Fine-Grained Access Control in Multi-Client Functional Encryption

Ky Nguyen[1], Duong Hieu Phan[2], David Pointcheval[1]

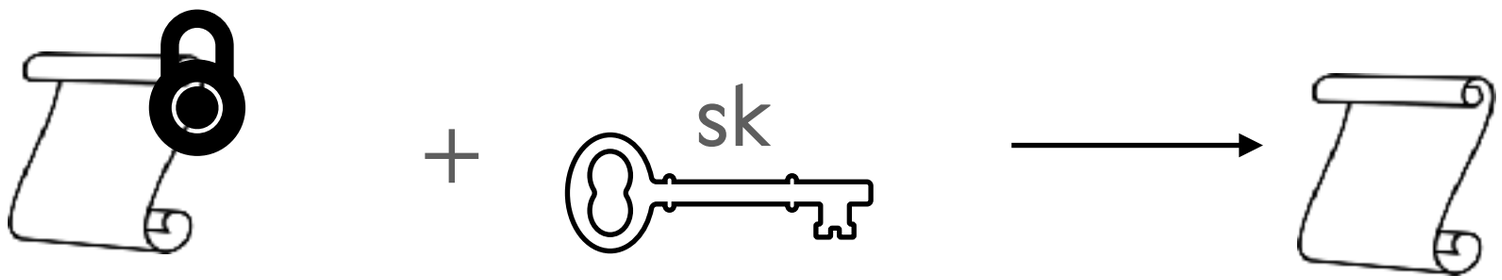
[1] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[2] LTCI, Telecom Paris, Institut Polytechnique de Paris, France

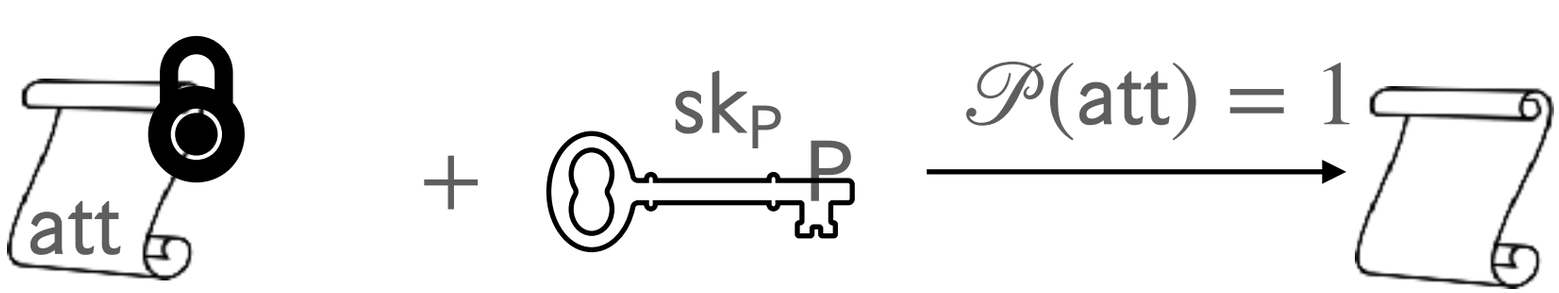
Outline

1. Motivation and Context
2. Related Works and Our Contributions
3. Technical Tools
4. Overview of Constructions

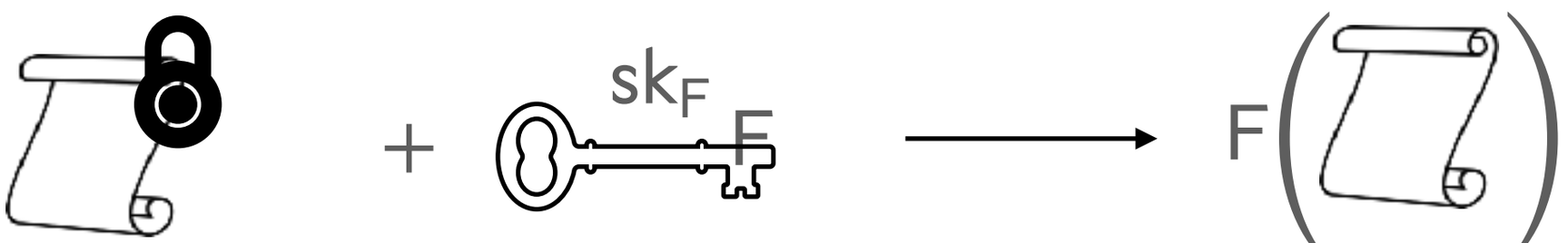
Motivation

Functional Encryption [O'N10,BSW11]



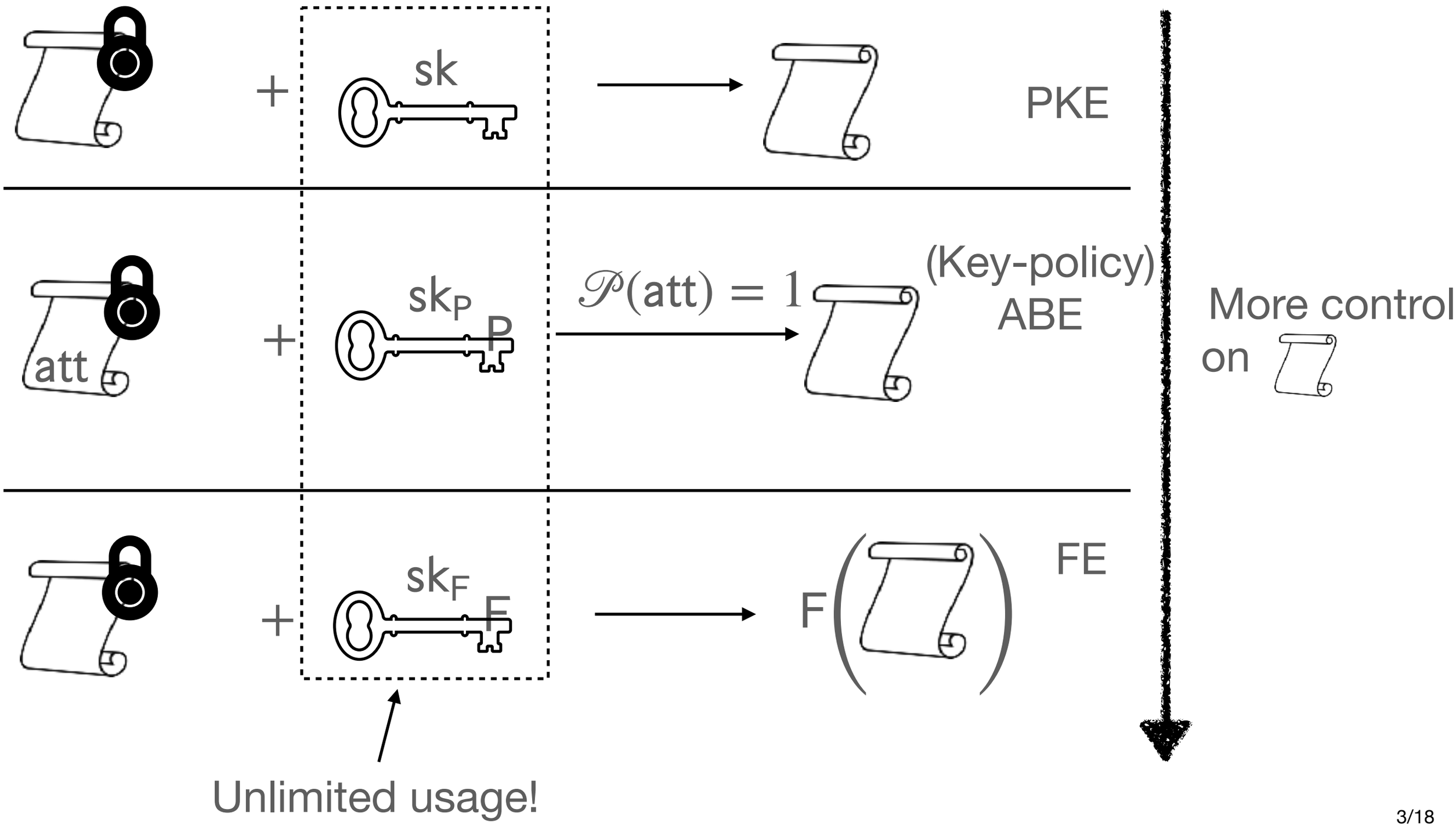


More control on



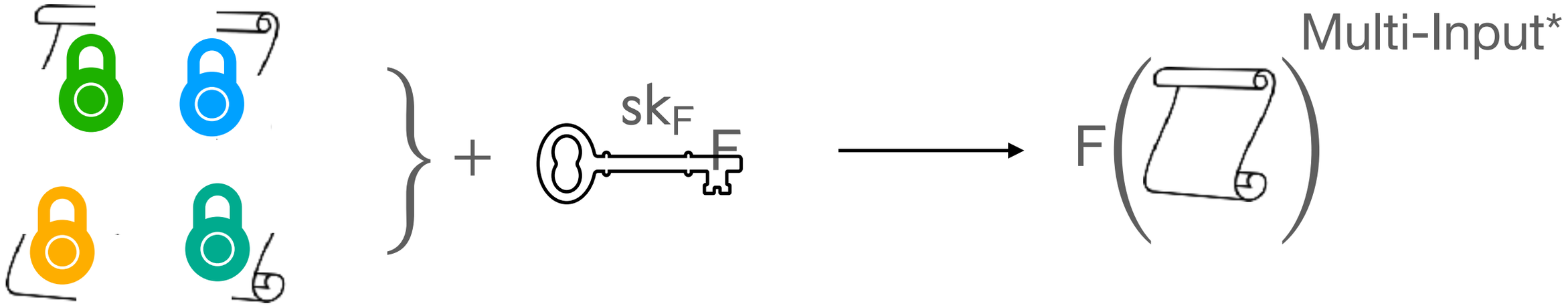
Motivation

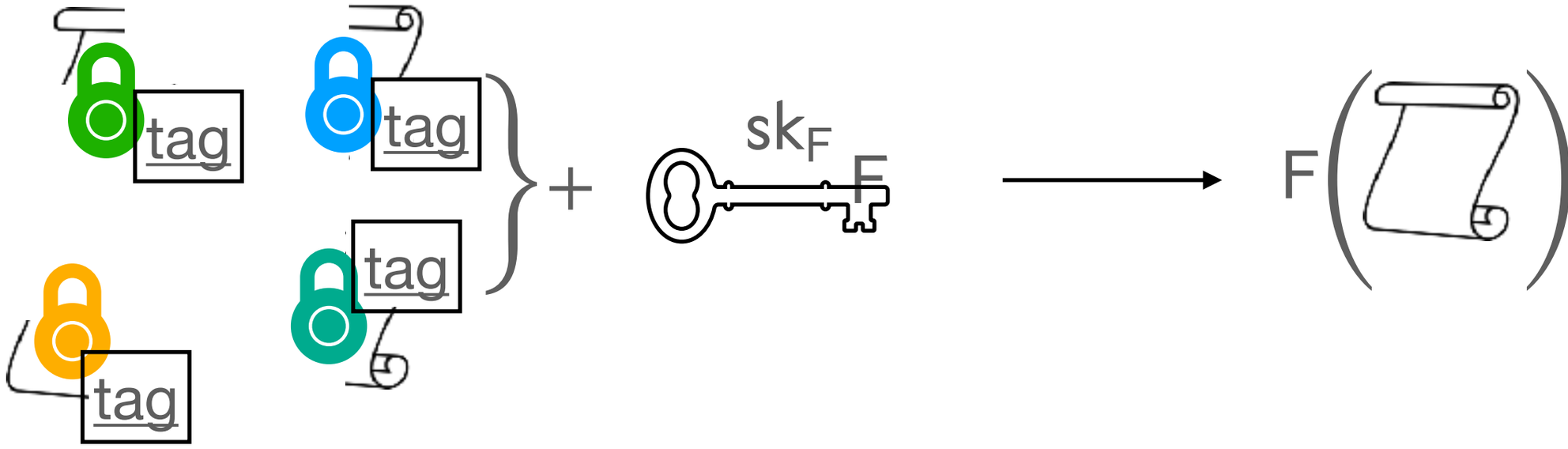
Functional Encryption [O'N10,BSW11]



Motivation

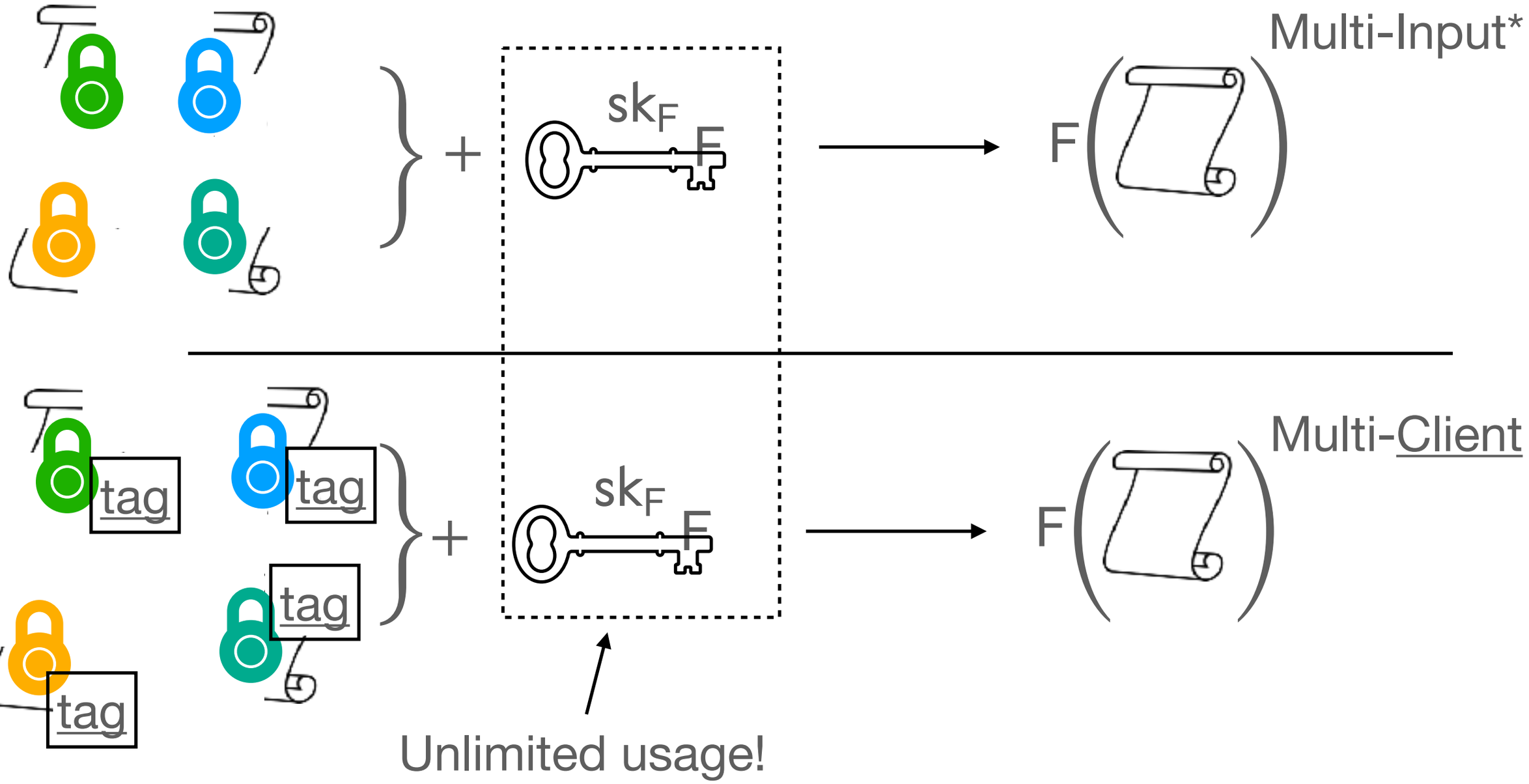
FE in Multi-User Setting [GGG+14, GKL+13]





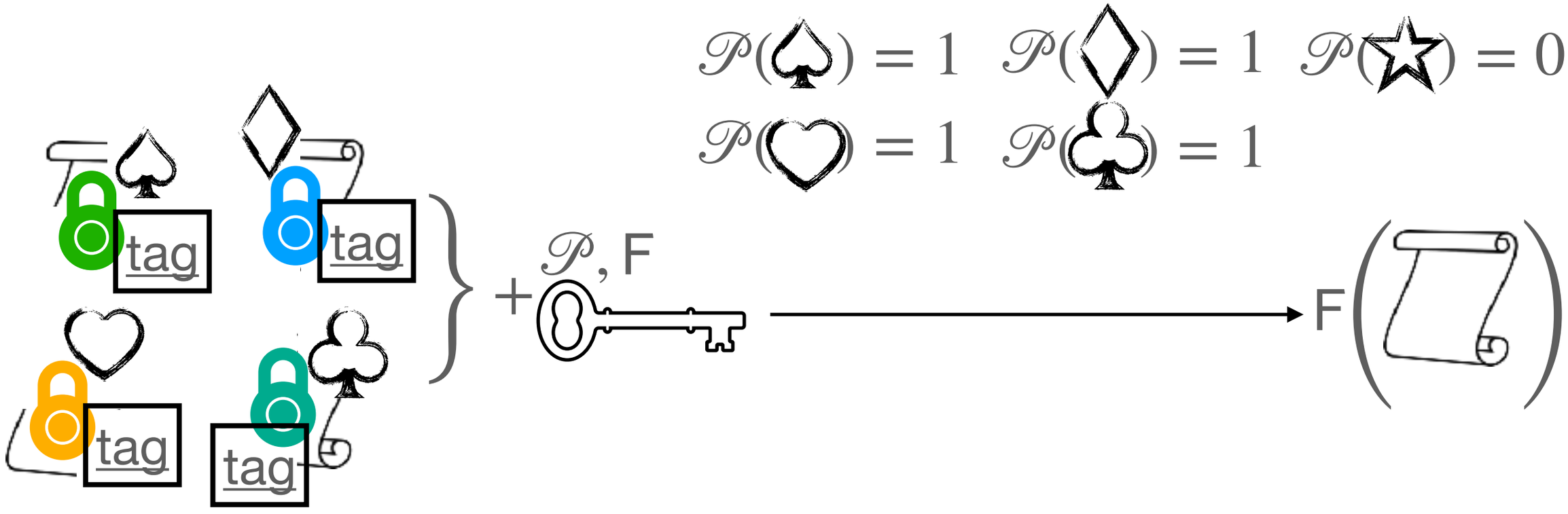
Motivation

FE in Multi-User Setting [GGG+14, GKL+13]



Motivation

Controlling Decryption Keys in Multi-Client FE (MCFE)

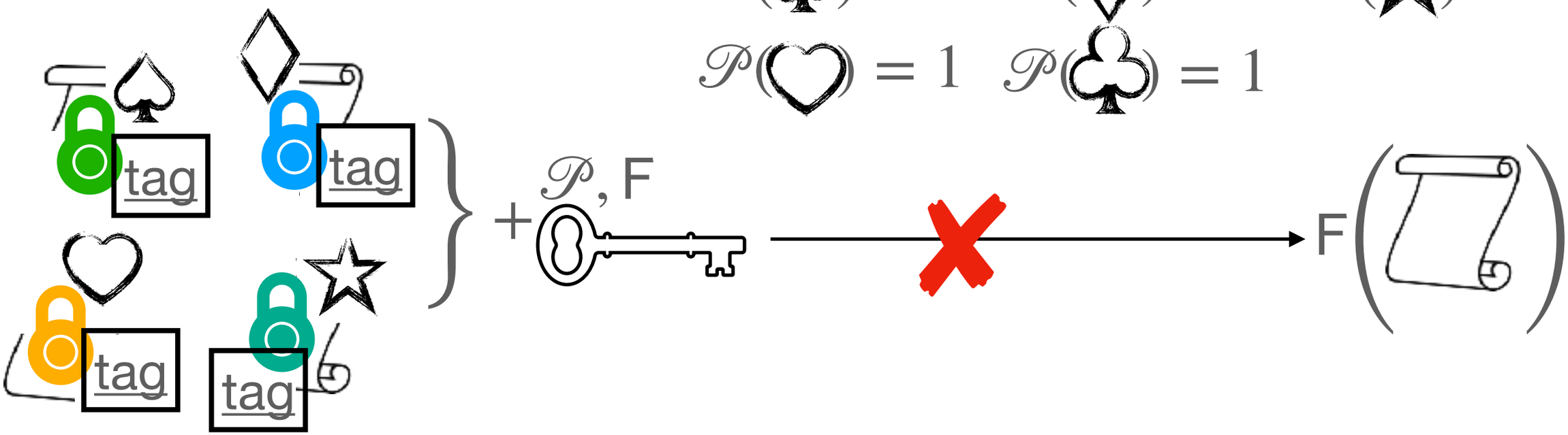


Motivation

Controlling Decryption Keys in Multi-Client FE (MCFE)

$\mathscr{P}(\spadesuit) = 1 \quad \mathscr{P}(\diamondsuit) = 1 \quad \mathscr{P}(\star) = 0$

$\mathscr{P}(\heartsuit) = 1 \quad \mathscr{P}(\clubsuit) = 1$

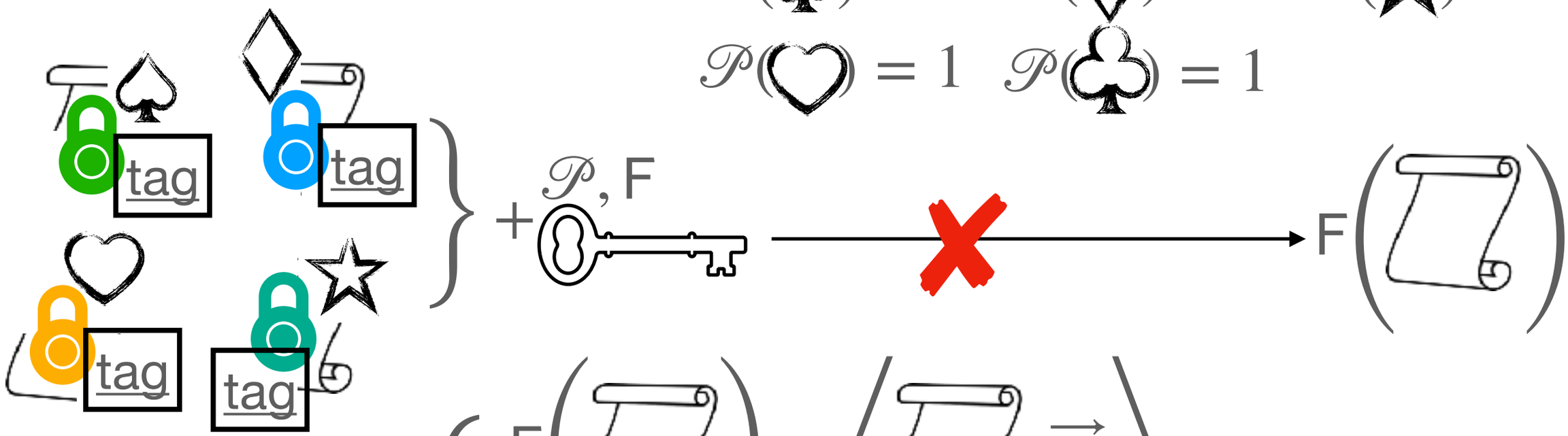


Motivation

Controlling Decryption Keys in Multi-Client FE (MCFE)

$\mathscr{P}(\spadesuit) = 1 \quad \mathscr{P}(\diamondsuit) = 1 \quad \mathscr{P}(\star) = 0$

$\mathscr{P}(\heartsuit) = 1 \quad \mathscr{P}(\clubsuit) = 1$



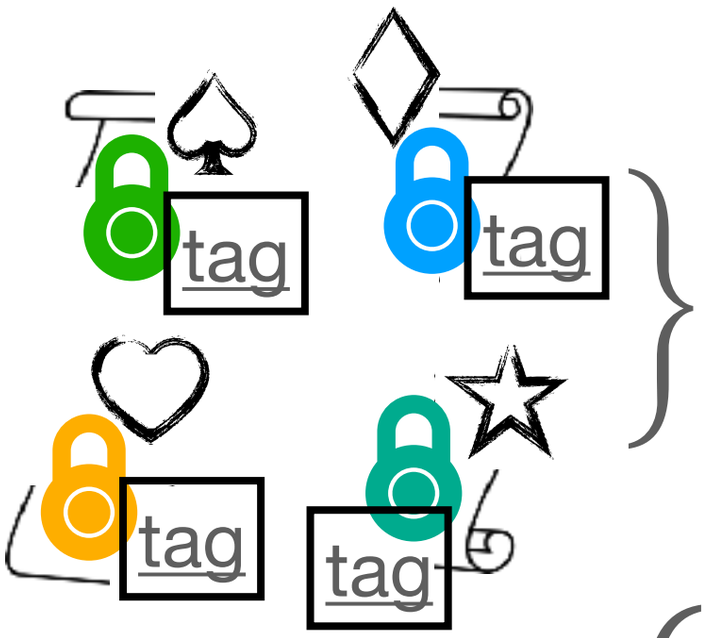
$$\begin{cases} \mathrm{F}(\quad) = \langle \quad, \vec{y} \rangle \\ \mathscr{P}\text{: } \textit{Linear Secret Sharing Schemes} \text{ (LSSS)} \end{cases}$$

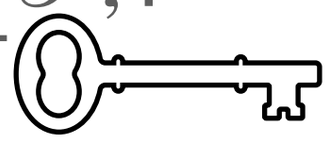
Motivation

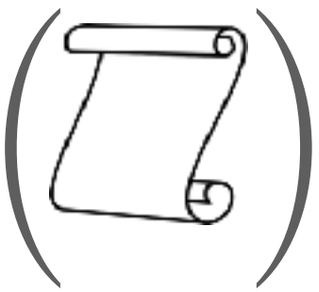
Controlling Decryption Keys in Multi-Client FE (MCFE)

$\mathscr{P}(\spadesuit) = 1 \quad \mathscr{P}(\diamondsuit) = 1 \quad \mathscr{P}(\star) = 0$

$\mathscr{P}(\heartsuit) = 1 \quad \mathscr{P}(\clubsuit) = 1$



$+ \; \mathscr{P}, \mathsf{F}$ 

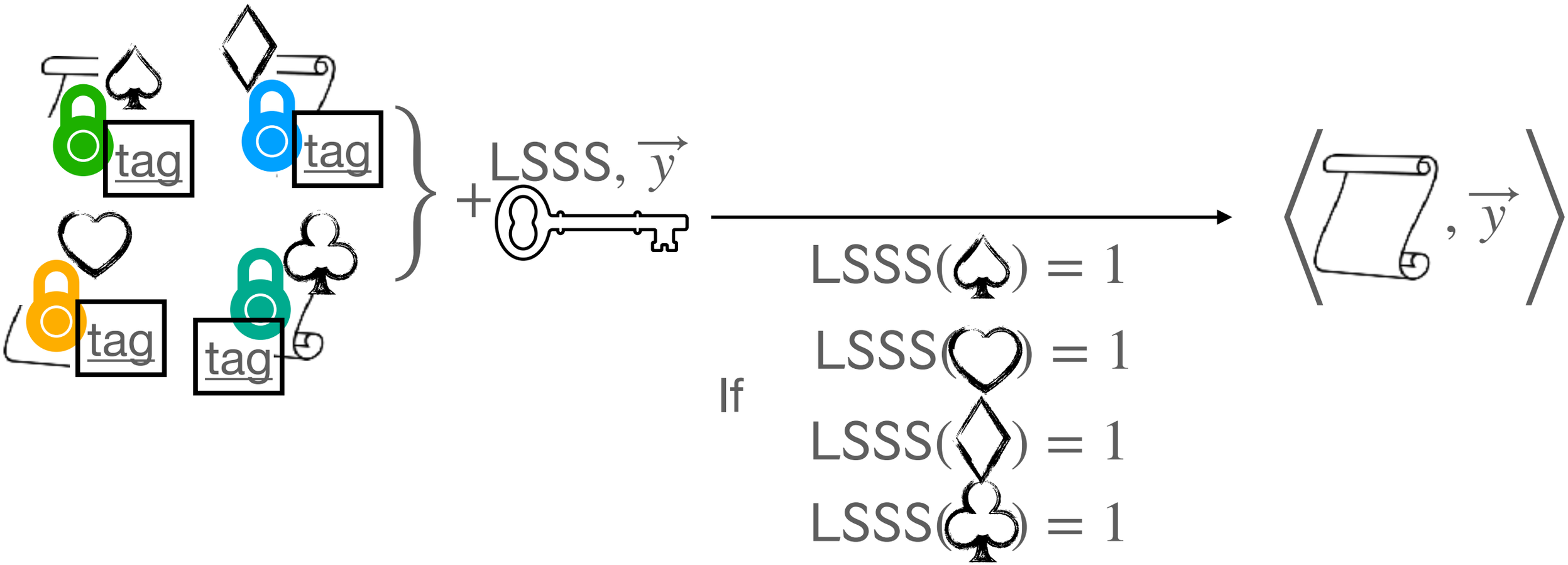
$\mathsf{F}\left(\text{  }\right)$

$$\begin{cases} \mathsf{F}\left(\text{  }\right) = \left\langle \text{  }, \vec{y} \right\rangle \\ \mathscr{P}\text{: } \textit{Linear Secret Sharing Schemes} \text{ (LSSS)} \end{cases}$$

Included in general MCFE if $\mathsf{F}(\cdot)$ can express the policy $\mathscr{P}$

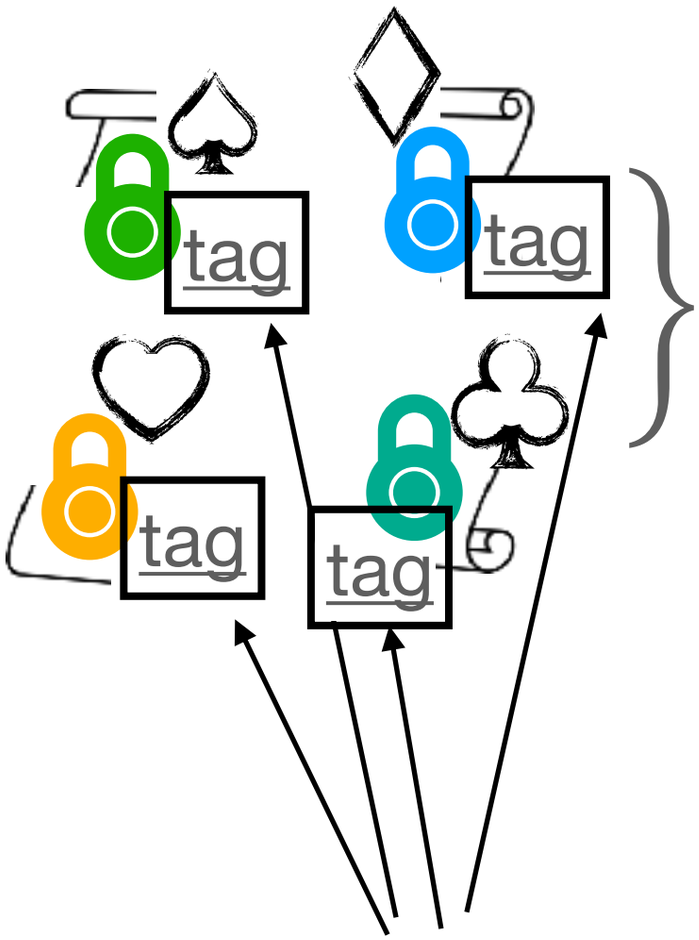
Motivation

Controlling Decryption Keys - Why MCFE? 🤔



Motivation

Controlling Decryption Keys - Why MCFE? 🤔



$+$ LSSS, $\vec{y}$

$\text{LSSS}(\spadesuit) = 1$

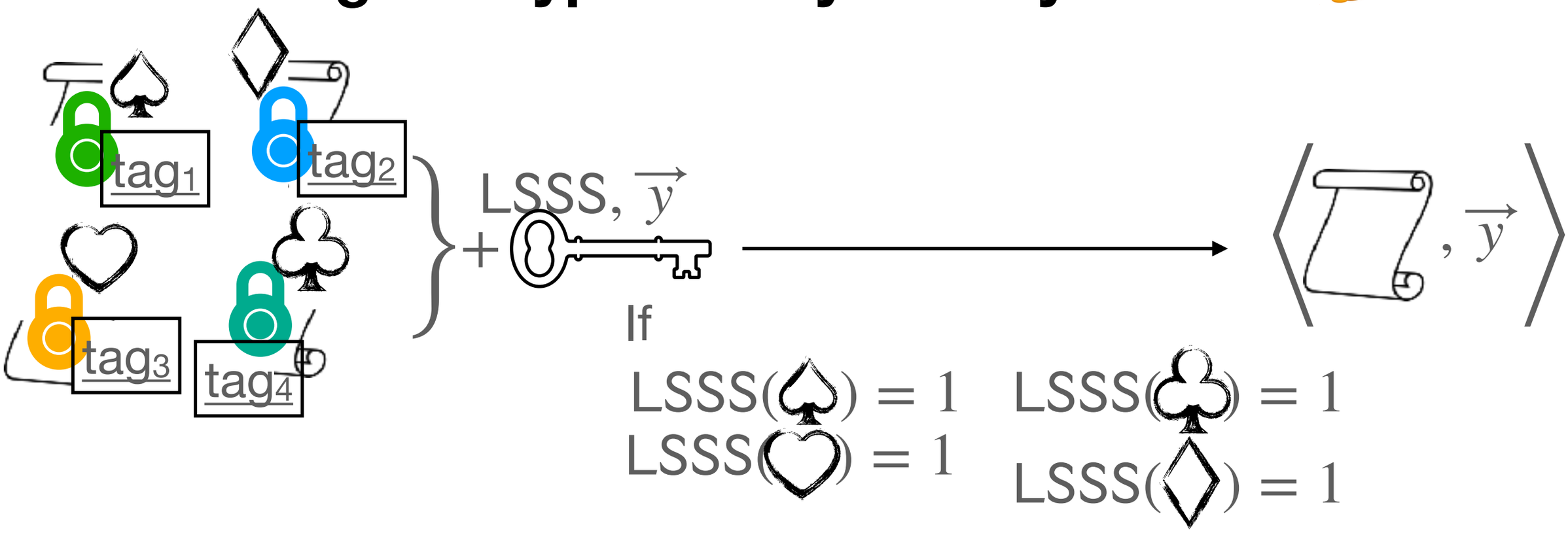
If $\text{LSSS}(\heartsuit) = 1$, $\text{LSSS}(\diamondsuit) = 1$, $\text{LSSS}(\clubsuit) = 1$

$\langle \quad , \vec{y} \rangle$

Hashed by H(tag) ⇒ Fixing and publishing H(tag), we obtain MIFE

Motivation

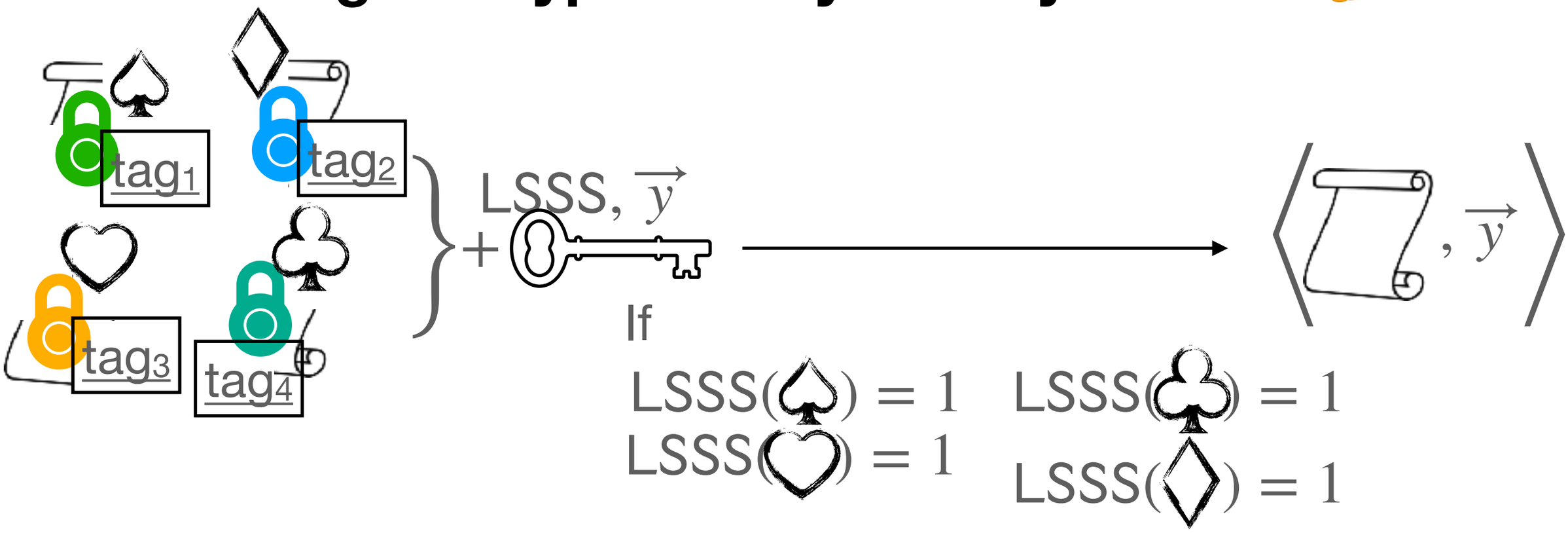
Controlling Decryption Keys - Why MCFE? 🤔



MIFE for
$\langle \cdot, \cdot \rangle$ with LSSS
$\Rightarrow$ MCFE for
$\langle \cdot, \cdot \rangle$with LSSS?

Motivation

Controlling Decryption Keys - Why MCFE? 🤔

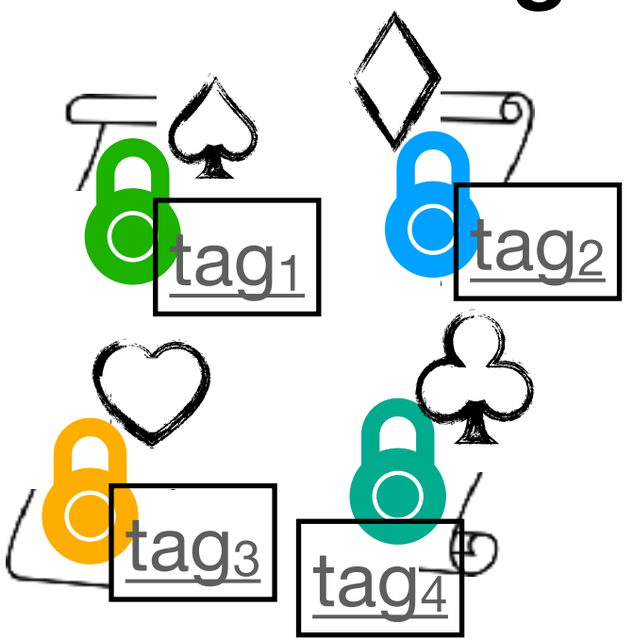


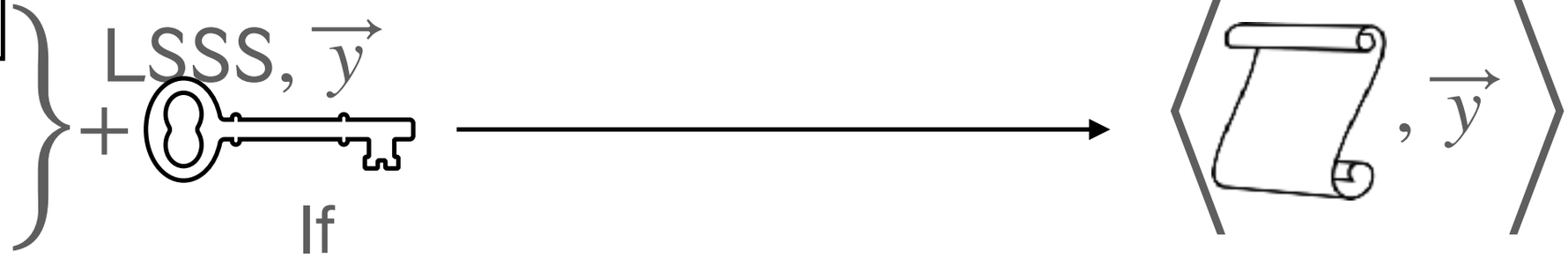
$\underline{\text{tag}_1}, \underline{\text{tag}_2}, \underline{\text{tag}_3}, \underline{\text{tag}_4}$
in ciphertext

MIFE for
$\langle \cdot, \cdot \rangle$ with LSSS
$\Rightarrow$ MCFE for
$\langle \cdot, \cdot \rangle$with LSSS?

Motivation

Controlling Decryption Keys - Why MCFE? 🤔





If

$\text{LSSS}(\spadesuit) = 1$ $\quad \text{LSSS}(\clubsuit) = 1$

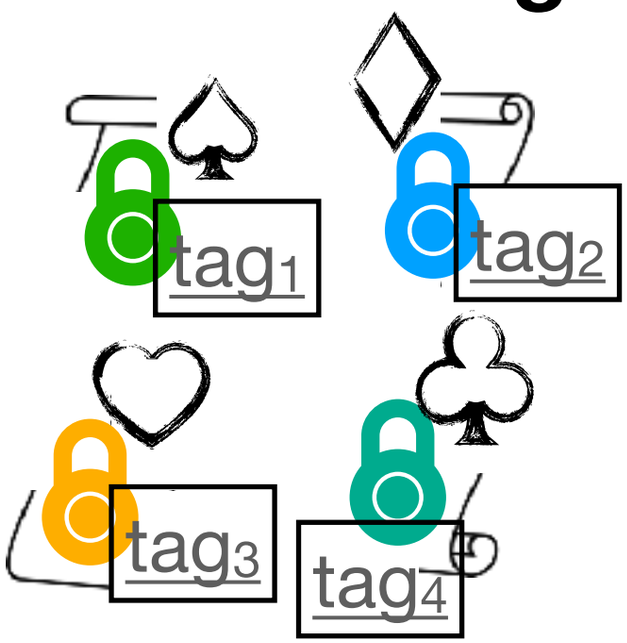
$\text{LSSS}(\heartsuit) = 1$ $\quad \text{LSSS}(\diamondsuit) = 1$

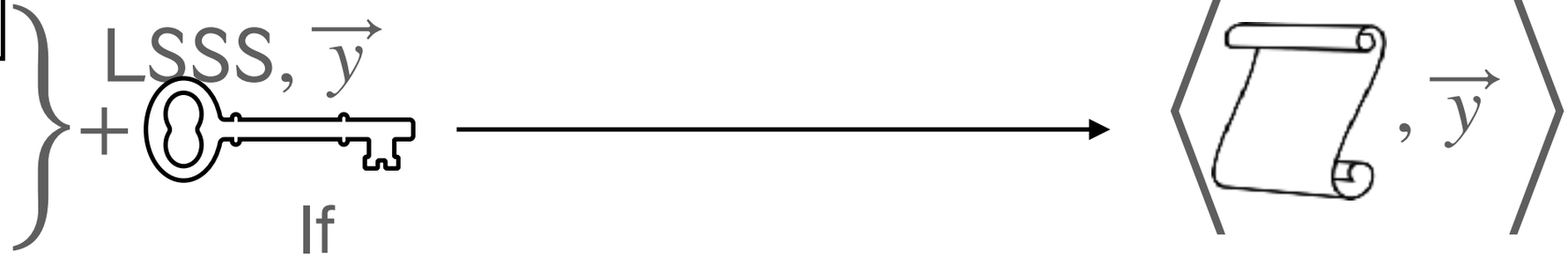
MIFE for
$\langle \cdot , \cdot \rangle$ with LSSS
$\Rightarrow$ MCFE for
$\langle \cdot , \cdot \rangle$with LSSS?

$\underline{\text{tag}_1}, \underline{\text{tag}_2}, \underline{\text{tag}_3}, \underline{\text{tag}_4}$ in ciphertext $\longrightarrow$ Must check that $\underline{\text{tag}_1} = \underline{\text{tag}_2} = \underline{\text{tag}_3} = \underline{\text{tag}_4}$

Motivation

Controlling Decryption Keys - Why MCFE? 🤔





If

$\text{LSSS}(\spadesuit) = 1$ $\text{LSSS}(\clubsuit) = 1$

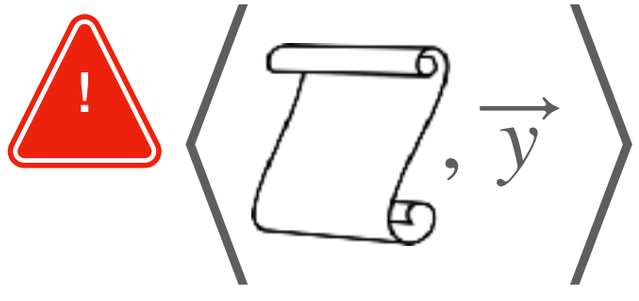
$\text{LSSS}(\heartsuit) = 1$ $\text{LSSS}(\diamondsuit) = 1$

MIFE for $\langle \cdot , \cdot \rangle$ with LSSS

~~⇒~~ MCFE for $\langle \cdot , \cdot \rangle$ with LSSS?

$\underline{tag_1}, \underline{tag_2}, \underline{tag_3}, \underline{tag_4}$ in ciphertext → Must check that $\underline{tag_1} = \underline{tag_2} = \underline{tag_3} = \underline{tag_4}$

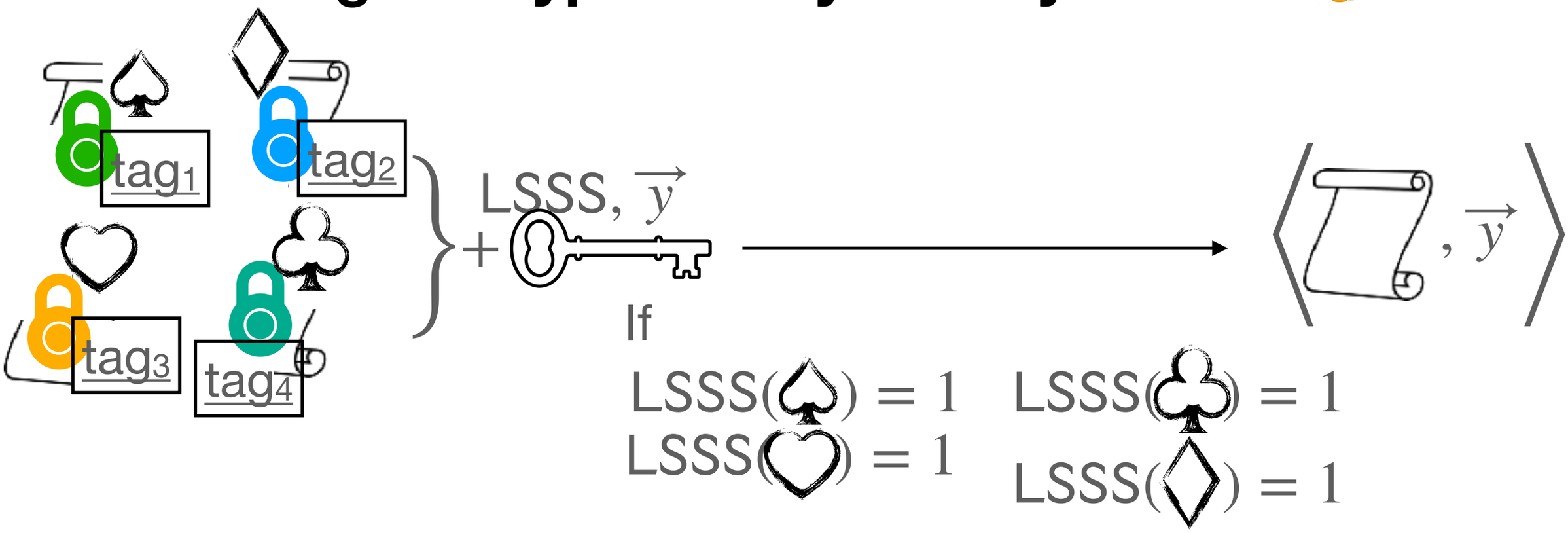
Infeasible in this case of



and LSSS

Motivation

Controlling Decryption Keys - Why MCFE? 🤔



MIFE for $\langle \cdot, \cdot \rangle$ with LSSS

~~⇒~~ MCFE for $\langle \cdot, \cdot \rangle$with LSSS?

$\underline{tag_1}, \underline{tag_2}, \underline{tag_3}, \underline{tag_4}$ in ciphertext → Must check that $\underline{tag_1} = \underline{tag_2} = \underline{tag_3} = \underline{tag_4}$

Infeasible in this case of $\langle \cdot, \vec{y} \rangle$ and LSSS

More details in Sect. 3.1 of our paper

Our Goal

Constructing Multi-Client Functional Encryption Schemes for Inner Products, With Access Control using LSSS

Related Works and Our Contributions

FE + Access Control in the Multi-User Regime

[ACGU20]

- From pairings and LWE
- $\mathscr{P}$: *Monotone Span Programs*
- F: *Inner Products*
- (Multi-Input) Generic from single-client, **adaptive** security in standard model
- **Quadratic** total communication

Related Works and Our Contributions

FE + Access Control in the Multi-User Regime

[ACGU20]

- From pairings and LWE
- $\mathscr{P}$: *Monotone Span Programs*
- F: *Inner Products*
- (Multi-Input) Generic from single-client, **adaptive** security in standard model
- **Quadratic** total communication

Our work

- From pairings
- $\mathscr{P}$: *LSSS*
- F: *Inner Products*
- (Multi-Client) **Adaptive** security in ROM
  linear total communication

Related Works and Our Contributions

FE + Access Control in the Multi-User Regime

[ACGU20]

- From pairings and LWE
- $\mathscr{P}$: *Monotone Span Programs*
- F: *Inner Products*
- (Multi-Input) Generic from single-client, **adaptive** security in standard model
- **Quadratic** total communication

Our work

- From pairings
- $\mathscr{P}$: *LSSS*
- F: *Inner Products*
- (Multi-Client) **Adaptive** security in ROM
  linear total communication
  ⇒ (Multi-Input) **adaptive** security in standard model, **linear** total comm.

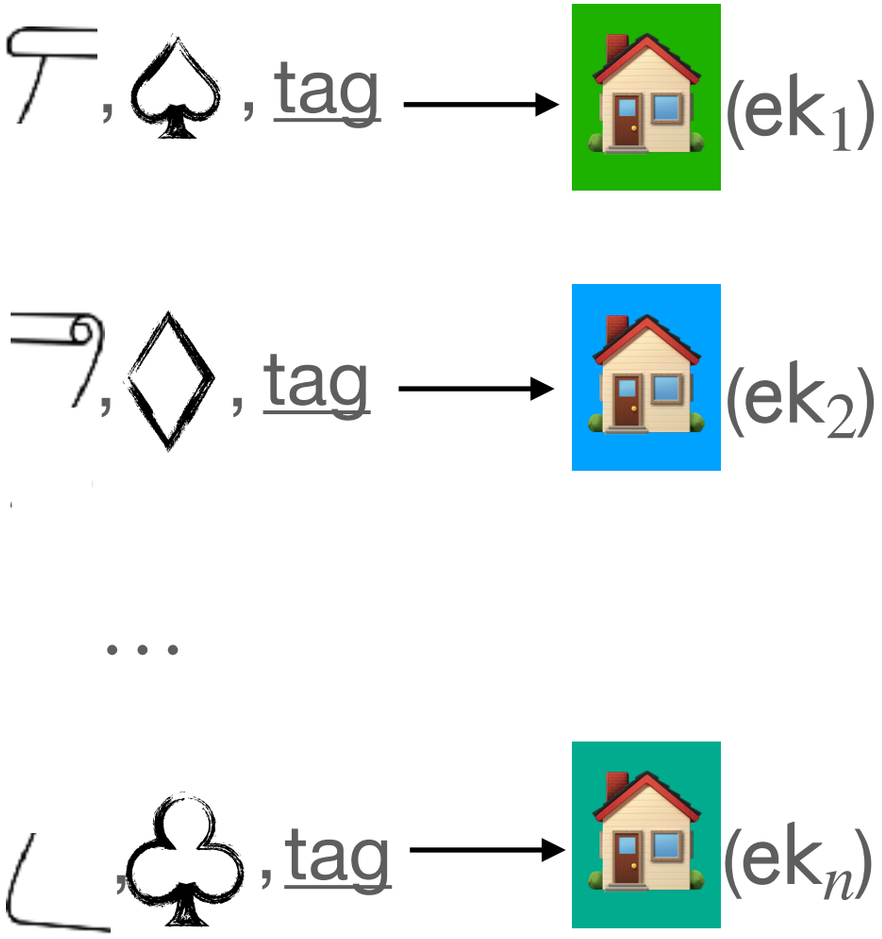
Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) (♠) (♢) (♣)

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$

$\mathscr{P}, \overrightarrow{y}$ ⟶ ⚖(msk)

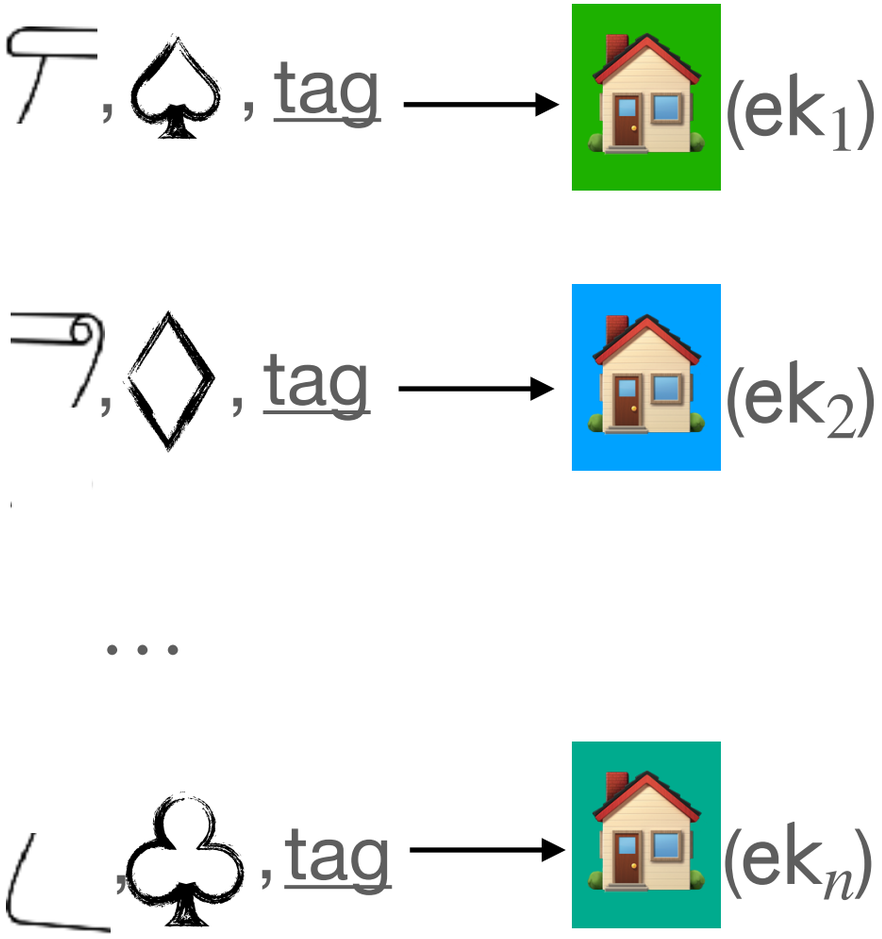


Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) (♠) (♢) (♣)

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$

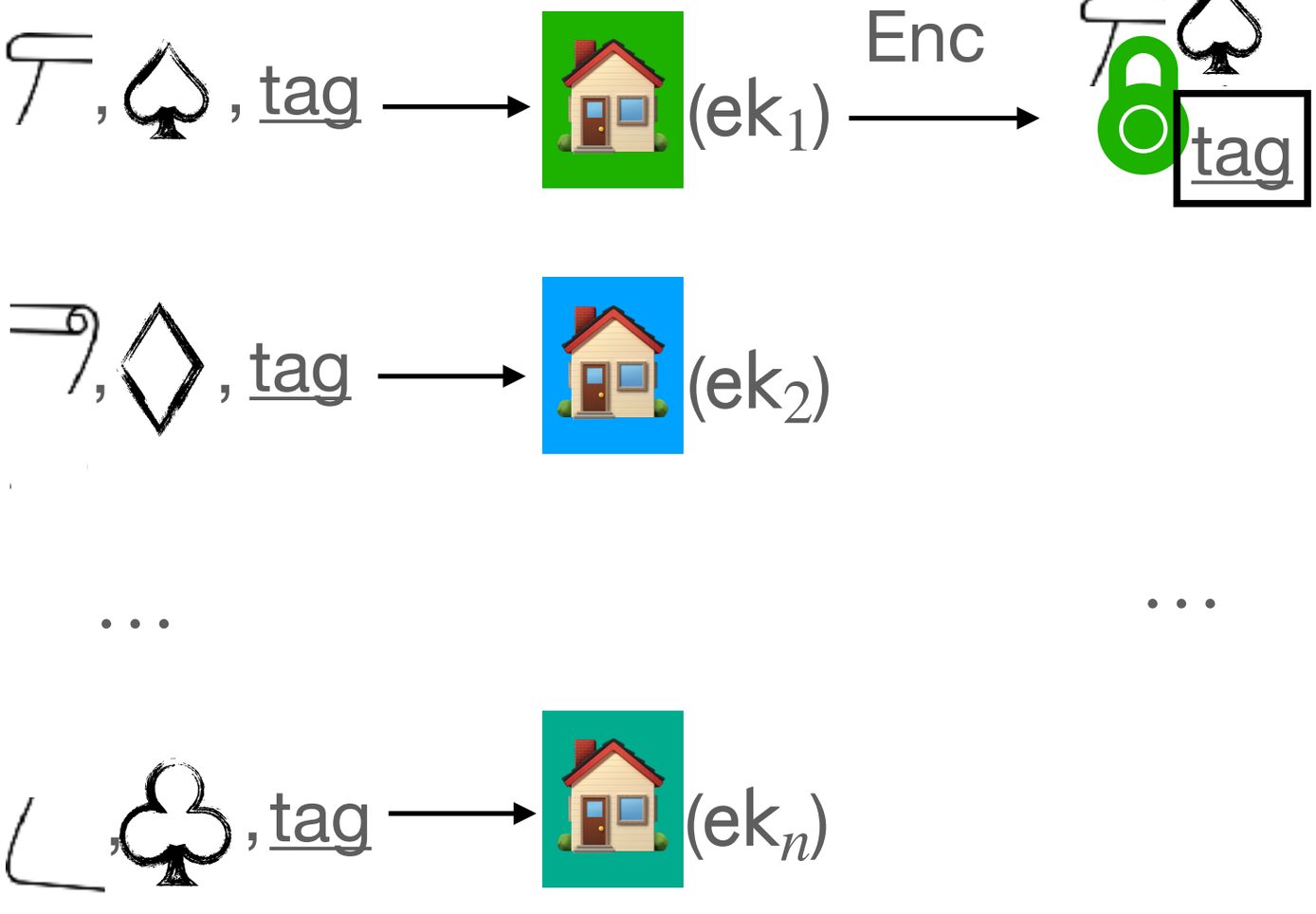


Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) (♠) (♢) (♣)

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$

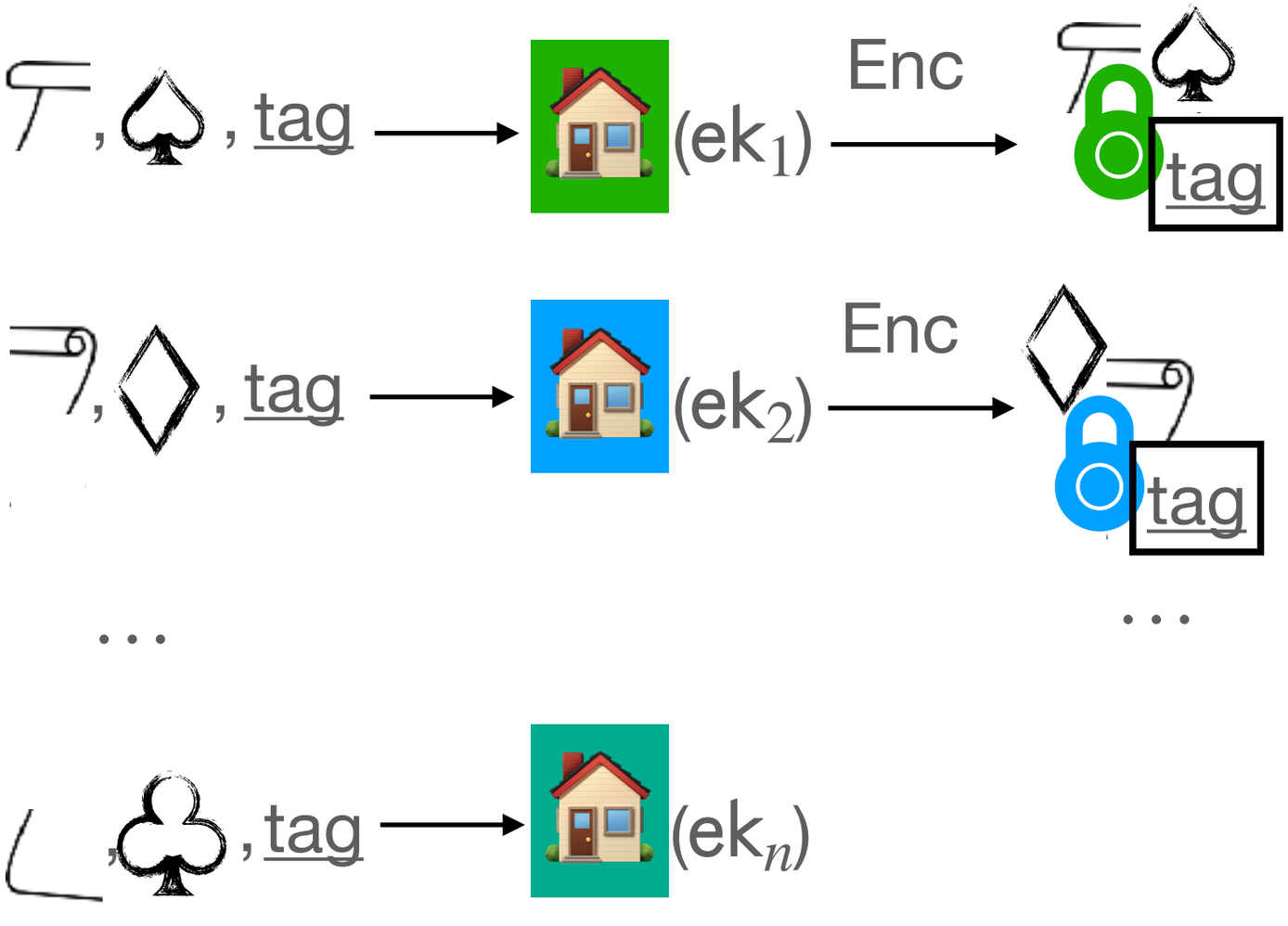


Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) (♠) (♢) (♣)

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$

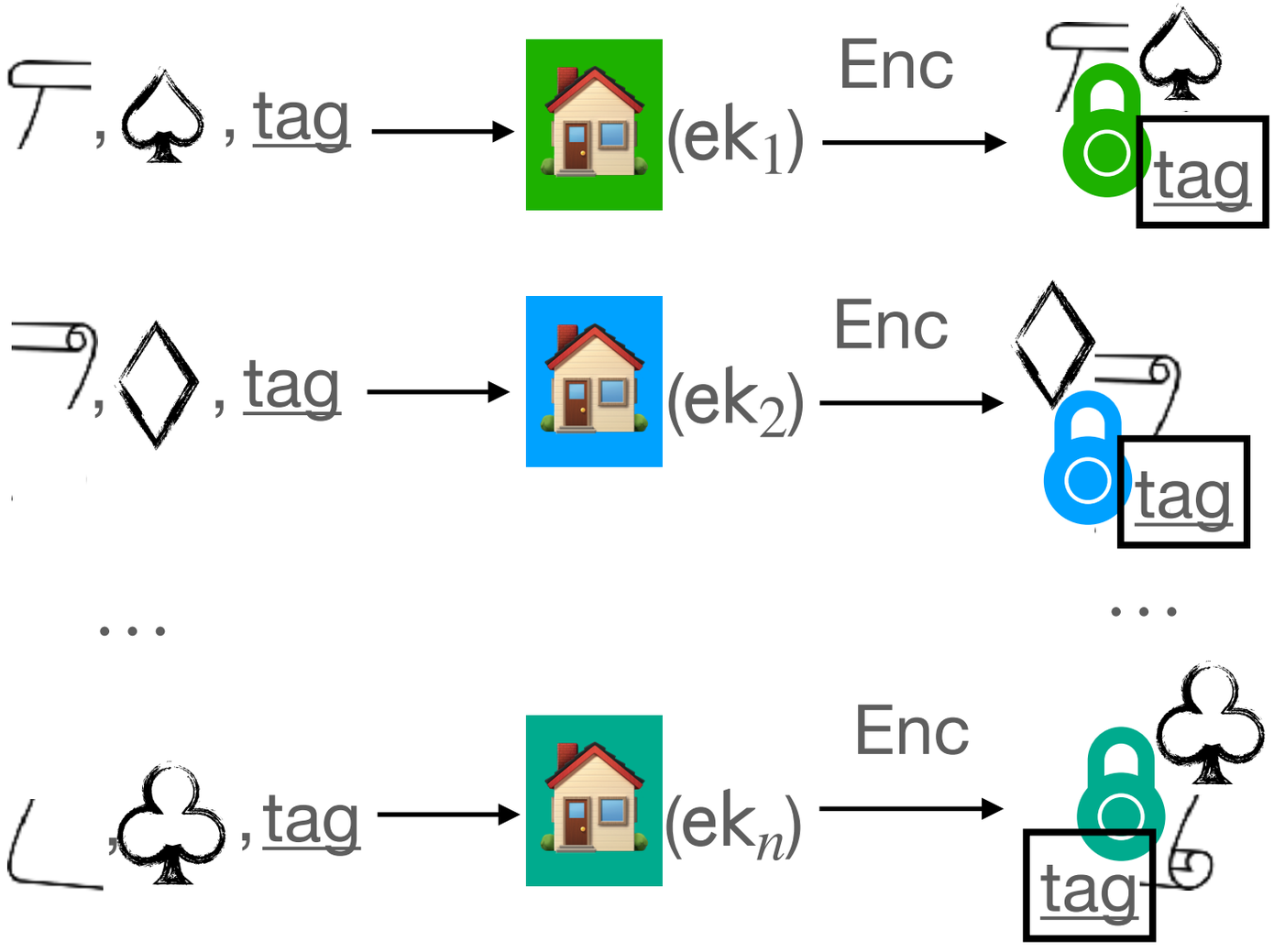


Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) (♠) (♢) (♣)

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$

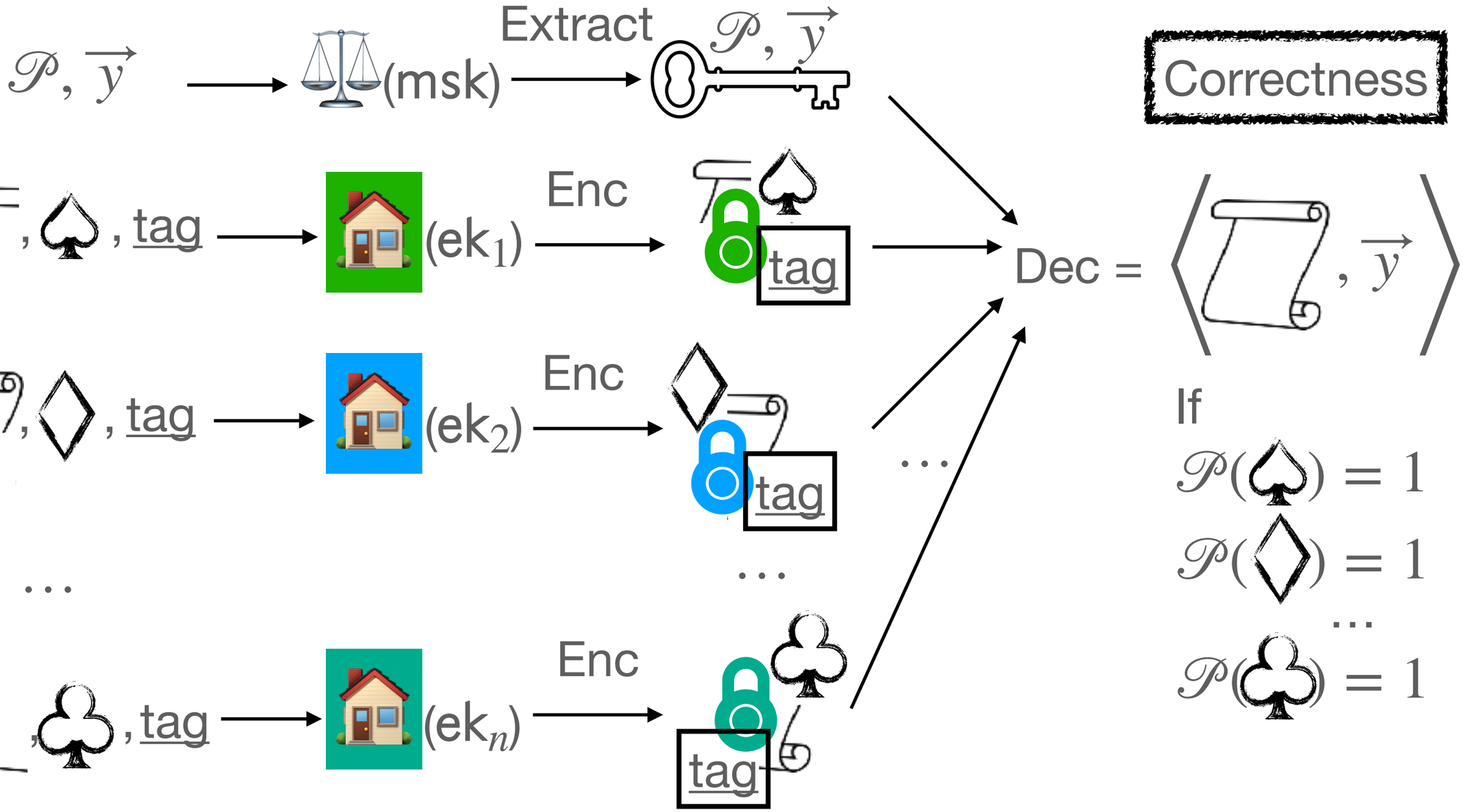


Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS) $(\spadesuit)$ $(\diamondsuit)$ $(\clubsuit)$

Access Control: AC-K $\times$ AC-Ct$_1$ $\times$ AC-Ct$_2$ $\times \cdots \times$ AC-Ct$_n$ $\rightarrow \{0,1\}$



Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

(msk)

(ek$_1$)

…

(ek$_i$)

(ek$_2$)

Adversary

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

(msk)

Extract($\mathscr{P}$, $\overrightarrow{y}$)

(ek_1)

...

(ek_i)

(ek_2)

Adversary

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

(msk)

(ek$_1$)

...

(ek$_2$)

(ek$_i$)

Extract($\mathscr{P}$, $\overrightarrow{y}$)

Corrupt(i)

ek$_i$

Adversary

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

Extract($\mathscr{P}$, $\overrightarrow{y}$)

Corrupt(i)

ek_i

Chall($i, x_{0,i}, x_{1,i}, tag,$ ♡)

Enc($x_{b,i}, tag,$ ♡)

Adversary

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

Extract($\mathscr{P}$, $\overrightarrow{y}$)

Corrupt(i)

ek_i

Chall($i, x_{0,i}, x_{1,i}, tag,$ ♡)

Enc($x_{b,i}, tag,$ ♡)

$(i, x_i, tag',$ ♡)

Enc($x_i, tag',$ ♡)

Adversary

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

$b' \in \{0,1\}$

Adversary

(msk)

Extract($\mathscr{P}$, $\overrightarrow{y}$)

(ek_1)

...

(ek_i)

(ek_2)

Corrupt(i)

ek_i

Chall($i, x_{0,i}, x_{1,i}, tag,$ ♡)

Enc($x_{b,i}, tag,$ ♡)

($i, x_i, tag',$ ♡)

Enc($x_i, tag',$ ♡)

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

$b' \in \{0,1\}$

Adversary

- **AD**/SEL Chall
- **DYN**/STAT Corruption

(msk)

Extract($\mathscr{P}$, $\overrightarrow{y}$)

(ek_1)

(ek_i)

...

(ek_2)

Corrupt(i)

ek_i

Chall($i, x_{0,i}, x_{1,i}, tag,$ ♡)

Enc($x_{b,i}, tag,$ ♡)

($i, x_i, tag',$ ♡)

Enc($x_i, tag',$ ♡)

Definition

MCFE with Fine-Grained Access Control

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

Challenger: $b \leftarrow^{\$} \{0,1\}$

Security

$b' \in \{0,1\}$

Adversary

(msk)

Extract($\mathscr{P}$, $\vec{y}$)

(ek_1)

...

(ek_i)

(ek_2)

Corrupt(i)

ek_i

Chall($i, x_{0,i}, x_{1,i}, tag,$ ♡)

Enc($x_{b,i}, tag,$ ♡)

$(i, x_i, tag',$ ♡)

Enc($x_i, tag',$ ♡)

$\mathscr{P}, \vec{y}$

$\mathscr{P}$ allows decrypting

$\Rightarrow \langle \vec{x}_0 - \vec{x}_1, \vec{y} \rangle = 0$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathscr{B} = \begin{bmatrix} \\ \\ \\ \dots \\ \\ \\ \end{bmatrix} \in GL_N(\mathbb{Z}_q)$$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathscr{B} = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \end{bmatrix} ; \left(\mathscr{B}^{-1}\right)^{\dagger} = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \end{bmatrix} \in GL_N(\mathbb{Z}_q)$$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathscr{B} = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \\ \end{bmatrix} ; \left(\mathscr{B}^{-1}\right)^{\dagger} = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \\ \end{bmatrix} \in GL_N(\mathbb{Z}_q)$$

$\cdot g_1$ $\cdot g_2$

$$\mathbf{B} = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \\ \end{bmatrix}_1 \in \mathbb{G}_1^{N \times N} ; \quad \mathbf{B}^* = \begin{bmatrix} \\ \\ \\ \cdots \\ \\ \\ \\ \end{bmatrix}_2 \in \mathbb{G}_2^{N \times N}$$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathbf{B} = \begin{bmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{bmatrix}_1 ; \quad \mathbf{B}^* = \begin{bmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{bmatrix}_2$$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathbf{B} = \begin{bmatrix} \\ \\ \\ \dots \\ \\ \\ \end{bmatrix}_1 ; \quad \mathbf{B}^* = \begin{bmatrix} \\ \\ \\ \dots \\ \\ \\ \end{bmatrix}_2$$

$$(x_1, .., x_n) \cdot \mathbf{B}$$

$$(x_1 \mid \dots \mid x_n)_{\mathbf{B}} \in \mathbb{G}_1^N$$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathbf{B} = \begin{bmatrix} \cdots \\ \cdots \end{bmatrix}_1 \quad ; \quad \mathbf{B}^* = \begin{bmatrix} \cdots \\ \cdots \end{bmatrix}_2$$

$(x_1, .., x_n) \cdot$ ↓ $\quad (x_1 \mid \ldots \mid x_n)_{\mathbf{B}} \in \mathbb{G}_1^N$

$(y_1, .., y_n) \cdot$ ↓ $\quad (y_1 \mid \ldots \mid y_n)_{\mathbf{B}^*} \in \mathbb{G}_2^N$

Technical Tools

Dual Pairing Vector Spaces [Okamoto,Takashima'10,12]

Prime-order (additive) bilinear group: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e, q), N \in \mathbb{N}$

Public vectors Private vectors

$$\mathbf{B} = \begin{bmatrix} \cdots \\ \cdots \end{bmatrix}_1 ; \quad \mathbf{B}^* = \begin{bmatrix} \cdots \\ \cdots \end{bmatrix}_2$$

$(x_1, .., x_n) \cdot$ $(y_1, .., y_n) \cdot$

$(x_1 \mid \ldots \mid x_n)_{\mathbf{B}} \in \mathbb{G}_1^N$ $(y_1 \mid \ldots \mid y_n)_{\mathbf{B}^*} \in \mathbb{G}_2^N$

$\times$

$[\![x_1 y_1 + \cdots + x_n y_n]\!]_t$

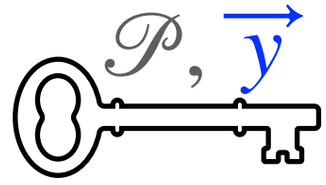
Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P} : \text{LSSS})$

$\text{AC-K} \times \text{AC-Ct} \rightarrow \{0,1\}$

$= (x_1, \ldots, x_n)$



$\mathscr{P}, \overrightarrow{y}$

$S = (s_1, .., s_n)$

$U = (u_1, .., u_n)$

$$\text{Ciphertext} \begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \\ \overrightarrow{c}_{att} = (\cdots | \psi | 0 | \cdots)_{\mathbf{F}} \quad ; \quad \overrightarrow{c}_0 = (\omega | \mu\omega | \psi | 0)_{\mathbf{H}} \end{cases}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | 0 | \cdots)_{\mathbf{F}^*}; \quad \overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle | \langle U, \overrightarrow{y} \rangle | a_0 | 0)_{\mathbf{H}^*}$$

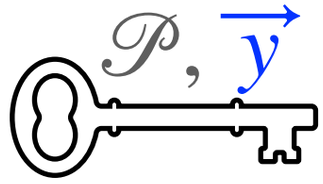
Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P} : \text{LSSS})$

$\text{AC-K} \times \text{AC-Ct} \to \{0,1\}$

$= (x_1, \ldots, x_n)$



$S = (s_1, .., s_n)$

$U = (u_1, .., u_n)$

$$\text{Ciphertext} \begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \\ \overrightarrow{c}_{att} = (\cdots | \psi | 0 | \cdots)_{\mathbf{F}} \quad ; \quad \overrightarrow{c}_0 = (\omega | \mu\omega | \psi | 0)_{\mathbf{H}} \end{cases}$$

Key-Policy, $(a_{att})_{att} \leftarrow \text{LSSS-Share}(a_0)$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | 0 | \cdots)_{\mathbf{F}^*}; \quad \overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle | \langle U, \overrightarrow{y} \rangle | a_0 | 0)_{\mathbf{H}^*}$$

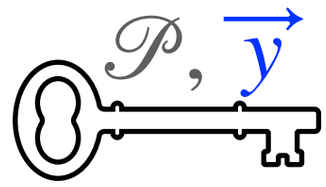
Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P} : \text{LSSS})$

$\text{AC-K} \times \text{AC-Ct} \to \{0,1\}$

$= (x_1, \ldots, x_n)$



$S = (s_1, .., s_n)$

$U = (u_1, .., u_n)$

Randomness

Ciphertext $\begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \\ \overrightarrow{c}_{att} = (\cdots \mid \psi \mid 0 \mid \cdots)_{\mathbf{F}} \quad ; \end{cases}$

$\overrightarrow{c}_0 = (\omega \mid \mu\omega \mid \psi \mid 0)_{\mathbf{H}}$

Key-Policy, $(a_{att})_{att} \leftarrow \text{LSSS-Share}(a_0)$

$\overrightarrow{k}^*_{att} = (\cdots \mid a_{att} \mid 0 \mid \cdots)_{\mathbf{F}^*}; \qquad \overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle \mid \langle U, \overrightarrow{y} \rangle \mid a_0 \mid 0)_{\mathbf{H}^*}$

ALS key

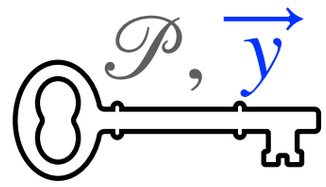
Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P} : \text{LSSS})$

$\text{AC-K} \times \text{AC-Ct} \to \{0,1\}$

$= (x_1, \ldots, x_n)$



$S = (s_1, .., s_n)$

$U = (u_1, .., u_n)$

Randomness

Ciphertext $\begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \\ \overrightarrow{c}_{att} = (\cdots \mid \psi \mid 0 \mid \cdots)_{\mathbf{F}} \quad ; \end{cases}$

$\overrightarrow{c}_0 = (\omega \mid \mu\omega \mid \psi \mid 0)_{\mathbf{H}}$

Key-Policy, $(a_{att})_{att} \leftarrow \text{LSSS-Share}(a_0)$

$\overrightarrow{k}^*_{att} = (\cdots \mid a_{att} \mid 0 \mid \cdots)_{\mathbf{F}^*}$;

$\overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle \mid \langle U, \overrightarrow{y} \rangle \mid a_0 \mid 0)_{\mathbf{H}^*}$

$$\Rightarrow \sum_{att \in \text{LSSS-Reconstr}} \overrightarrow{c}_{att} \times \overrightarrow{k}^*_{att}$$

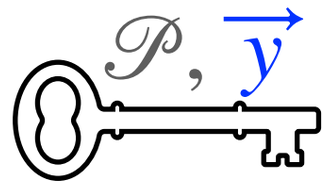
$$= [\![\sum_{att \in \text{LSSS-Reconstr}} \psi a_{att}]\!]_t = [\![\psi a_0]\!]_t$$

ALS key

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P}$: LSSS)
AC-K × AC-Ct → $\{0,1\}$


$= (x_1, \ldots, x_n)$


$S = (s_1, .., s_n)$
$U = (u_1, .., u_n)$

Randomness

Ciphertext $\begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \\ \overrightarrow{c}_{att} = (\cdots \mid \psi \mid 0 \mid \cdots)_{\mathbf{F}} \quad ; \end{cases}$

$\overrightarrow{c}_0 = (\omega \mid \mu\omega \mid \psi \mid 0)_{\mathbf{H}}$

Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$\overrightarrow{k}^*_{att} = (\cdots \mid a_{att} \mid 0 \mid \cdots)_{\mathbf{F}^*}$; $\overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle \mid \langle U, \overrightarrow{y} \rangle \mid a_0 \mid 0)_{\mathbf{H}^*}$

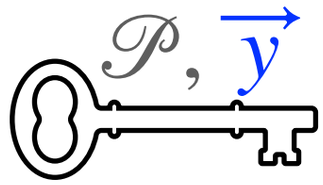
$$\Rightarrow \sum_{att \in \text{LSSS-Reconstr}} \overrightarrow{c}_{att} \times \overrightarrow{k}^*_{att}$$
$$= [\![\sum_{att \in \text{LSSS-Reconstr}} \psi a_{att}]\!]_t = [\![\psi a_0]\!]_t$$

$$\Rightarrow \overrightarrow{c}_0 \times \overrightarrow{k}^*_0$$
$$= [\![\omega \langle S + \mu U, \overrightarrow{y} \rangle]\!]_t + [\![\psi a_0]\!]_t$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P}$: LSSS)
AC-K × AC-Ct → $\{0,1\}$


$= (x_1, \ldots, x_n)$


$S = (s_1, .., s_n)$
$U = (u_1, .., u_n)$

Ciphertext $\begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \xrightarrow{\cdot\, y_i} \sum_{i=1}^{n} \xrightarrow{} [\![\omega \langle S + \mu U, \overrightarrow{y} \rangle]\!]_t + [\![\langle \overrightarrow{x}, \overrightarrow{y} \rangle]\!]_t \\ \overrightarrow{c}_{att} = (\cdots \mid \psi \mid 0 \mid \cdots)_{\mathbf{F}} \quad ; \quad \overrightarrow{c}_0 = (\omega \mid \mu\omega \mid \psi \mid 0)_{\mathbf{H}} \end{cases}$

Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$$\overrightarrow{k}^*_{att} = (\cdots \mid a_{att} \mid 0 \mid \cdots)_{\mathbf{F}^*}; \qquad \overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle \mid \langle U, \overrightarrow{y} \rangle \mid a_0 \mid 0)_{\mathbf{H}^*}$$

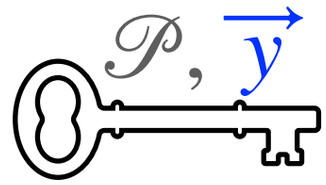
$$\Rightarrow \sum_{att \in \text{LSSS-Reconstr}} \overrightarrow{c}_{att} \times \overrightarrow{k}^*_{att}$$
$$= [\![\sum_{att \in \text{LSSS-Reconstr}} \psi a_{att}]\!]_t = [\![\psi a_0]\!]_t$$

$$\Rightarrow \overrightarrow{c}_0 \times \overrightarrow{k}^*_0$$
$$= [\![\omega \langle S + \mu U, \overrightarrow{y} \rangle]\!]_t + [\![\psi a_0]\!]_t$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

$(\mathscr{P}$: LSSS)
$\text{AC-K} \times \text{AC-Ct} \to \{0,1\}$


$= (x_1, \ldots, x_n)$


$S = (s_1, .., s_n)$
$U = (u_1, .., u_n)$

Ciphertext $\begin{cases} \left([\![\omega(s_i + \mu u_i) + x_i]\!]_1 \right)_{i \in [n]} \xrightarrow{\cdot\, y_i} \sum_{i=1}^{n} \xrightarrow{} [\![\omega \langle S + \mu U, \overrightarrow{y} \rangle]\!]_t + [\![\langle \overrightarrow{x}, \overrightarrow{y} \rangle]\!]_t \\ \overrightarrow{c}_{att} = (\cdots \mid \psi \mid 0 \mid \cdots)_{\mathbf{F}} \quad ; \quad \overrightarrow{c}_0 = (\omega \mid \mu\omega \mid \psi \mid 0)_{\mathbf{H}} \end{cases}$

Key-Policy, $(a_{att})_{att} \leftarrow \text{LSSS-Share}(a_0)$

$$\overrightarrow{k}^*_{att} = (\cdots \mid a_{att} \mid 0 \mid \cdots)_{\mathbf{F}^*}; \qquad \overrightarrow{k}^*_0 = (\langle S, \overrightarrow{y} \rangle \mid \langle U, \overrightarrow{y} \rangle \mid a_0 \mid 0)_{\mathbf{H}^*}$$

$$\Rightarrow \sum_{att \in \text{LSSS-Reconstr}} \overrightarrow{c}_{att} \times \overrightarrow{k}^*_{att}$$
$$= [\![\sum_{att \in \text{LSSS-Reconstr}} \psi a_{att}]\!]_t = [\![\psi a_0]\!]_t$$

$$\Rightarrow \overrightarrow{c}_0 \times \overrightarrow{k}^*_0$$
$$= [\![\omega \langle S + \mu U, \overrightarrow{y} \rangle]\!]_t + [\![\psi a_0]\!]_t$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K $\times$ AC-Ct $\rightarrow \{0,1\}$, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

Random z_{att}

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

Random z_{att}

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

$(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can't decrypt: ABE technique

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

Random z_{att}

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

ok $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can't decrypt: ABE technique

! $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can decrypt $\Rightarrow \langle \Delta x, \overrightarrow{y} \rangle = 0$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

Random z_{att}

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

ok $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can't decrypt: ABE technique ← ? **Adaptive Security** 🤔

! $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can decrypt $\Rightarrow \langle \Delta x, \overrightarrow{y} \rangle = 0$ ← ?

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s_i + \mu u_i) + x_{b,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

Random z_{att}

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | a'_0 y_1 | \cdots | a'_0 y_n)_{\mathbf{H}^*}$$

ok $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can't decrypt: ABE technique ← ?

! $(\overrightarrow{k}^*_{att}, \overrightarrow{k}_0)$ can decrypt $\Rightarrow \langle \Delta x, \overrightarrow{y} \rangle = 0$ ← ?

Adaptive Security 🤔

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s'_i + \mu u'_i) + x_{0,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | r'_0 y_1 | \cdots | r'_0 y_n)_{\mathbf{H}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s'_i + \mu u'_i) + x_{0,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau\Delta x_1 z_{att} | \cdots | \tau\Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | r'_0 y_1 | \cdots | r'_0 y_n)_{\mathbf{H}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau\Delta x_1 | \cdots | \tau\Delta x_n)_{\mathbf{H}}$$

n coordinates

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s'_i + \mu u'_i) + x_{0,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | r'_0 y_1 | \cdots | r'_0 y_n)_{\mathbf{H}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

$$|\text{ciphertext}| = nd + 2n + 7d + 3$$

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s'_i + \mu u'_i) + x_{0,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | r'_0 y_1 | \cdots | r'_0 y_n)_{\mathbf{H}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

Vector's length

$$|\text{ciphertext}| = nd + 2n + 7d + 3$$

#att's needed in ciphertext

Technical Overview - Single-Client

AB-IPFE for LSSS - Adaptive Security, linear ciphertext's size

AC-K × AC-Ct → {0,1}, Key-Policy, $(a_{att})_{att} \leftarrow$ LSSS-Share(a_0)

$[\![\omega(s'_i + \mu u'_i) + x_{0,i}]\!]_1$, $\Delta x_i = x_{0,i} - x_{b,i}$, $(a'_{att})_{att} \leftarrow$ LSSS-Share(a'_0)

Changes using Basis changes In DPVS

$$\overrightarrow{c}_{att} = (\cdots | \psi | \tau \Delta x_1 z_{att} | \cdots | \tau \Delta x_n z_{att})_{\mathbf{F}}$$

$$\overrightarrow{k}^*_{att} = (\cdots | a_{att} | a'_{att} y_1 / z_{att} | \cdots | a'_{att} y_n / z_{att})_{\mathbf{F}^*}$$

$$\overrightarrow{k}^*_0 = (\cdots | a_0 | r'_0 y_1 | \cdots | r'_0 y_n)_{\mathbf{H}^*}$$

$$\overrightarrow{c}_0 = (\omega | \mu\omega | \psi | \tau \Delta x_1 | \cdots | \tau \Delta x_n)_{\mathbf{H}}$$

More details in Sect. 5.3 of our paper

Vector's length

$$|\text{ciphertext}| = nd + 2n + 7d + 3$$

#att's needed in ciphertext

E.g.: $d = 1$ for Identity-based Control

Towards Multi-Client - “Compress-and-Duplicate”

MC-AB-IPFE for LSSS - Adaptive Security, linear total communication

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$

More details in Sect. 5.2 of our paper

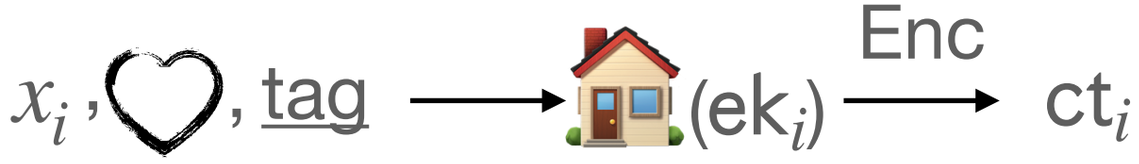
“Compress” in $(\mathbf{F}, \mathbf{F}^*)$

$$\vec{k}^*_{i,att} = (\cdots \mid a_{att} \mid a'_{att} y_i / z_{att} \mid 0..0)_{\mathbf{F}^*}$$

“Duplicate”$(\mathbf{H_i}, \mathbf{H_i^*})$

$$\vec{k}^*_{i,0} = (\cdots \mid a_0 \mid r'_0 y_i \mid 0..0)_{\mathbf{H_i^*}}$$

$\Theta(1)$ aux coordinates per i,att



“Compress” in $(\mathbf{F}, \mathbf{F}^*)$

“Duplicate”$(\mathbf{H_i}, \mathbf{H_i^*})$

$$\vec{c}_{i,att} = (\cdots \mid \psi_i \mid \tau \Delta x_i z_{att} \mid 0..0)_{\mathbf{F}}$$

$$\vec{c}_{i,0} = (\omega \mid \omega' \mid \psi_i \mid \tau \Delta x_i \mid 0..0)_{\mathbf{H_i}}$$

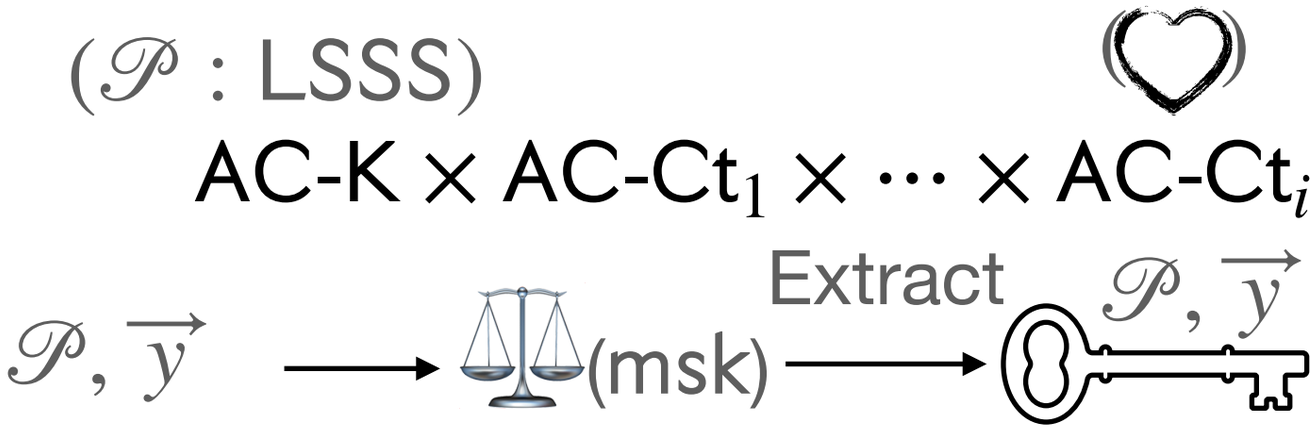
Hash(tag) as RO

Towards Multi-Client - “Compress-and-Duplicate”

MC-AB-IPFE for LSSS - Adaptive Security, linear total communication

$(\mathscr{P}$: LSSS)

$$\text{AC-K} \times \text{AC-Ct}_1 \times \cdots \times \text{AC-Ct}_i \times \cdots \times \text{AC-Ct}_n \to \{0,1\}$$



More details in Sect. 5.2 of our paper

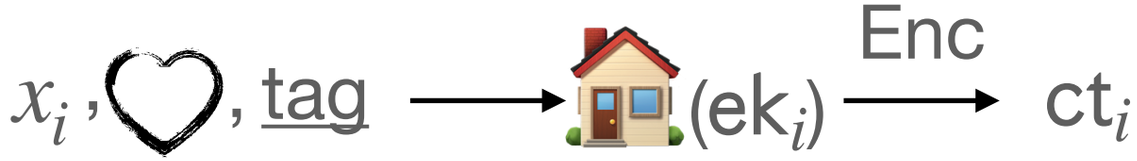
“Compress” in $(\mathbf{F}, \mathbf{F}^*)$

“Duplicate”$(\mathbf{H_i}, \mathbf{H_i^*})$

$$\vec{k}^*_{i,att} = (\cdots \mid a_{att} \mid a'_{att} y_i / z_{att} \mid 0..0)_{\mathbf{F}^*}$$

$$\vec{k}^*_{i,0} = (\cdots \mid a_0 \mid r'_0 y_i \mid 0..0)_{\mathbf{H}_i^*}$$

$\Theta(1)$ aux coordinates per i,att



“Compress” in $(\mathbf{F}, \mathbf{F}^*)$

“Duplicate”$(\mathbf{H_i}, \mathbf{H_i^*})$

$$\vec{c}_{i,att} = (\cdots \mid \psi_i \mid \tau \Delta x_i z_{att} \mid 0..0)_{\mathbf{F}}$$

$$\vec{c}_{i,0} = (\omega \mid \omega' \mid \psi_i \mid \tau \Delta x_i \mid 0..0)_{\mathbf{H_i}}$$

Hash(tag) as RO

$$|\text{ciphertexts}| = 8nd + 5n$$

Conclusion

MCFE Candidates for Inner Products with LSSS, achieving Adaptive Security in ROM and Linear Total Communication

Conclusion

MCFE Candidates for Inner Products with LSSS, achieving Adaptive Security in ROM and Linear Total Communication

Thanks to
“Compress-and-Duplicate”