mathbf{x})[c', c] = \begin{cases} 1 & \text{if } c \to_M c' \\ 0 & \text{otherwise} \end{cases}$$

c' = (k', j', w', q') *c = (k, j, w, q)*

- we have $\mathbf{T}(\mathbf{x}) \cdot \mathbf{e}_c^\top = \mathbf{e}_{c'}^\top$ and more general

$$M|_{N,S,T}(\mathbf{x}) = (\mathbf{1} \otimes \mathbf{y}_{\text{acc}}) \cdot \big(\mathbf{T}(\mathbf{x})\big)^T \cdot \mathbf{e}_{c_0}^\top$$

... so we only need to garble **matrix multiplication**

Arithmetic Key Garbling for Logspace TMs

- **garbling:** sample $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_T \leftarrow_\$ \mathbb{Z}_p^{\mathcal{C}}$ and output label functions

$$
\begin{aligned}
L_{\mathsf{init}}(\mathbf{x}) &= \sigma_0 + \mathbf{r}_0 \cdot \mathbf{e}_{c_0}^\top \\
L_t(\mathbf{x}) = \big(L_t[c](\mathbf{x})\big)_{c \in \mathcal{C}} &= -\mathbf{r}_{t-1} + \mathbf{r}_t \cdot \mathbf{T}(\mathbf{x}) \\
L_{T+1}(\mathbf{x}) = \big(L_{T+1}[c](\mathbf{x})\big)_{c \in \mathcal{C}} &= -\mathbf{r}_T + \sigma_1 \cdot \big(\mathbf{1}_{[N]\times[S]\times\{0,1\}^s} \otimes \mathbf{y}_{\mathsf{acc}}\big)
\end{aligned}
$$

- **evaluation:** given labels $\ell_{\mathsf{init}} := L_{\mathsf{init}}(\mathbf{x})$, $\{\boldsymbol{\ell}_t := L_t(\mathbf{x})\}_{t \in [T+1]}$, output

$$
\ell_{\mathsf{init}} + \sum_{t \in [T+1]} \boldsymbol{\ell}_t \cdot \mathbf{T}(\mathbf{x})^{t-1} \cdot \mathbf{e}_{c_0}^\top = \sigma_0 + \sigma_1 \big(\mathbf{1}_{[N]\times[S]\times\{0,1\}^s} \otimes \mathbf{y}_{\mathsf{acc}}\big) \cdot \mathbf{T}(\mathbf{x})^T \cdot \mathbf{e}_{c_0}^\top = \sigma_0 + \sigma_1 \cdot M|_{N,S,T}(\mathbf{x})
$$

telescoping sum

Security? (1) $L_1(\mathbf{x}), \ldots, L_{T+1}(\mathbf{x})$ are jointly pseudorandom
(2) $L_{init}(\mathbf{x})$ deterministically computable given $L_1(\mathbf{x}), \ldots, L_{T+1}(\mathbf{x})$

Decomposition of the Labels

- **garbling:** sample $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_T \leftarrow_\$ \mathbb{Z}_p^{\mathcal{C}}$ and output label functions

  $\mathcal{C} = [N] \times [S] \times \{0,1\}^S \times [Q]$

$$L_{\text{init}}(\mathbf{x}) = \sigma_0 + \mathbf{r}_0 \cdot \mathbf{e}_{c_0}^\top$$
$$L_t(\mathbf{x}) = -\mathbf{r}_{t-1} + \mathbf{r}_t \cdot \mathbf{T}(\mathbf{x})$$
$$L_{T+1}(\mathbf{x}) = -\mathbf{r}_T + \sigma_1 \cdot (\mathbf{1} \otimes \mathbf{y}_{\text{acc}})$$

✔

- **idea:** set $\mathbf{r}_t = \mathbf{r}_{\text{x},t} \otimes \mathbf{r}_{\text{f}}$ where $\mathbf{r}_{\text{x},t} \leftarrow_\$ \mathbb{Z}_p^{[N]\times[S]\times\{0,1\}^S}$ and $\mathbf{r}_{\text{f}} \leftarrow_\$ \mathbb{Z}_p^{[Q]}$
- **decomposition of the labels:**

$$L_{\text{init}}(\mathbf{x}) = s \cdot \sigma_0 + (\mathbf{r}_{\text{x},0} \otimes \mathbf{r}_{\text{f}})[1, 1, \mathbf{0}_S, 1] = \underbrace{s \cdot \sigma_0 + \mathbf{r}_{\text{x},0}[1, 1, \mathbf{0}_S]}_{u} \cdot \underbrace{\mathbf{r}_{\text{f}}[1]}_{P_i}$$

Decomposition of the Labels

- **garbling:** sample $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_T \leftarrow_\$ \mathbb{Z}_p^{\mathcal{C}}$ and output label functions

  $\mathcal{C} = [N] \times [S] \times \{0,1\}^S \times [Q]$

$$L_{\text{init}}(\mathbf{x}) = \sigma_0 + \mathbf{r}_0 \cdot \mathbf{e}_{c_0}^\top$$
$$L_t(\mathbf{x}) = -\mathbf{r}_{t-1} + \mathbf{r}_t \cdot \mathbf{T}(\mathbf{x})$$
$$L_{T+1}(\mathbf{x}) = -\mathbf{r}_T + \sigma_1 \cdot \left(\mathbf{1} \otimes \mathbf{y}_{\text{acc}}\right)$$

- **idea:** set $\mathbf{r}_t = \mathbf{r}_{\text{x},t} \otimes \mathbf{r}_{\text{f}}$ where $\mathbf{r}_{\text{x},t} \leftarrow_\$ \mathbb{Z}_p^{[N]\times[S]\times\{0,1\}^S}$ and $\mathbf{r}_{\text{f}} \leftarrow_\$ \mathbb{Z}_p^{[Q]}$
- **decomposition of the labels:**

$$L_{T+1}[\underbrace{k, j, \mathbf{w}}_{=: c}, q](\mathbf{x}) = -(\mathbf{r}_{\text{x},T} \otimes \mathbf{r}_{\text{f}})[c, q] + s \cdot \sigma_1 \cdot \left(\mathbf{1} \otimes \mathbf{y}_{\text{acc}}\right)[c, q] = -\mathbf{r}_{\text{x},T}[c] \cdot \mathbf{r}_{\text{f}}[q] + s \cdot \sigma_1 \cdot \mathbf{y}_{\text{acc}}[q]$$

u, V_i, P_i

Decomposition of the Labels

- **garbling:** sample $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_T \leftarrow_\$ \mathbb{Z}_p^{\mathcal{C}}$ and output label functions

  $\mathcal{C} = [N] \times [S] \times \{0,1\}^S \times [Q]$

$$L_{\mathsf{init}}(\mathbf{x}) = \sigma_0 + \mathbf{r}_0 \cdot \mathbf{e}_{c_0}^\top \quad ✔$$
$$L_t(\mathbf{x}) = -\mathbf{r}_{t-1} + \mathbf{r}_t \cdot \mathbf{T}(\mathbf{x}) \quad ✘$$
$$L_{T+1}(\mathbf{x}) = -\mathbf{r}_T + \sigma_1 \cdot (\mathbf{1} \otimes \mathbf{y}_{\mathsf{acc}}) \quad ✔$$

- **idea:** set $\mathbf{r}_t = \mathbf{r}_{\mathsf{x},t} \otimes \mathbf{r}_{\mathsf{f}}$ where $\mathbf{r}_{\mathsf{x},t} \leftarrow_\$ \mathbb{Z}_p^{[N]\times[S]\times\{0,1\}^S}$ and $\mathbf{r}_{\mathsf{f}} \leftarrow_\$ \mathbb{Z}_p^{[Q]}$
- **decomposition of the labels:**

$$L_t[\underline{c}, q](\mathbf{x}) = -(\mathbf{r}_{\mathsf{x},t-1} \otimes \mathbf{r}_{\mathsf{f}})[\underline{c}, q] + \big((\mathbf{r}_{\mathsf{x},t} \otimes \mathbf{r}_{\mathsf{f}}) \cdot \mathbf{T}(\mathbf{x})\big)[\underline{c}, q] = \underbrace{-\mathbf{r}_{\mathsf{x},t-1}[\underline{c}] \cdot \mathbf{r}_{\mathsf{f}}[q]}_{u} + \underbrace{\big((\mathbf{r}_{\mathsf{x},t} \otimes \mathbf{r}_{\mathsf{f}}) \cdot \mathbf{T}(\mathbf{x})\big)[\underline{c}, q]}_{?}$$

P_i

Block Structure of Transition Matrix

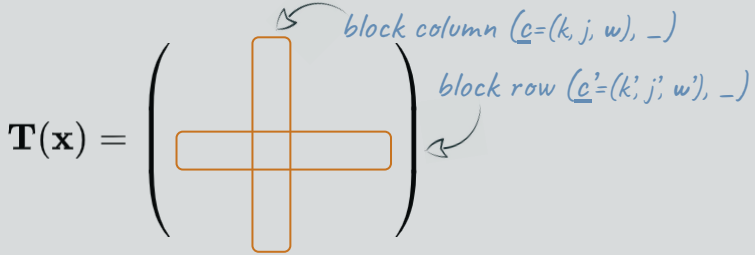
- transition matrix $\mathbf{T}(\mathbf{x})[c', c] = \begin{cases} 1 & \text{if } \delta(q, \mathbf{x}[k], \mathbf{w}[j]) = (q', \mathbf{w}'[j], k' - k, j' - j),\ \mathbf{w}[\neq j] = \mathbf{w}'[\neq j] \\ 0 & \text{otherwise} \end{cases}$

  c' = (k', j', w', q') c = (k, j, w, q)

- consider Q x Q blocks $\mathbf{T}(\mathbf{x})[(\underbrace{k', j', \mathbf{w}', \_}_{c'}), (\underbrace{k, j, \mathbf{w}, \_}_{c})]$

  -> either zero matrix
  -> or "**transition block**" in $\mathcal{B} = \{\mathbf{B}_{x,w,w',\Delta k,\Delta j} | x, w, w' \in \{0,1\},\ \Delta k, \Delta j \in \{0, \pm 1\}\}$

- **observation:** each $\mathbf{B}_{x,w,w',\Delta k,\Delta j}$ appears at most once per "block column"



Block Structure of Transition Matrix

- transition matrix $\mathbf{T}(\mathbf{x})[c', c] = \begin{cases} 1 & \text{if } \delta(q, \mathbf{x}[k], \mathbf{w}[j]) = (q', \mathbf{w}'[j], k' - k, j' - j),\ \mathbf{w}[\neq j] = \mathbf{w}'[\neq j] \\ 0 & \text{otherwise} \end{cases}$

  c' = (k', j', w', q') c = (k, j, w, q)

- consider Q x Q blocks $\mathbf{T}(\mathbf{x})[(\underbrace{k', j', \mathbf{w}'}_{c'}, \_), (\underbrace{k, j, \mathbf{w}}_{c}, \_)]$

  -> either zero matrix
  -> or "**transition block**" in $\mathcal{B} = \left\{ \mathbf{B}_{x,w,w',\Delta k,\Delta j} \middle| x, w, w' \in \{0,1\},\ \Delta k, \Delta j \in \{0, \pm 1\} \right\}$

- **observation:** each $\mathbf{B}_{x,w,w',\Delta k,\Delta j}$ appears at most once per "block column", position independent of $\mathbf{x}$

decomposition of the last term:

$$(\mathbf{r}_{\mathsf{x},t} \otimes \mathbf{r}_{\mathsf{f}}) \cdot \mathbf{T}(\mathbf{x})[(\_,\_), (c, q)] = \sum_{w', \Delta k, \Delta j} \mathbf{r}_{\mathsf{x},t}[c'] \cdot \mathbf{r}_{\mathsf{f}}[q] \cdot \mathbf{B}_{\mathbf{x}[k], \mathbf{w}[j], w', \Delta k, \Delta j}$$

u V_i P_i