Registered Functional Encryption for Attribute-Weighted Sums with Access Control

Tapas Pal[1] **Robert Schädlich**[2]

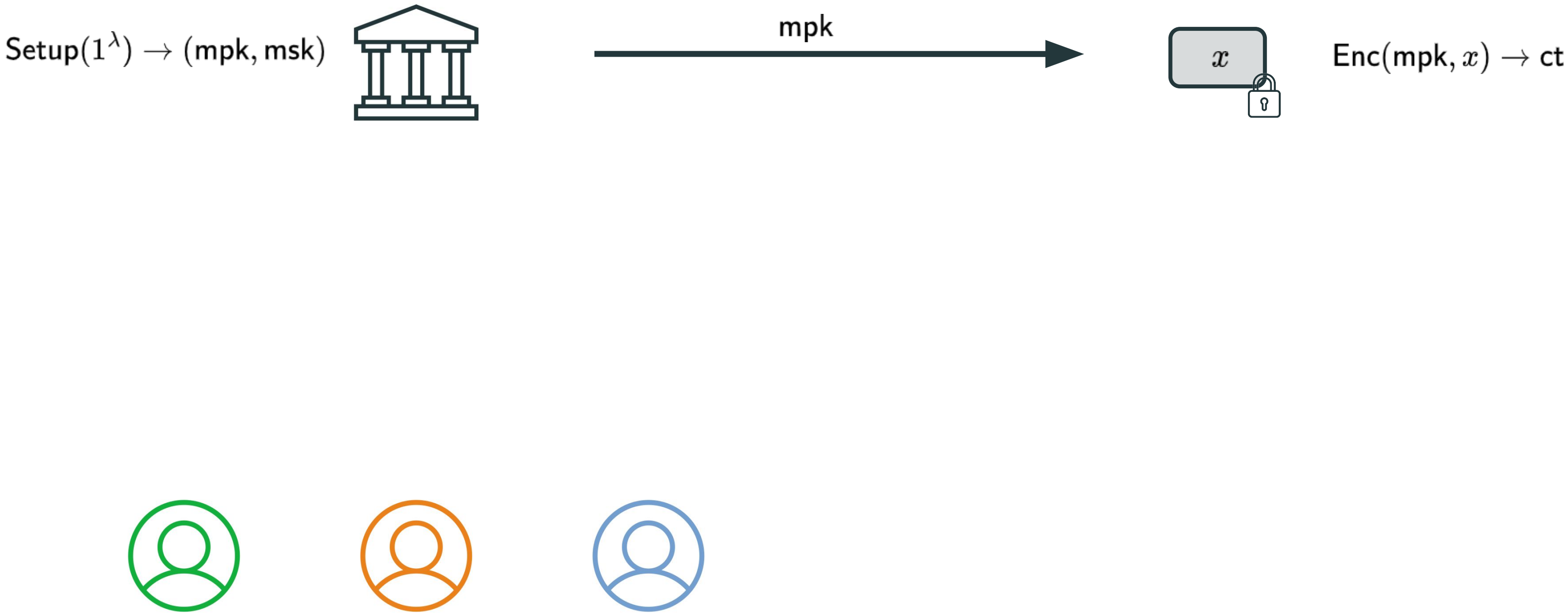
December 5, 2024

[1] Karlsruhe Institute of Technology, KASTEL Security Research Labs
[2] DIENS, École normale supérieure, PSL University, CNRS, Inria

Functional Encryption (FE) [TCC:BSW11]

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

Functional Encryption (FE) [TCC:BSW11]



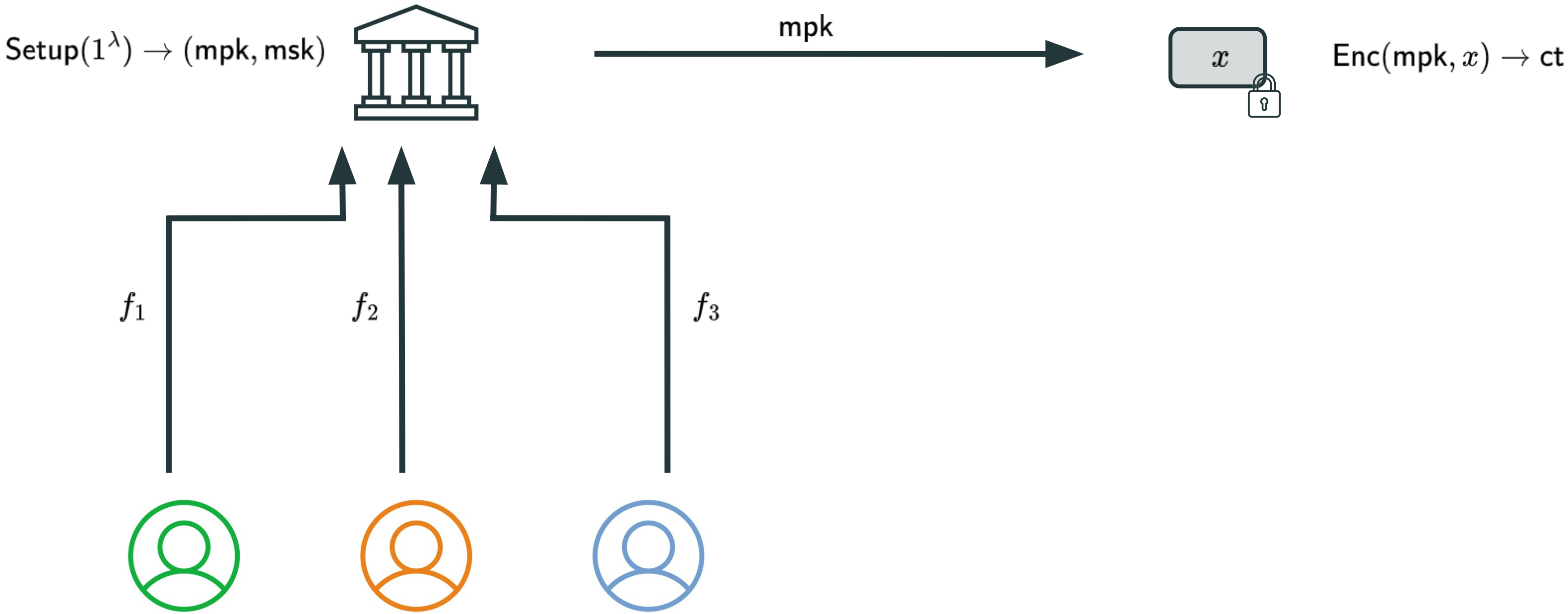
$\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{msk})$

mpk

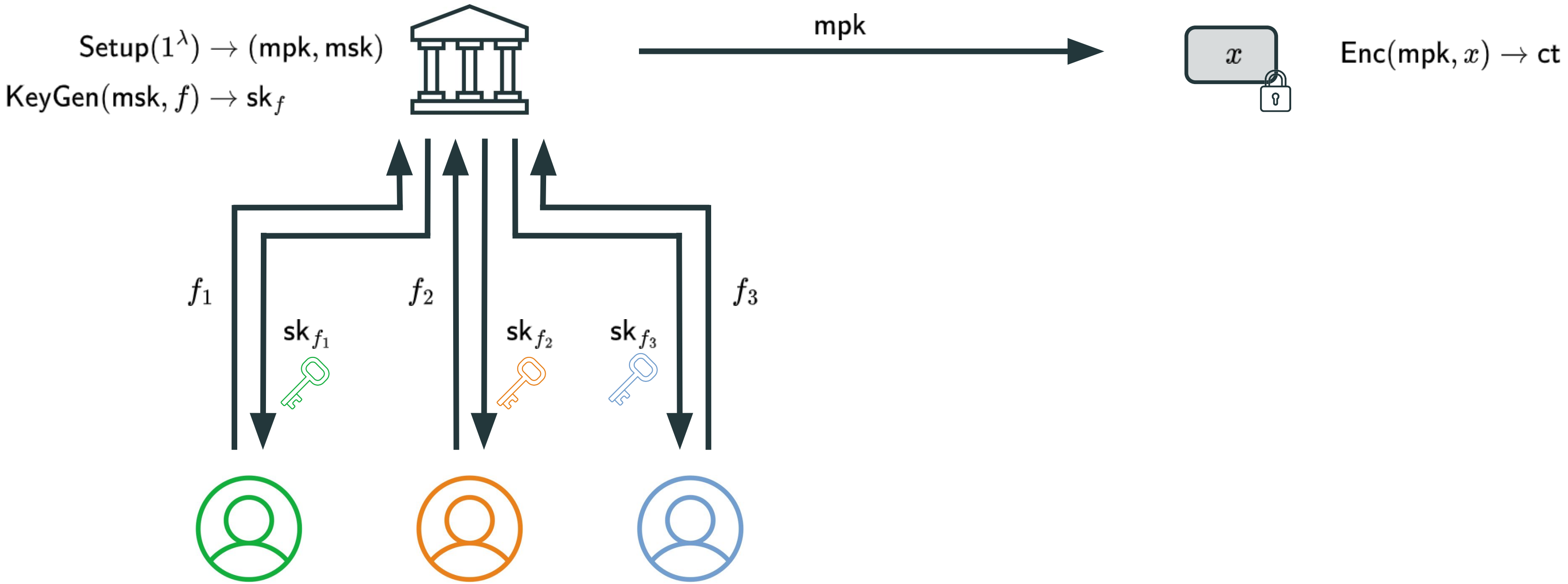
x

$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$

Functional Encryption (FE) [TCC:BSW11]


$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$
mpk
x
$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$
f_1
f_2
f_3

Functional Encryption (FE) [TCC:BSW11]

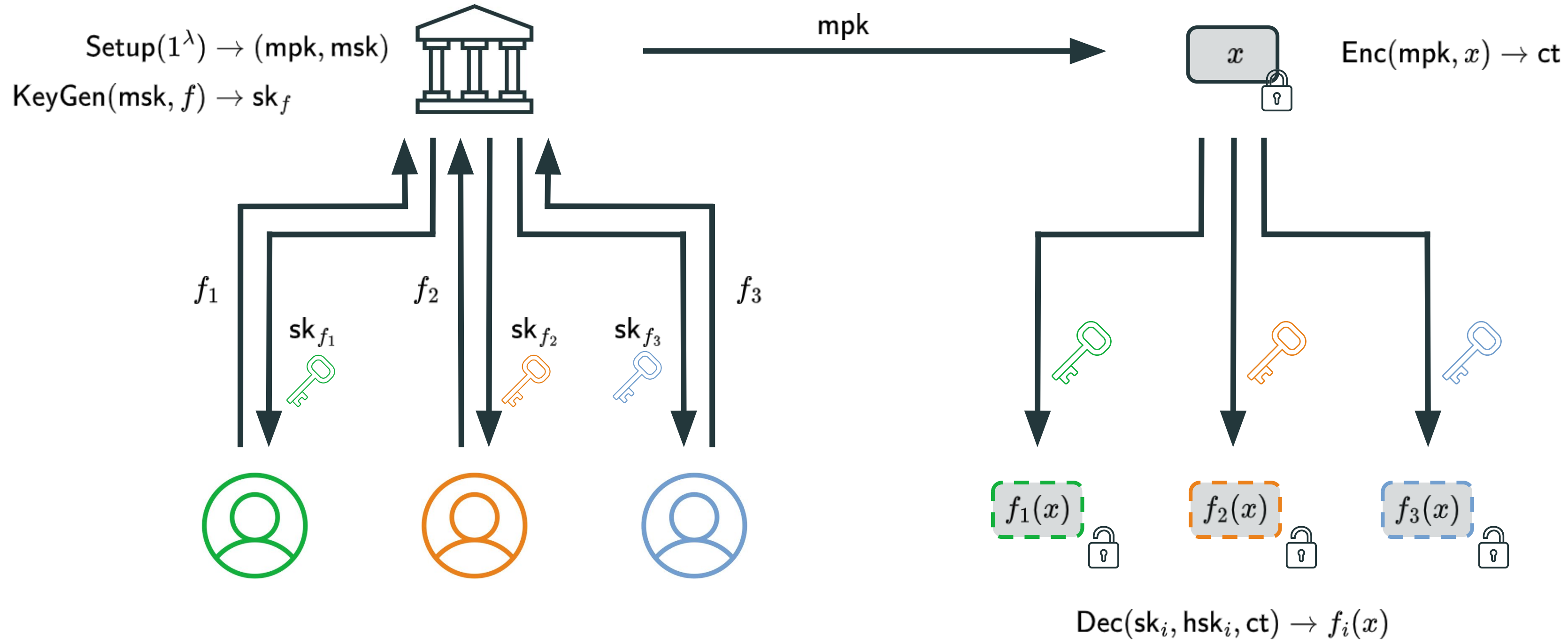


$\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{msk})$

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

$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$

Functional Encryption (FE) [TCC:BSW11]



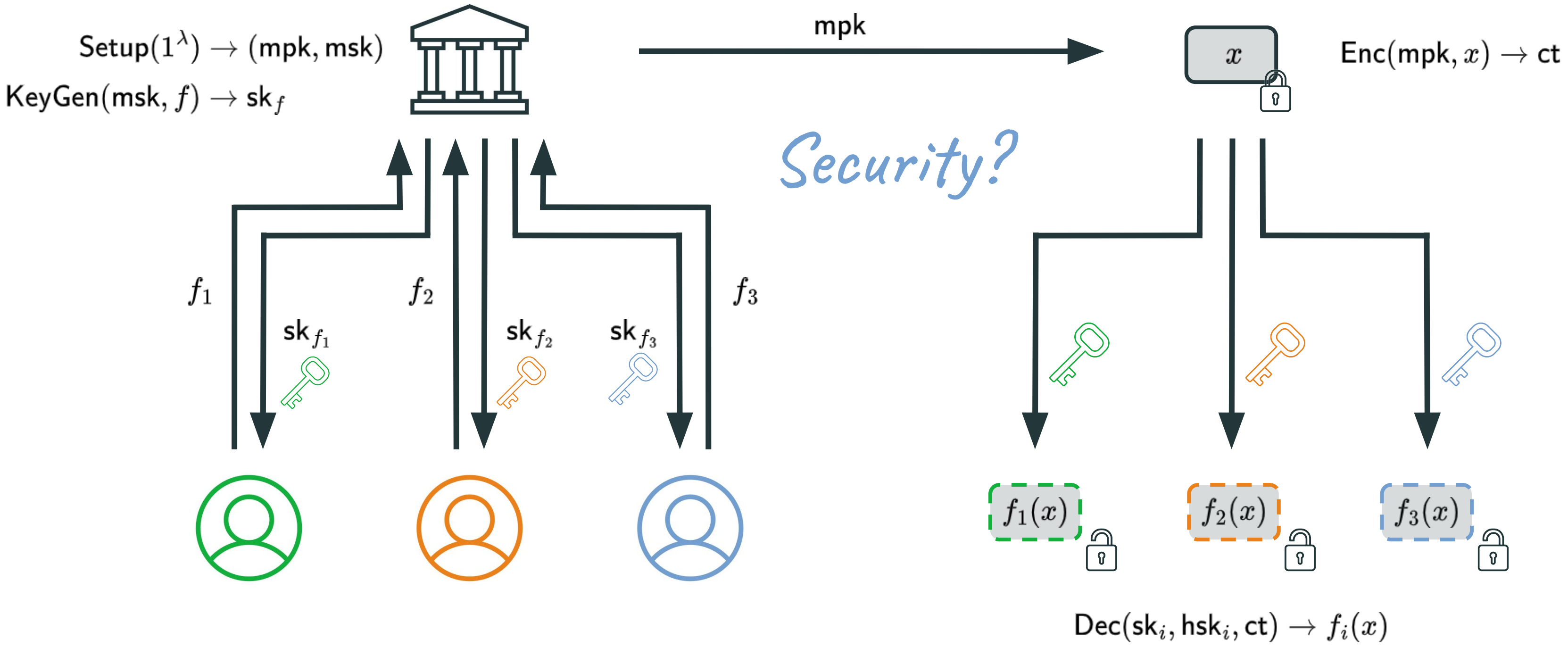
$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

Functional Encryption (FE) [TCC:BSW11]



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

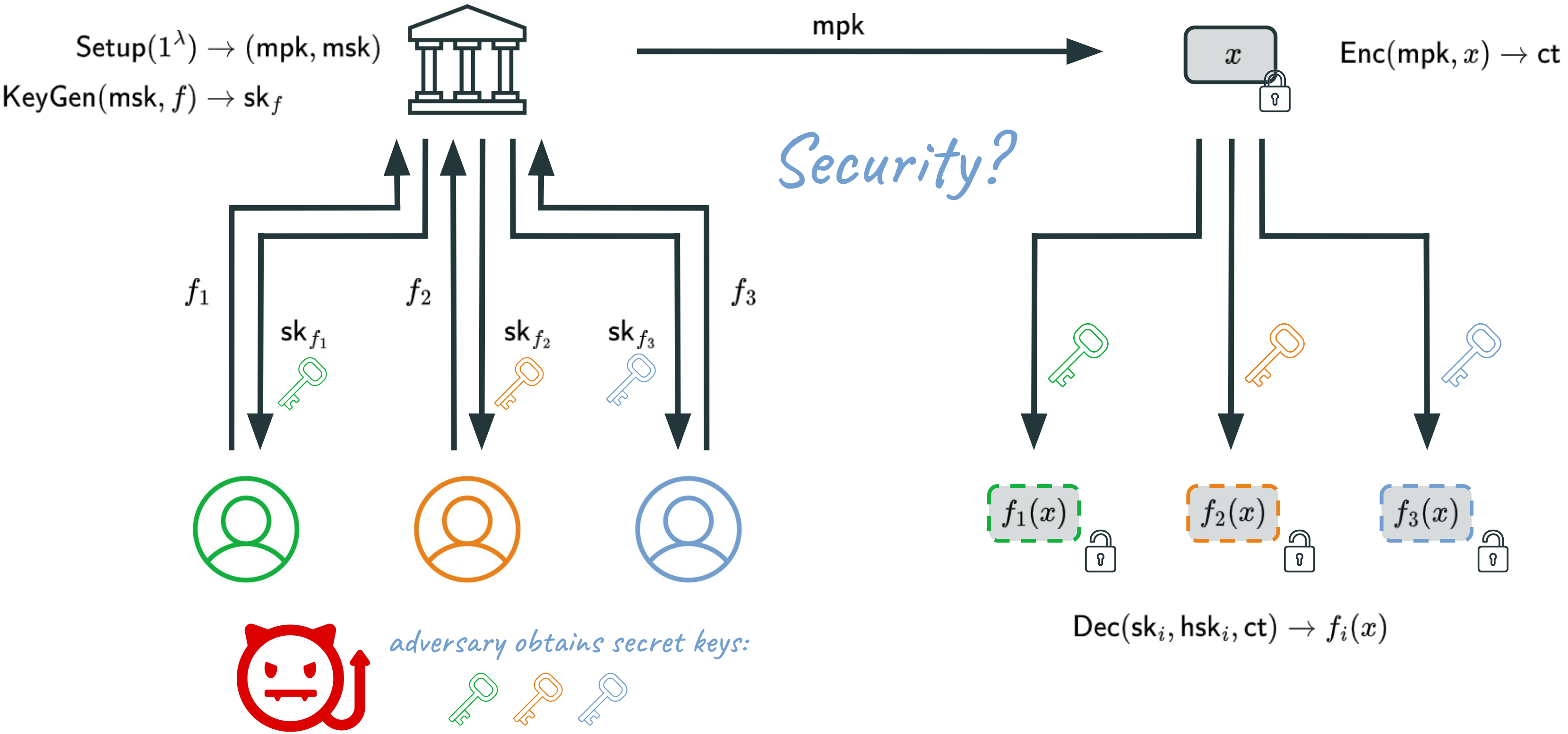
$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

Security?

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

Functional Encryption (FE) [TCC:BSW11]



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

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

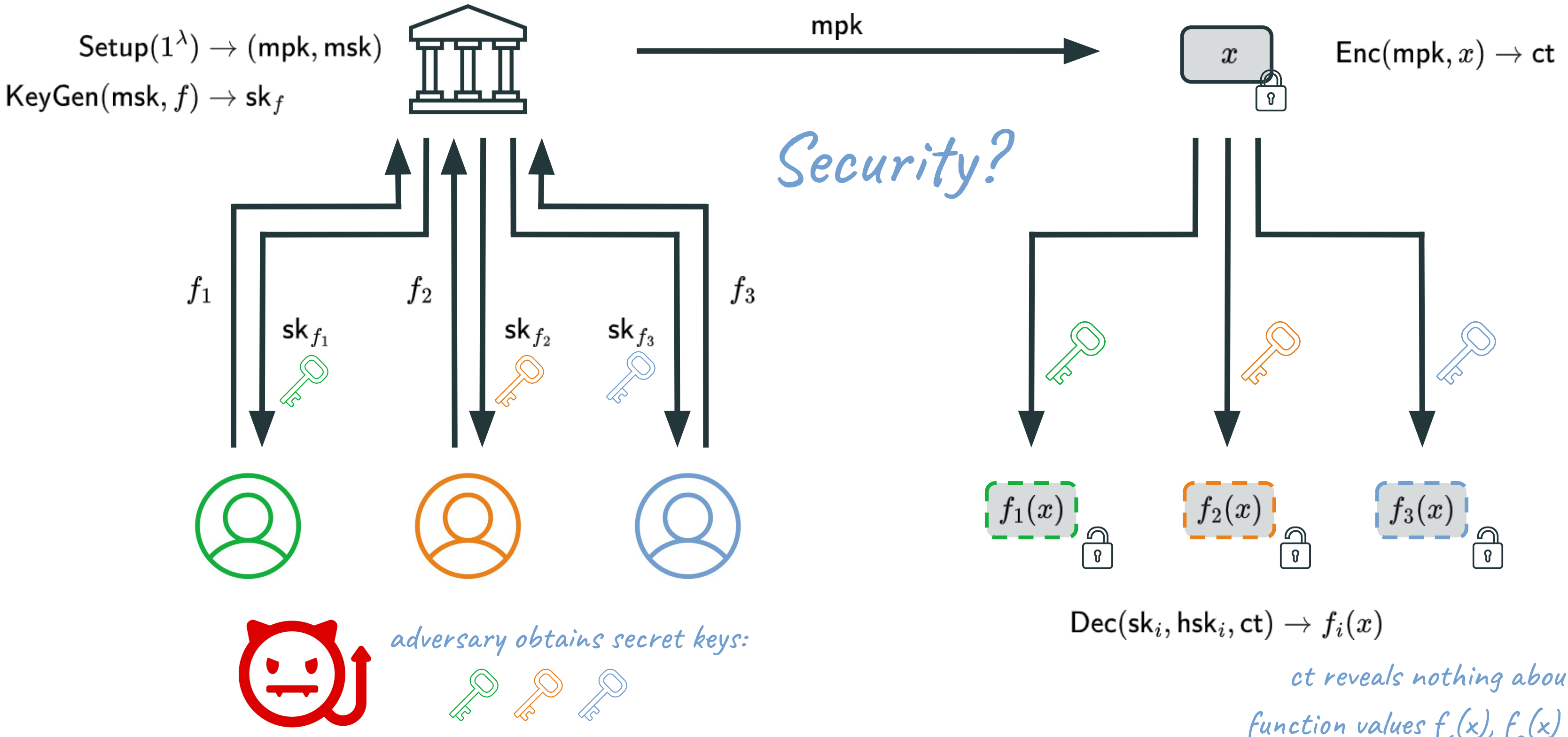
$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

Security?

adversary obtains secret keys:

Functional Encryption (FE) [TCC:BSW11]


Setup$(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$
KeyGen$(\mathsf{msk}, f) \to \mathsf{sk}_f$
mpk
x
Enc$(\mathsf{mpk}, x) \to \mathsf{ct}$
f_1
sk_{f_1}
f_2
sk_{f_2}
sk_{f_3}
f_3
$f_1(x)$
$f_2(x)$
$f_3(x)$
Dec$(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

Security?

adversary obtains secret keys:

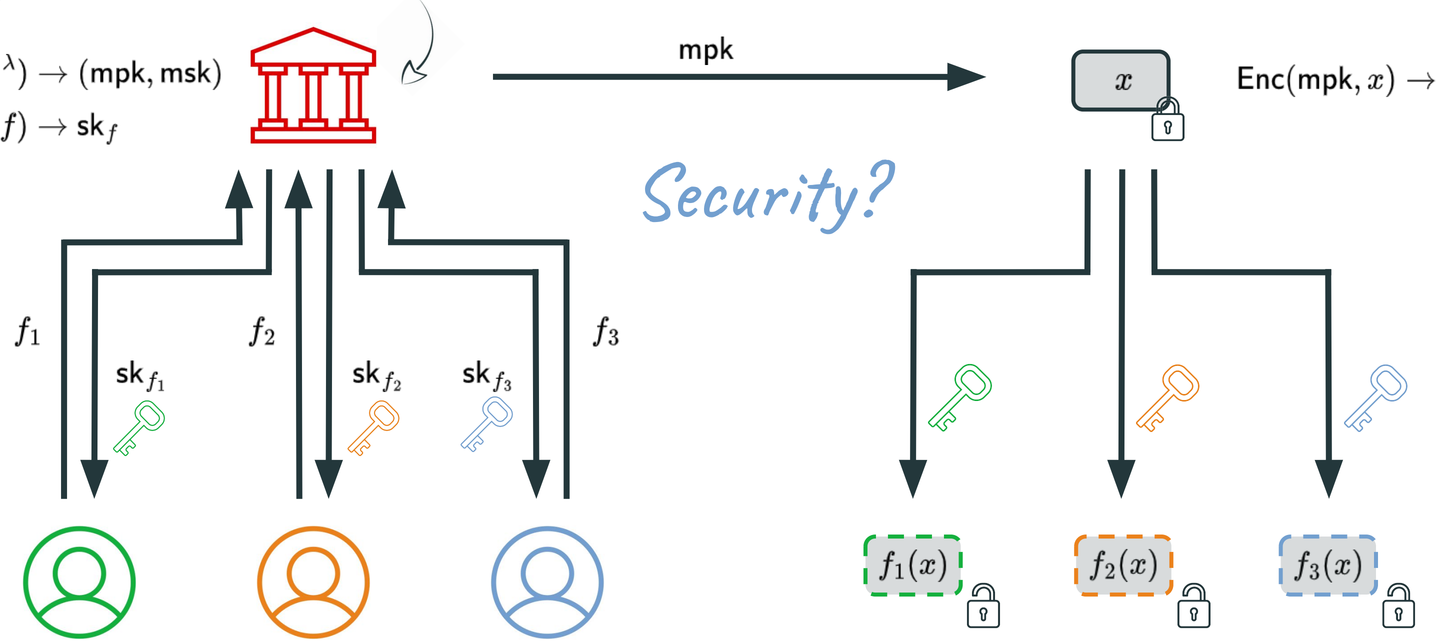
ct reveals nothing about x except function values $f_1(x)$, $f_2(x)$ and $f_3(x)$

Functional Encryption (FE) [TCC:BSW11]

key-escrow problem: msk reveals f(m) for all f :(

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

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

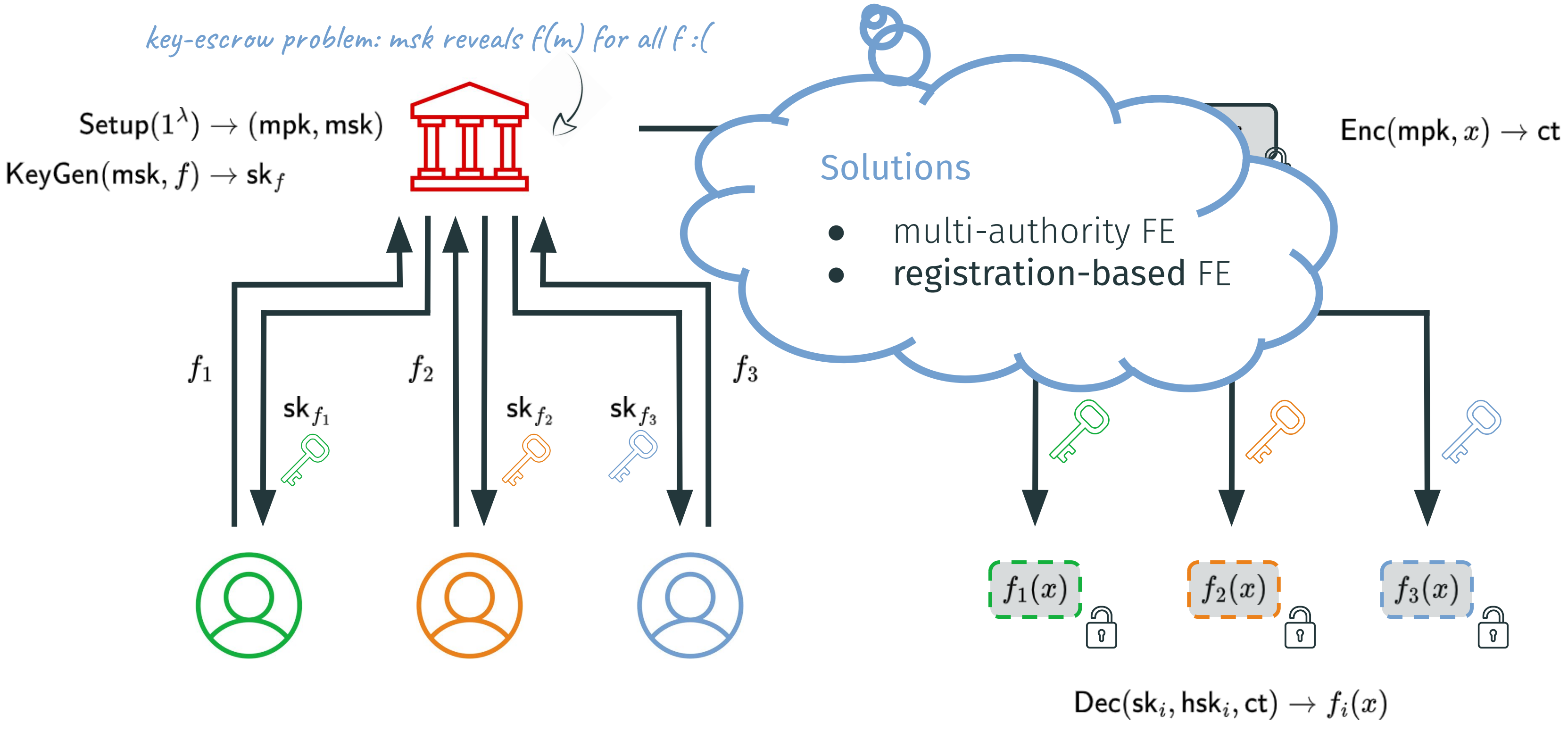


$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

Security?

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

Functional Encryption (FE) [TCC:BSW11]



key-escrow problem: msk reveals f(m) for all f :(

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

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

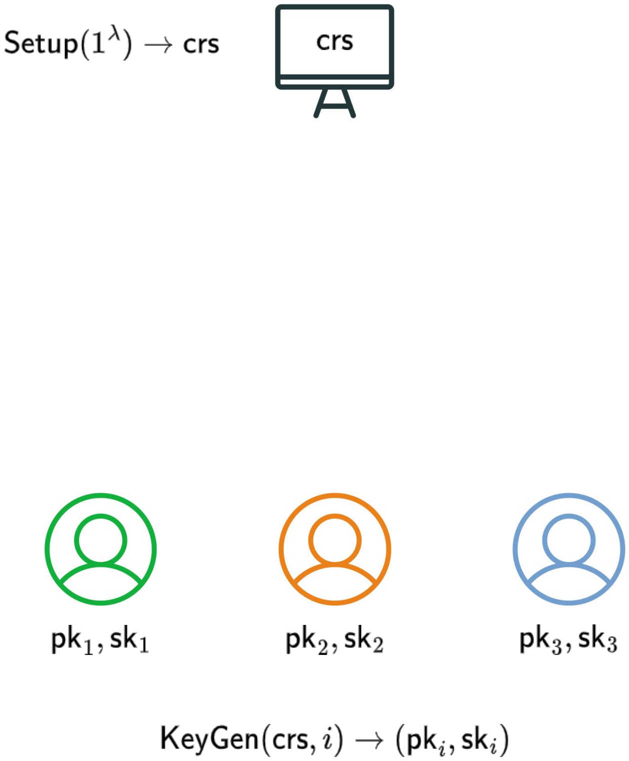
$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$

Solutions

- multi-authority FE
- **registration-based** FE

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \rightarrow f_i(x)$

Registered Functional Encryption (RFE) [AC:FFM+23]



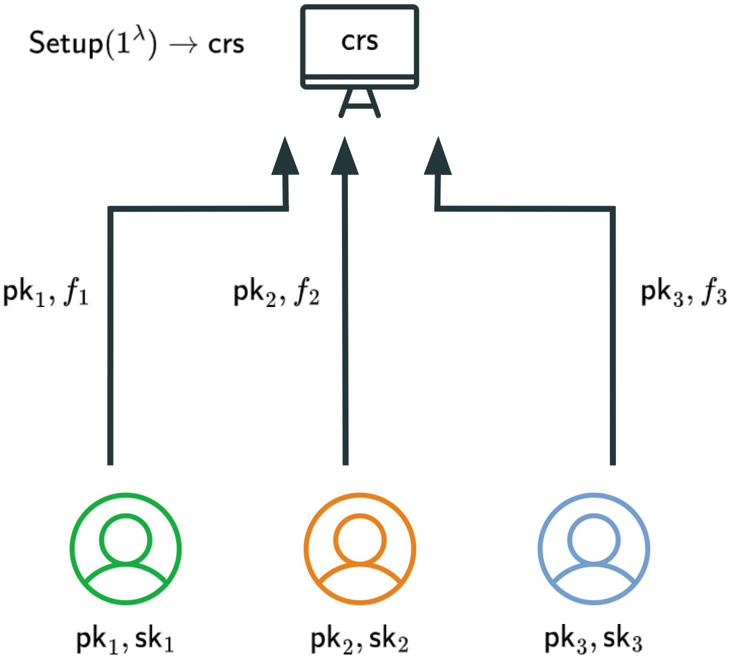
$\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$

$\mathsf{pk}_1, \mathsf{sk}_1$ $\mathsf{pk}_2, \mathsf{sk}_2$ $\mathsf{pk}_3, \mathsf{sk}_3$

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

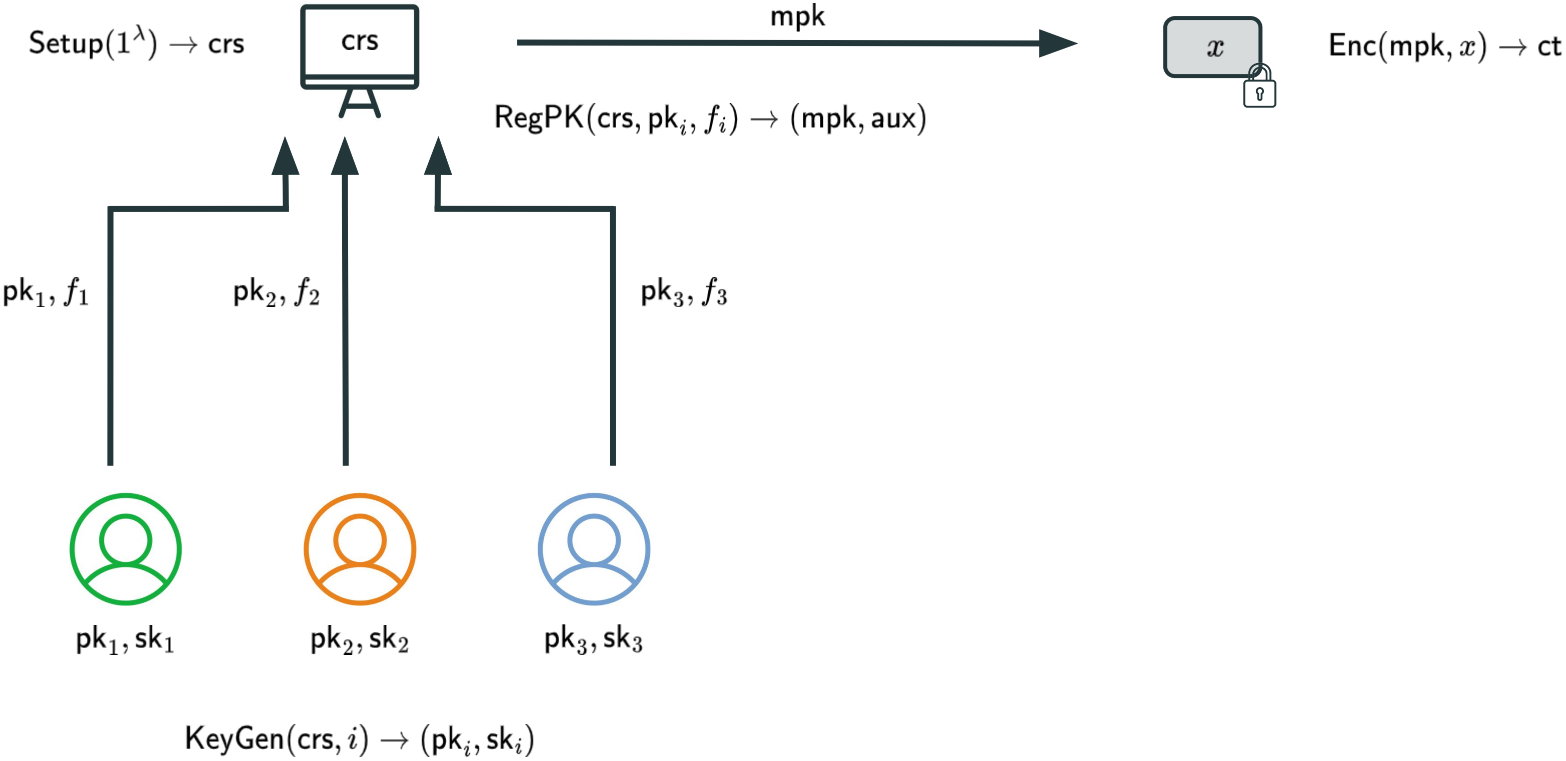
Registered Functional Encryption (RFE) [AC:FFM+23]

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


crs
pk_1, f_1
pk_2, f_2
pk_3, f_3
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$

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

Registered Functional Encryption (RFE) [AC:FFM+23]



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

crs

mpk

x

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \to (\mathsf{mpk}, \mathsf{aux})$

pk_1, f_1

pk_2, f_2

pk_3, f_3

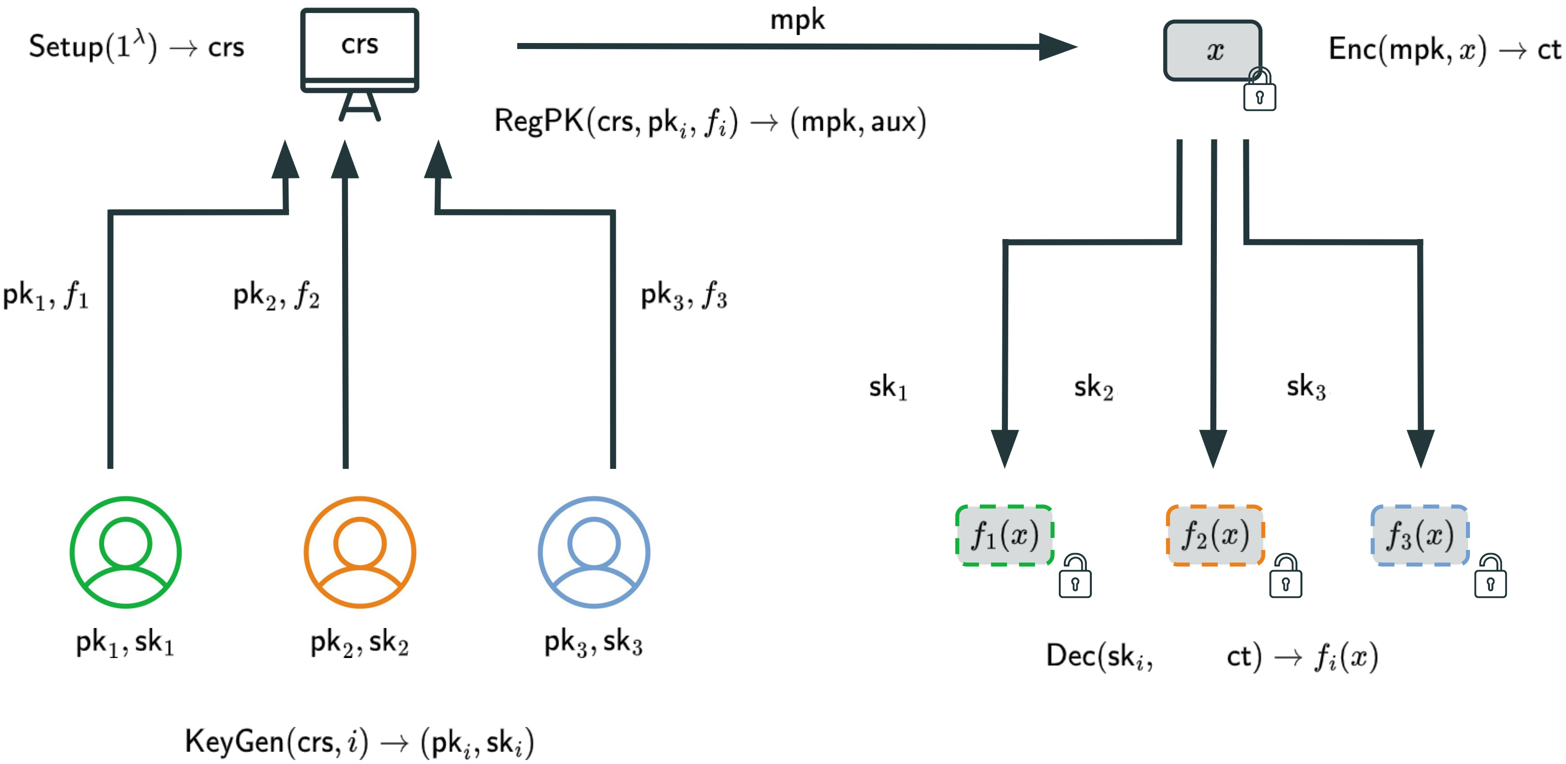
$\mathsf{pk}_1, \mathsf{sk}_1$

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

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

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

Registered Functional Encryption (RFE) [AC:FFM+23]



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

$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \to (\mathsf{mpk}, \mathsf{aux})$

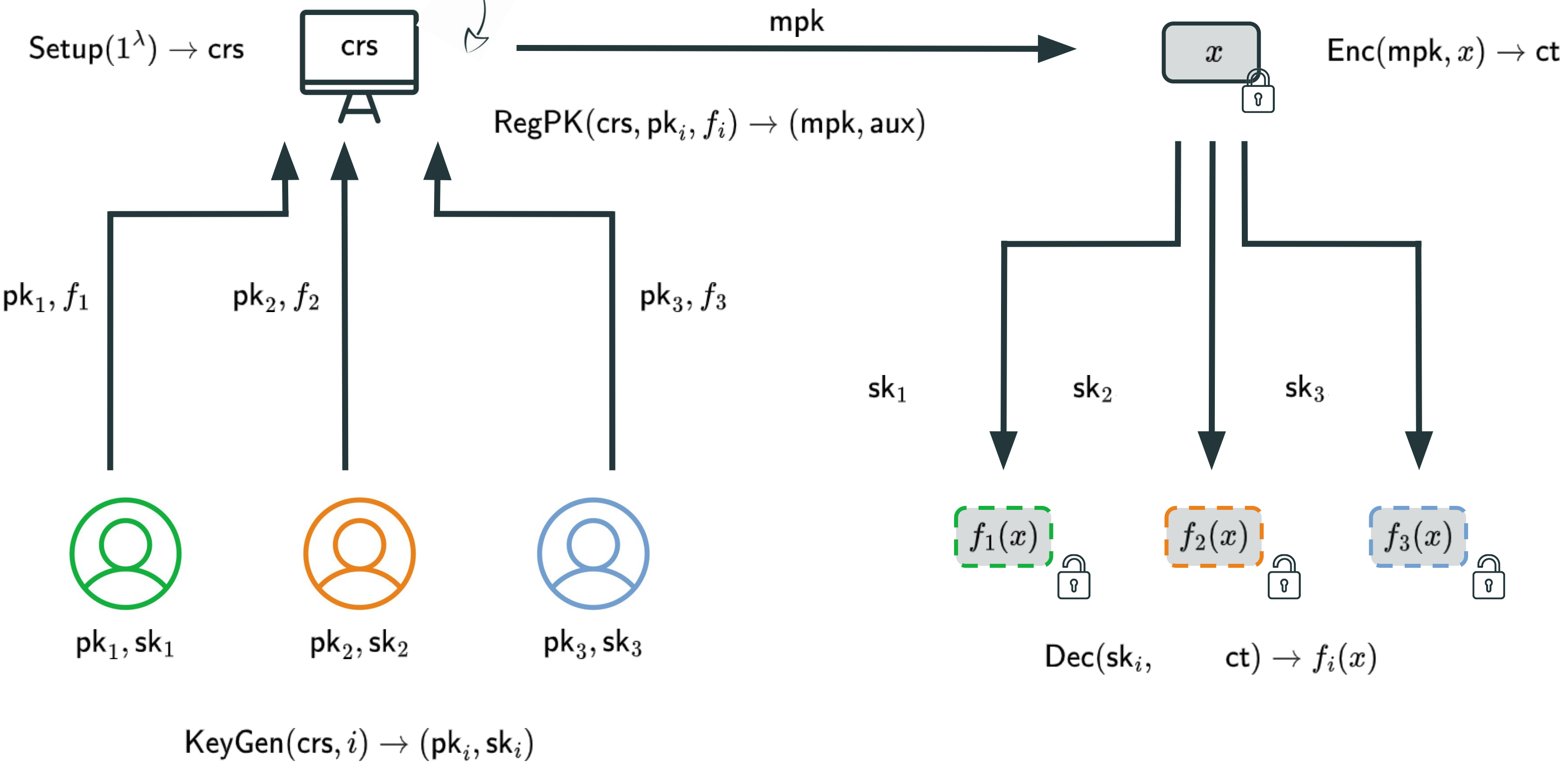
$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

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

$\mathsf{Dec}(\mathsf{sk}_i, \quad \mathsf{ct}) \to f_i(x)$

Registered Functional Encryption (RFE) [AC:FFM+23]

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



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

$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \rightarrow (\mathsf{mpk}, \mathsf{aux})$

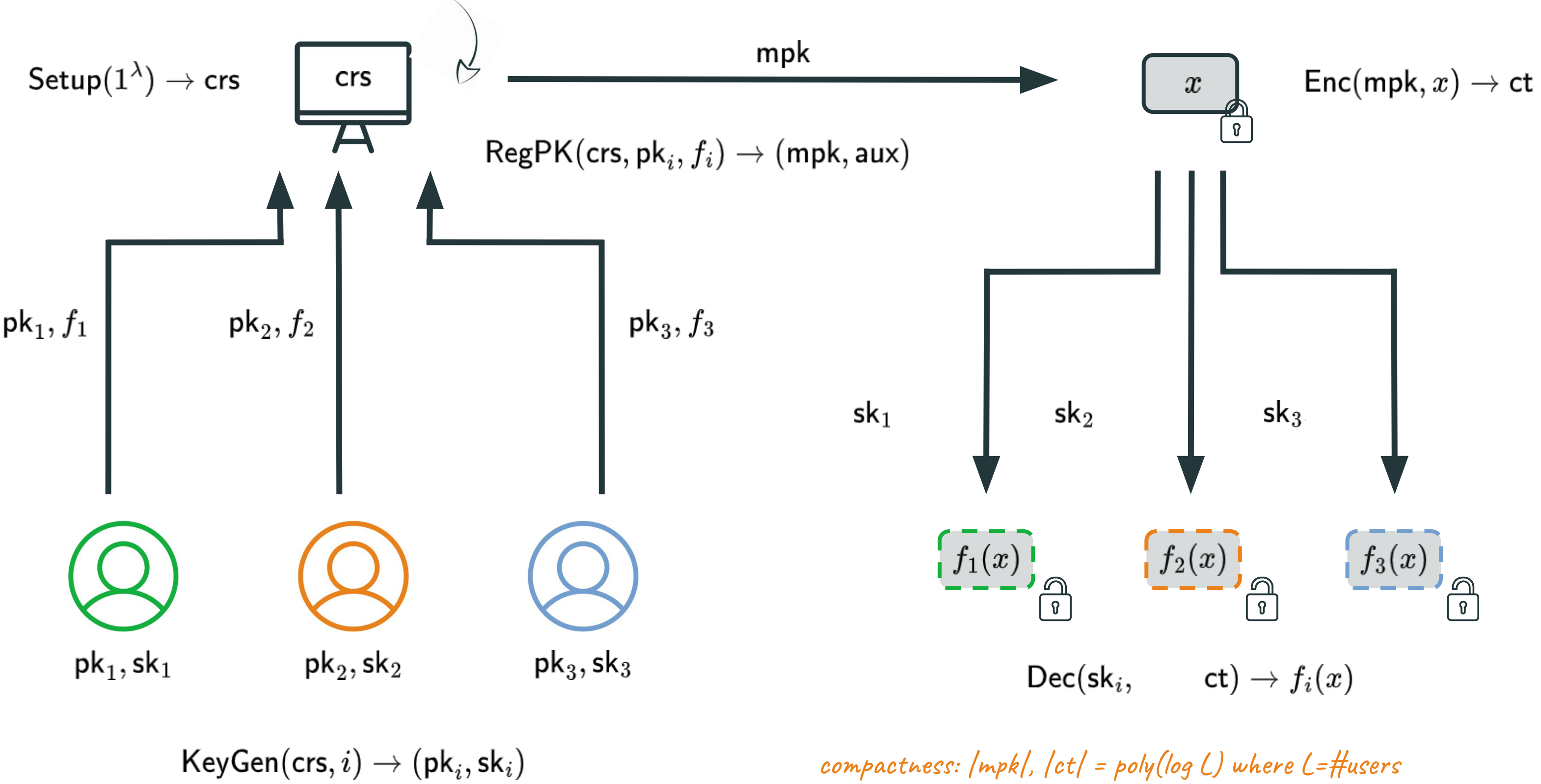
$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$

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

$\mathsf{Dec}(\mathsf{sk}_i, \quad \mathsf{ct}) \rightarrow f_i(x)$

Registered Functional Encryption (RFE) [AC:FFM+23]

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

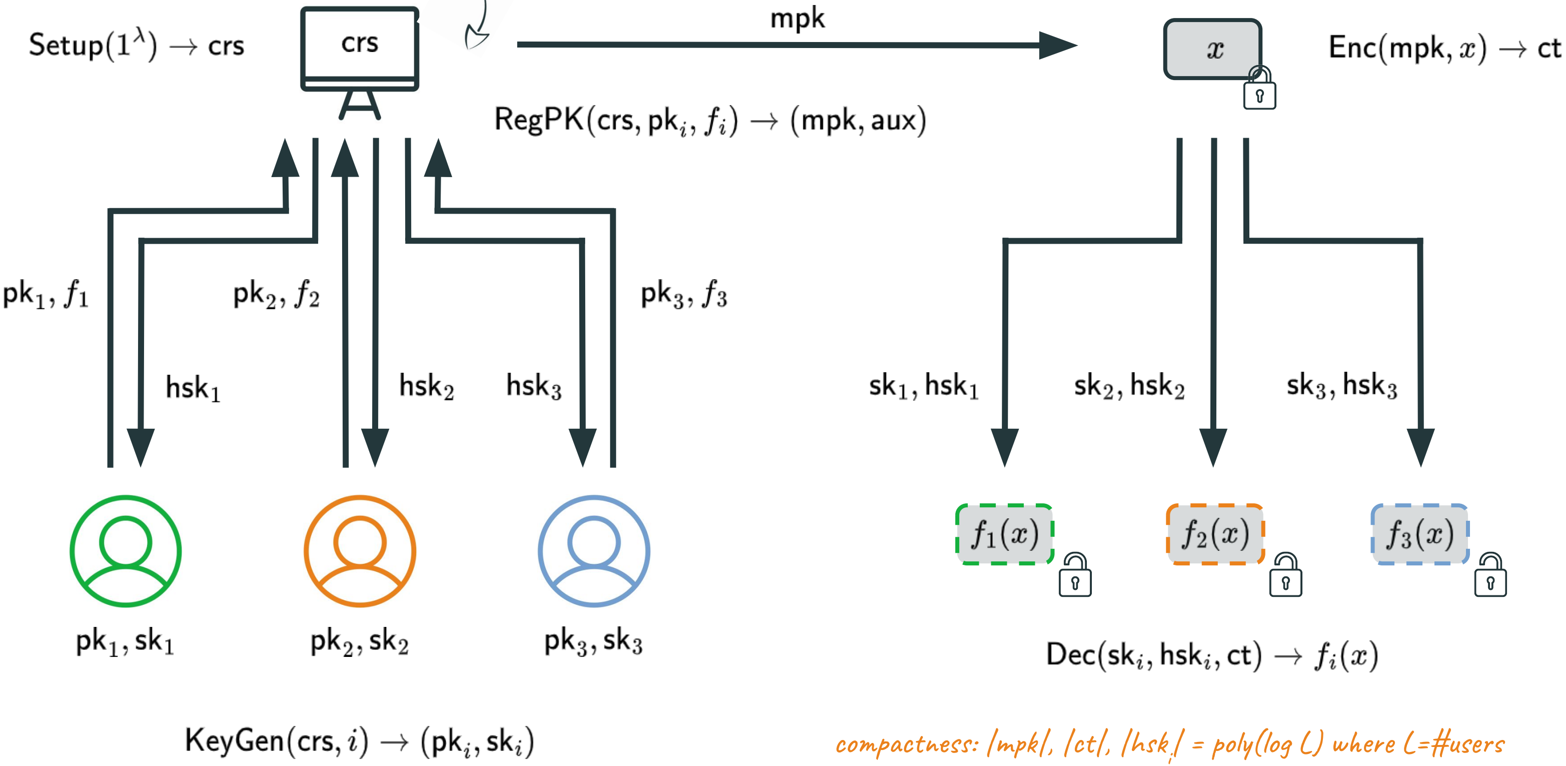


$\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$
crs
mpk
x
$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$
$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \to (\mathsf{mpk}, \mathsf{aux})$
pk_1, f_1
pk_2, f_2
pk_3, f_3
sk_1
sk_2
sk_3
$f_1(x)$
$f_2(x)$
$f_3(x)$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$
$\mathsf{Dec}(\mathsf{sk}_i, \quad \mathsf{ct}) \to f_i(x)$
$\mathsf{KeyGen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$

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

Registered Functional Encryption (RFE) [AC:FFM+23]

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

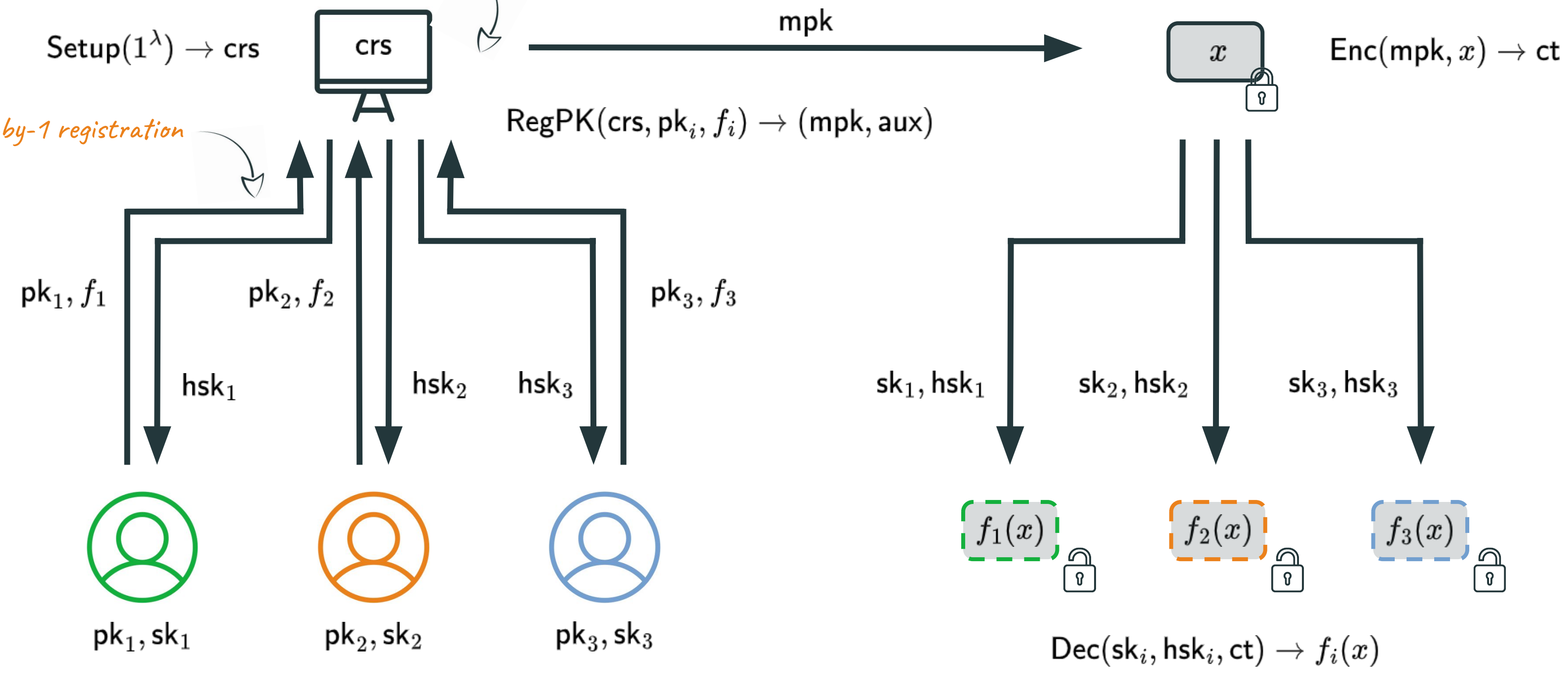


$\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{crs}$
crs
mpk
x
$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$
$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \rightarrow (\mathsf{mpk}, \mathsf{aux})$
pk_1, f_1
pk_2, f_2
pk_3, f_3
hsk_1
hsk_2
hsk_3
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$
$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \rightarrow f_i(x)$
$\mathsf{KeyGen}(\mathsf{crs}, i) \rightarrow (\mathsf{pk}_i, \mathsf{sk}_i)$

compactness: $|\mathsf{mpk}|, |\mathsf{ct}|, |\mathsf{hsk}_i| = \mathrm{poly}(\log L)$ where L=#users

Registered Functional Encryption (RFE) [AC:FFM+23]

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



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

crs

mpk

x

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

1-by-1 registration

$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \to (\mathsf{mpk}, \mathsf{aux})$

pk_1, f_1

pk_2, f_2

pk_3, f_3

hsk_1

hsk_2

hsk_3

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

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

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

$f_1(x)$

$f_2(x)$

$f_3(x)$

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

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

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

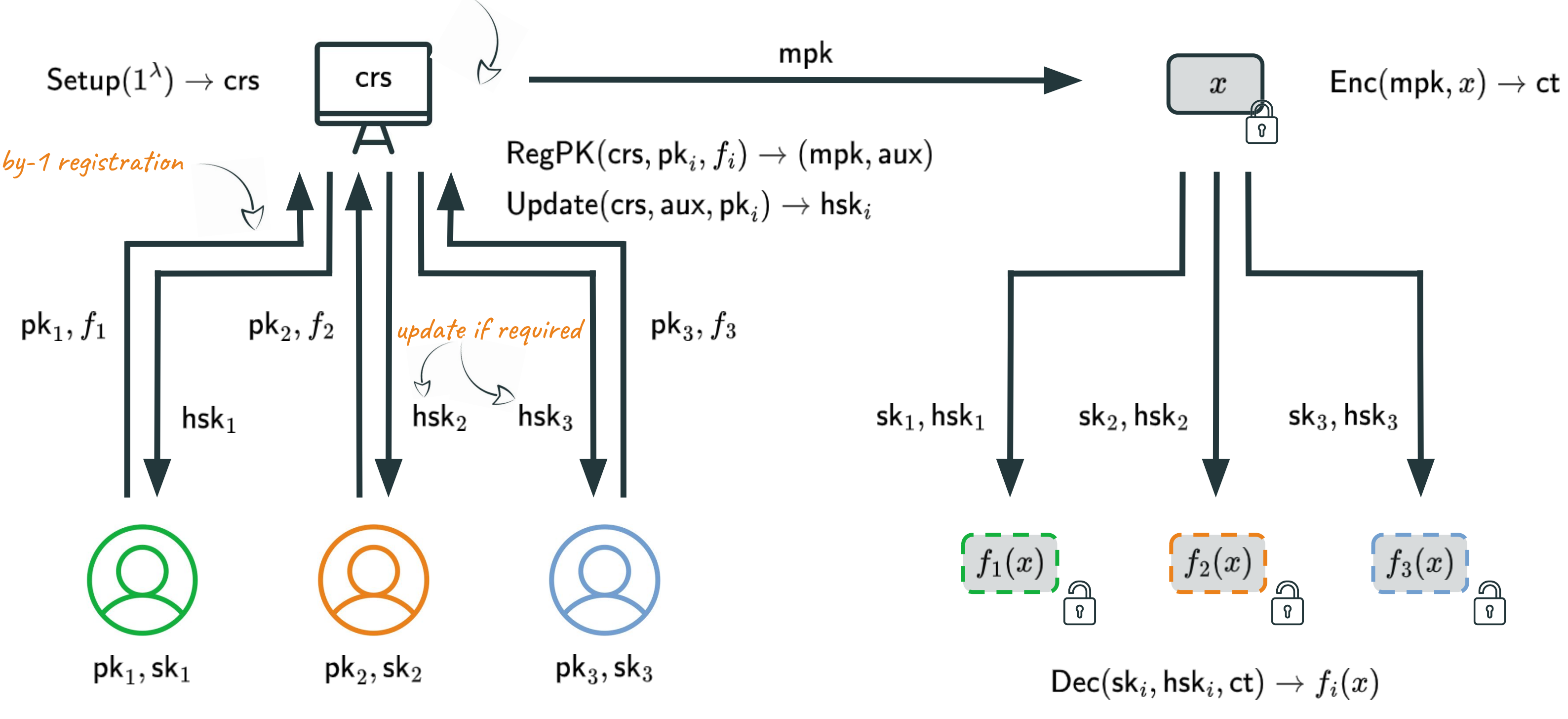
$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

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

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

Registered Functional Encryption (RFE) [AC:FFM+23]

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



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

$\mathsf{RegPK}(\mathsf{crs}, \mathsf{pk}_i, f_i) \to (\mathsf{mpk}, \mathsf{aux})$

$\mathsf{Update}(\mathsf{crs}, \mathsf{aux}, \mathsf{pk}_i) \to \mathsf{hsk}_i$

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

1-by-1 registration

update if required

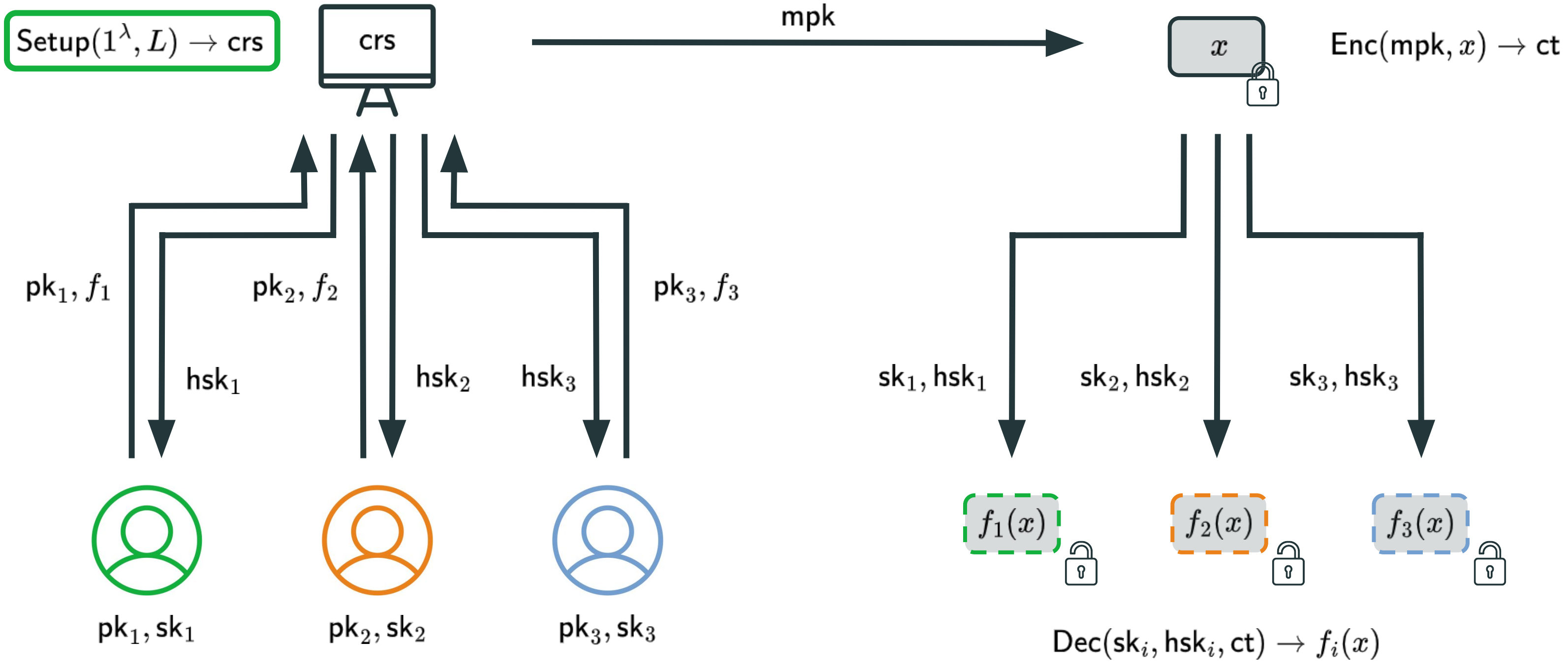
$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

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

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

crs
mpk
x
pk_1, f_1
hsk_1
pk_2, f_2
hsk_2
hsk_3
pk_3, f_3
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$

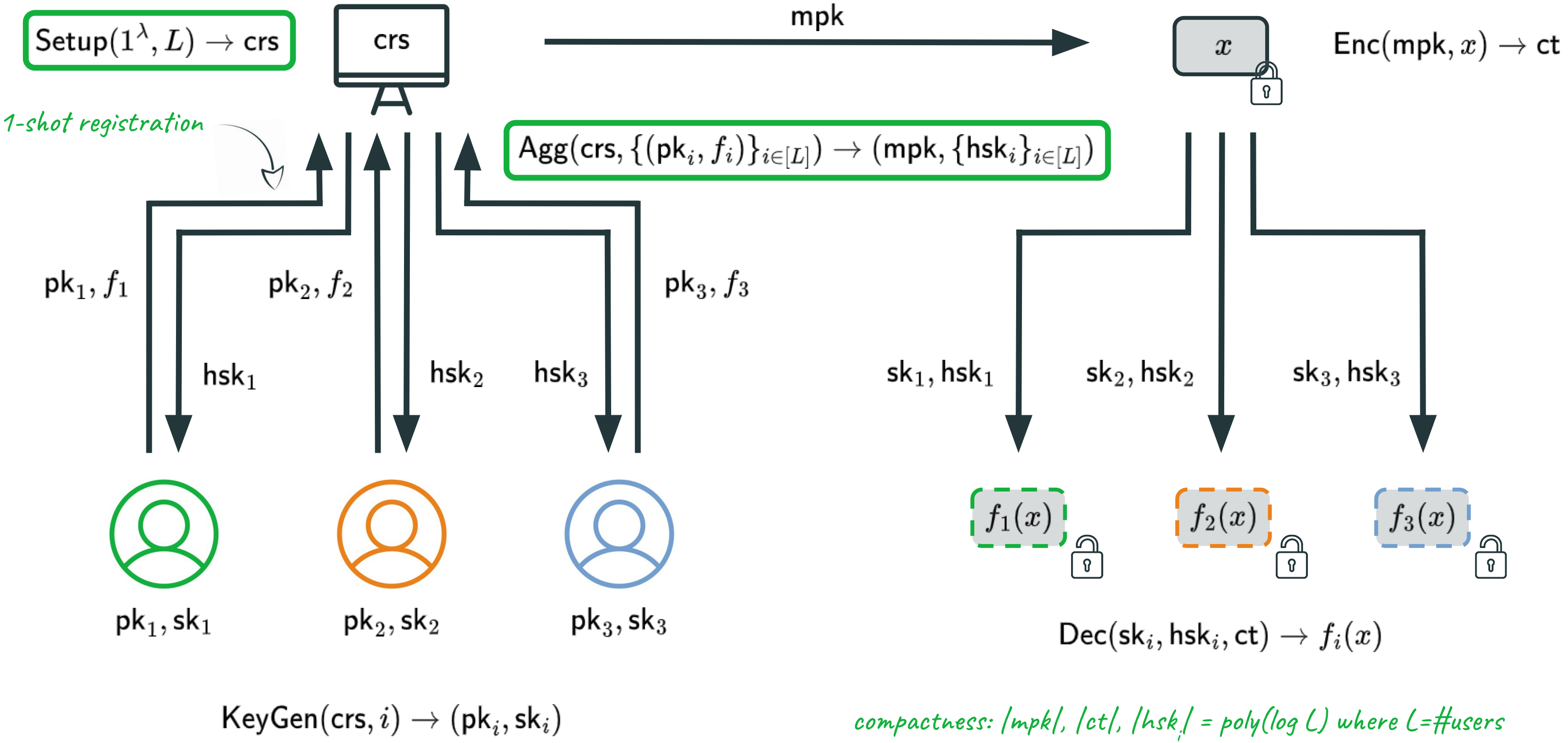
Slotted Registered Functional Encryption (sRFE)


$\text{Setup}(1^\lambda, L) \to \text{crs}$
crs
mpk
x
$\text{Enc}(\text{mpk}, x) \to \text{ct}$
pk_1, f_1
hsk_1
pk_2, f_2
hsk_2
hsk_3
pk_3, f_3
$\text{sk}_1, \text{hsk}_1$
$\text{sk}_2, \text{hsk}_2$
$\text{sk}_3, \text{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
pk_1, sk_1
pk_2, sk_2
pk_3, sk_3
$\text{Dec}(\text{sk}_i, \text{hsk}_i, \text{ct}) \to f_i(x)$

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

compactness: $|\text{mpk}|$, $|\text{ct}|$, $|\text{hsk}_i|$ = poly(log L) where L=#users

Slotted Registered Functional Encryption (sRFE)



$\mathsf{Setup}(1^\lambda, L) \rightarrow \mathsf{crs}$

crs
mpk
x
1-shot registration
pk_1, f_1
hsk_1
pk_2, f_2
hsk_2
hsk_3
pk_3, f_3
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$

$\mathsf{Enc}(\mathsf{mpk}, x) \rightarrow \mathsf{ct}$

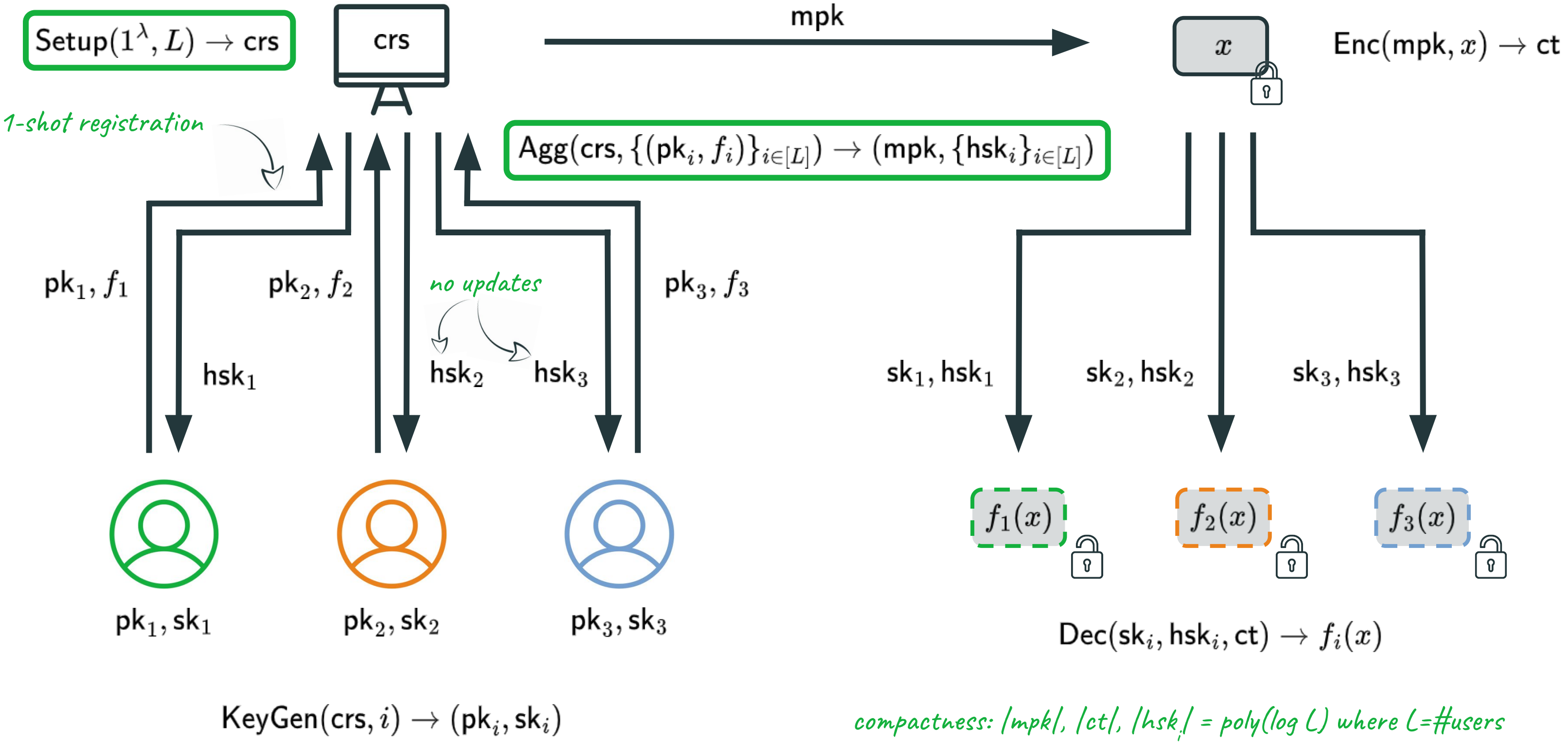
$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[L]}) \rightarrow (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \rightarrow f_i(x)$

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

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

Slotted Registered Functional Encryption (sRFE)



Setup$(1^\lambda, L) \to$ crs

crs

mpk

x

Enc(mpk, x) $\to$ ct

1-shot registration

Agg(crs, $\{(\mathsf{pk}_i, f_i)\}_{i\in[L]}$) $\to$ (mpk, $\{\mathsf{hsk}_i\}_{i\in[L]}$)

pk_1, f_1

hsk_1

pk_2, f_2

no updates

hsk_2

hsk_3

pk_3, f_3

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

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

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

$f_1(x)$

$f_2(x)$

$f_3(x)$

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

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

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

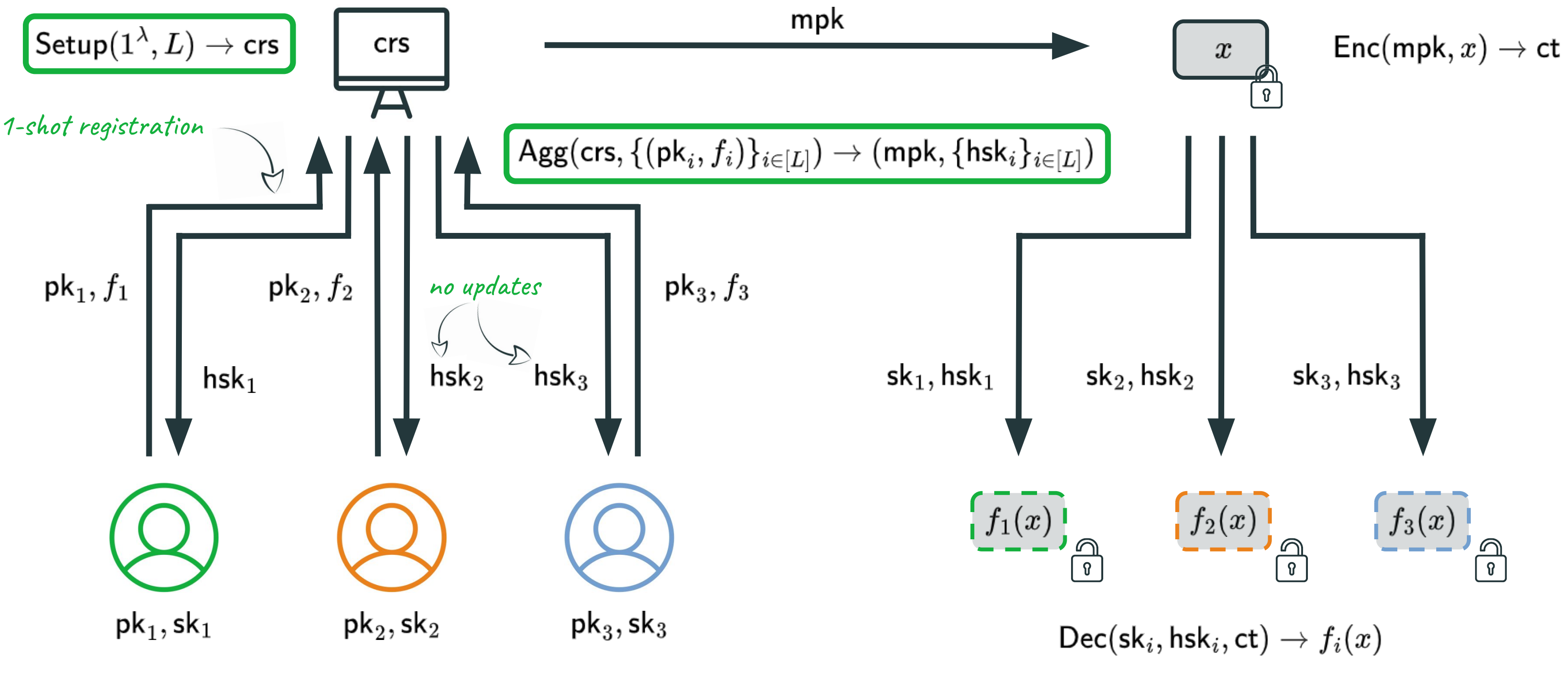
Dec($\mathsf{sk}_i, \mathsf{hsk}_i$, ct) $\to f_i(x)$

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

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

Slotted Registered Functional Encryption (sRFE)

[HLWW23]: sRFE => RFE ("powers-of-two compiler")



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

crs

mpk

x

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

1-shot registration

$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$

pk_1, f_1

pk_2, f_2

no updates

pk_3, f_3

hsk_1

hsk_2

hsk_3

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

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

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

$f_1(x)$

$f_2(x)$

$f_3(x)$

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

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

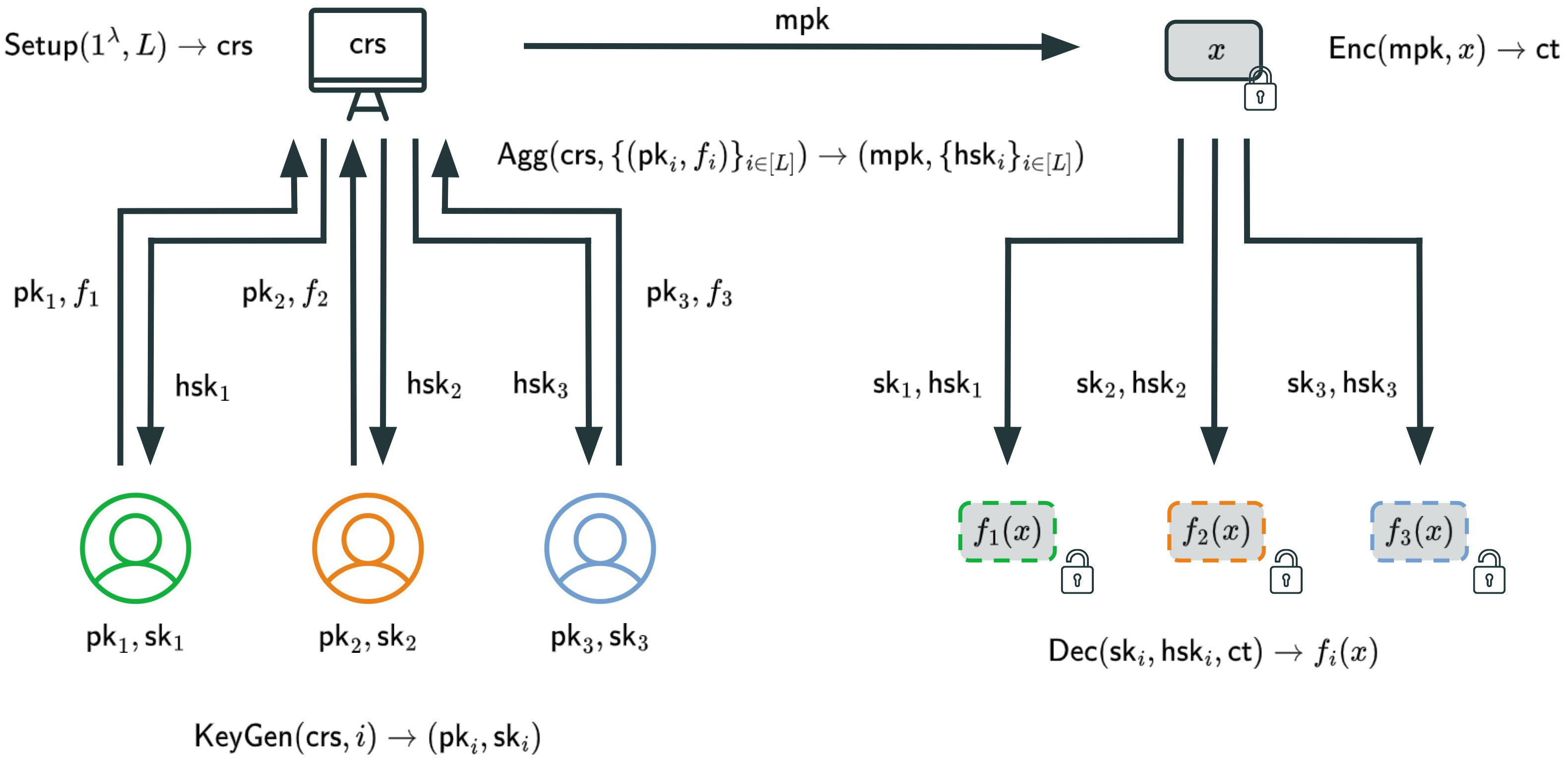
$\mathsf{pk}_3, \mathsf{sk}_3$

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

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

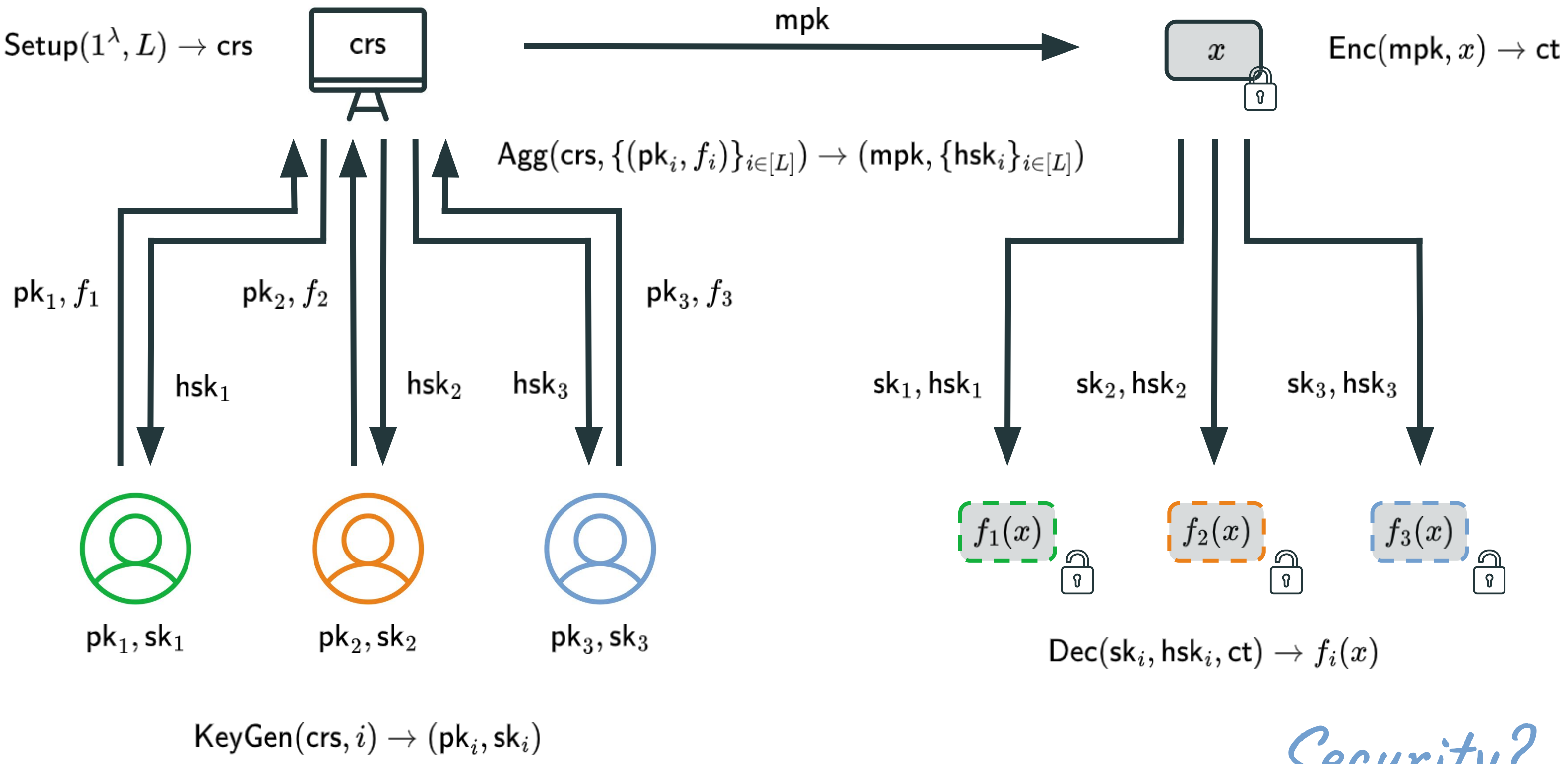
compactness: |mpk|, |ct|, |hsk$_i$| = poly(log L) where L=#users

Slotted Registered Functional Encryption (sRFE)



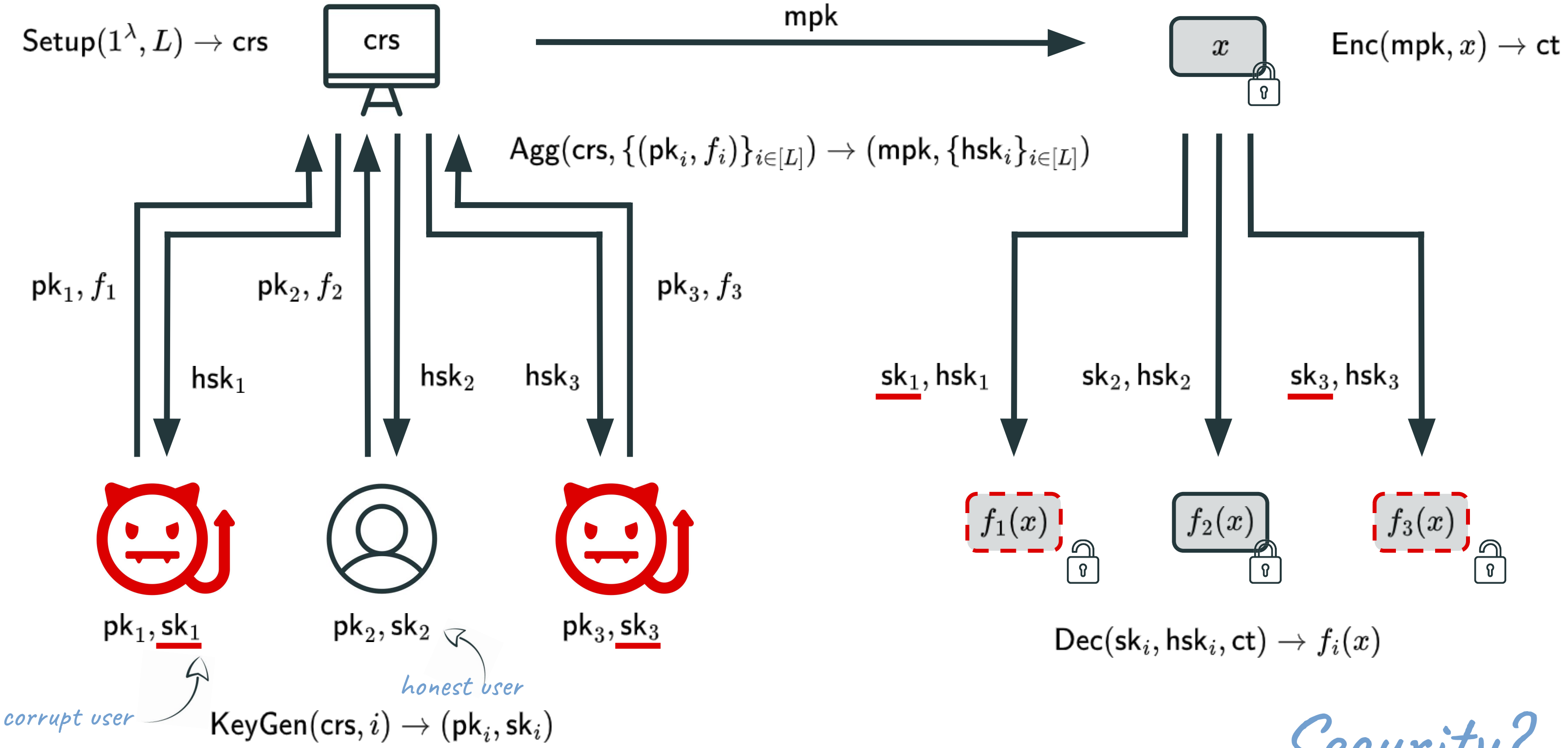
$\mathsf{Setup}(1^\lambda, L) \to \mathsf{crs}$
crs
mpk
x
$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$
$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$
pk_1, f_1
pk_2, f_2
pk_3, f_3
hsk_1
hsk_2
hsk_3
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$
$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$
$\mathsf{KeyGen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$

Slotted Registered Functional Encryption (sRFE)


Setup$(1^\lambda, L) \to$ crs
crs
mpk
x
Enc(mpk, x) $\to$ ct
Agg(crs, $\{(\mathsf{pk}_i, f_i)\}_{i\in[L]}$) $\to$ (mpk, $\{\mathsf{hsk}_i\}_{i\in[L]}$)
pk_1, f_1
pk_2, f_2
pk_3, f_3
hsk_1
hsk_2
hsk_3
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
Dec($\mathsf{sk}_i, \mathsf{hsk}_i$, ct) $\to f_i(x)$
KeyGen(crs, i) $\to$ ($\mathsf{pk}_i, \mathsf{sk}_i$)

Security?

Slotted Registered Functional Encryption (sRFE)



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

crs

mpk

x

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i \in [L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i \in [L]})$

pk_1, f_1

pk_2, f_2

pk_3, f_3

hsk_1

hsk_2

hsk_3

$\underline{\mathsf{sk}_1}, \mathsf{hsk}_1$

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

$\underline{\mathsf{sk}_3}, \mathsf{hsk}_3$

$f_1(x)$

$f_2(x)$

$f_3(x)$

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

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

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

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

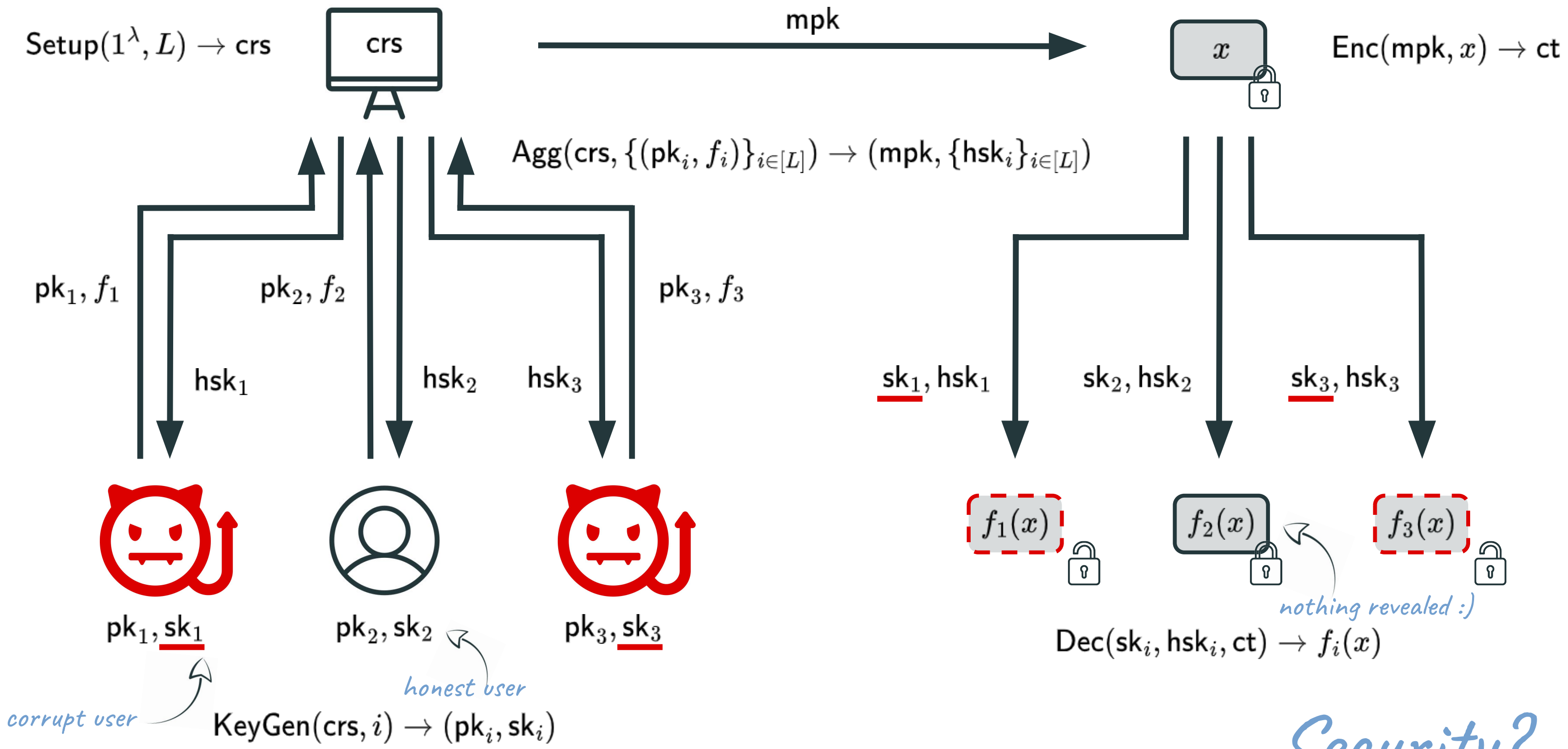
honest user

corrupt user

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

Security?

Slotted Registered Functional Encryption (sRFE)



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

$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

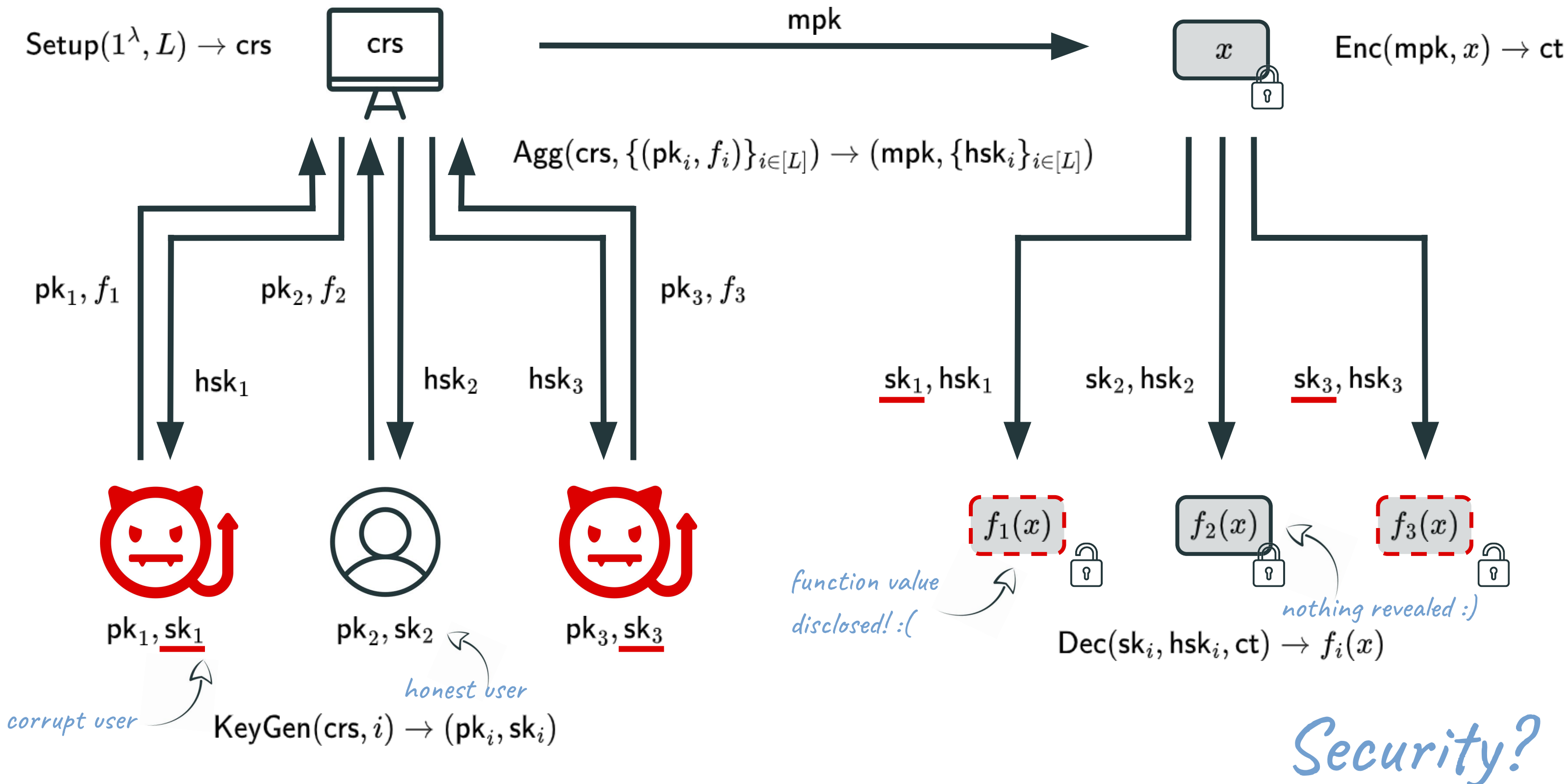
$\mathsf{KeyGen}(\mathsf{crs}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

crs
mpk
x
pk_1, f_1
hsk_1
pk_2, f_2
hsk_2
hsk_3
pk_3, f_3
$\mathsf{pk}_1, \mathsf{sk}_1$
$\mathsf{pk}_2, \mathsf{sk}_2$
$\mathsf{pk}_3, \mathsf{sk}_3$
corrupt user
honest user
$\mathsf{sk}_1, \mathsf{hsk}_1$
$\mathsf{sk}_2, \mathsf{hsk}_2$
$\mathsf{sk}_3, \mathsf{hsk}_3$
$f_1(x)$
$f_2(x)$
$f_3(x)$
nothing revealed :)

Security?

Slotted Registered Functional Encryption (sRFE)



Setup$(1^\lambda, L) \to$ crs

Enc(mpk, x) $\to$ ct

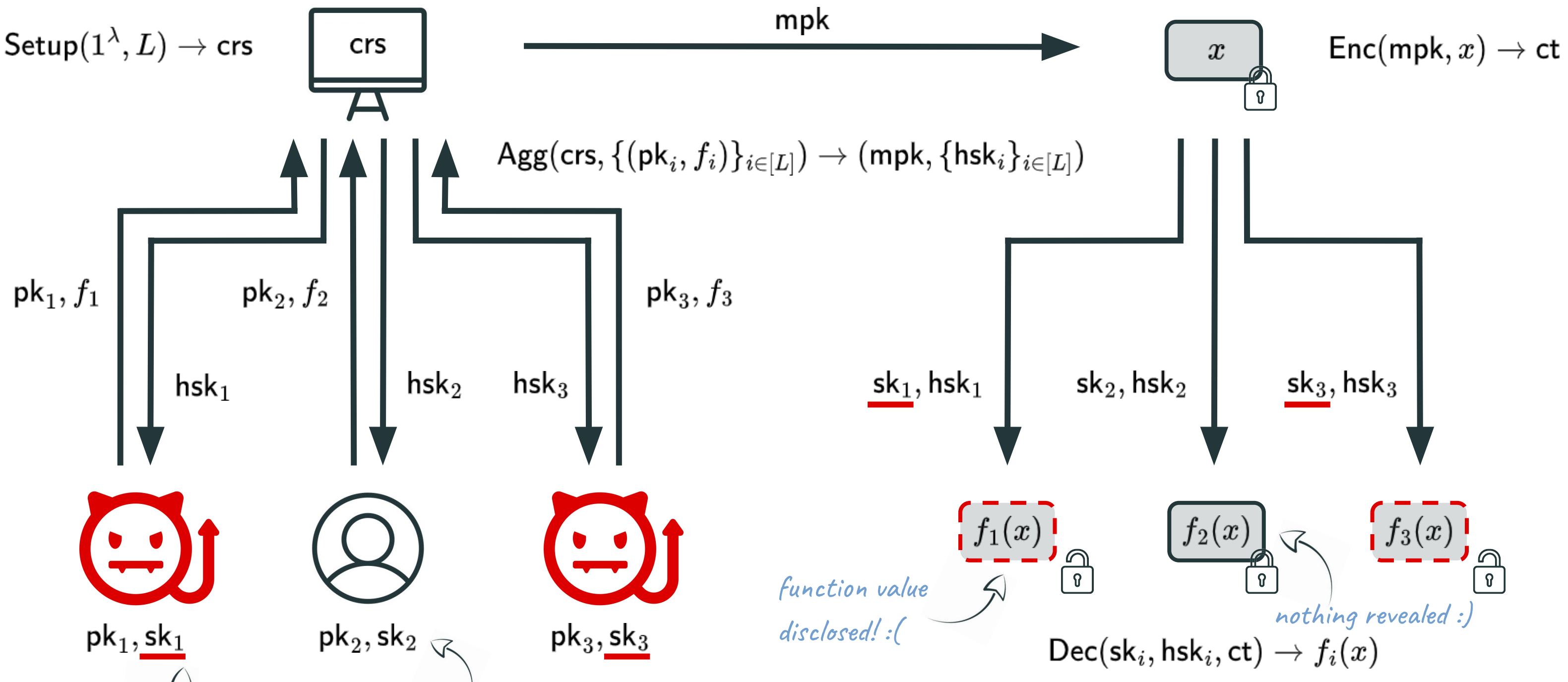
Agg(crs, $\{(\text{pk}_i, f_i)\}_{i\in[L]}$) $\to$ (mpk, $\{\text{hsk}_i\}_{i\in[L]}$)

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

Dec(sk$_i$, hsk$_i$, ct) $\to$ $f_i(x)$

Security?

Slotted Registered Functional Encryption (sRFE)



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

crs

mpk

x

$\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}$

$\mathsf{Agg}(\mathsf{crs}, \{(\mathsf{pk}_i, f_i)\}_{i\in[L]}) \to (\mathsf{mpk}, \{\mathsf{hsk}_i\}_{i\in[L]})$

pk_1, f_1 pk_2, f_2 pk_3, f_3

hsk_1 hsk_2 hsk_3

$\mathsf{sk}_1, \mathsf{hsk}_1$ $\mathsf{sk}_2, \mathsf{hsk}_2$ $\mathsf{sk}_3, \mathsf{hsk}_3$

$f_1(x)$ $f_2(x)$ $f_3(x)$

$\mathsf{pk}_1, \mathsf{sk}_1$ $\mathsf{pk}_2, \mathsf{sk}_2$ $\mathsf{pk}_3, \mathsf{sk}_3$

function value disclosed! :(

nothing revealed :)

$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{hsk}_i, \mathsf{ct}) \to f_i(x)$

honest user

corrupt user

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

this talk: no malicious users

Security?

Existing RFE beyond Predicates

Work	Function Class	Assumption	Remarks
[AC:FFM+23, AC:DPY24]	general	iO, SSB	
[AC:DPY24]	AB-IP	GGM	LSSS access policies
[AC:BLM+24]	IP, weak QF	*q*-type	

Existing RFE beyond Predicates

Work	Function Class	Assumption	Remarks
[AC:FFM$^+$23, AC:DPY24]	general	iO, SSB	
[AC:DPY24]	AB-IP	GGM	LSSS access policies
[AC:BLM$^+$24]	IP, weak QF	q-type	
[EC:ZLZ$^+$24]	IP, QF	bilateral MDDH	

Existing RFE beyond Predicates

Work	Function Class	Assumption	Remarks
[AC:FFM$^+$23, AC:DPY24]	general	iO, SSB	
[AC:DPY24]	AB-IP	GGM	LSSS access policies
[AC:BLM$^+$24]	IP, weak QF	q-type	
[EC:ZLZ$^+$24]	IP, QF	bilateral MDDH	
[this work]	AB-AWS	bilateral MDDH	ABP access policies

attribute-based attribute-weighted sums *(see next slide)*

Attribute-Weighted Sums [C:AGW20]

- inner product (IP) [EC:ZLZ+24] $f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{c}^\top$

Attribute-Weighted Sums [C:AGW20]

- inner product (IP) *[EC:ZLZ+24]* $f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{c}^\top$

- 1-input attribute-weighted sum (1AWS) $f(\mathbf{x}, \mathbf{z}) = \mathbf{z} \cdot h(\mathbf{x})^\top$

variable coefficient vectors (computable by ABP)

Attribute-Weighted Sums [C:AGW20]

- inner product (IP) *[EC:ZLZ+24]* $f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{c}^\top$

variable coefficient vectors (computable by ABP)

- 1-input attribute-weighted sum (1AWS) $f(\mathbf{x}, \mathbf{z}) = \mathbf{z} \cdot h(\mathbf{x})^\top$

unbounded-size data sets

- (unbounded-input) attribute-weighted sum (AWS) $f(\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j \in [N]}) = \sum_{j \in [N]} \mathbf{z}_j \cdot h(\mathbf{x}_j)^\top$

Attribute-Weighted Sums [C:AGW20]

- inner product (IP) *[EC:ZLZ+24]* $$f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{c}^\top$$

variable coefficient vectors (computable by ABP)

- 1-input attribute-weighted sum (1AWS) $$f(\mathbf{x}, \mathbf{z}) = \mathbf{z} \cdot h(\mathbf{x})^\top$$

unbounded-size data sets

- (unbounded-input) attribute-weighted sum (AWS) $$f(\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j \in [N]}) = \textstyle\sum_{j \in [N]} \mathbf{z}_j \cdot h(\mathbf{x}_j)^\top$$

- attribute-based attribute-weighted sum (AB-AWS) $$f(\mathbf{y}, \{(\mathbf{x}_j, \mathbf{z}_j)\}_{j \in [N]}) = \begin{cases} \displaystyle\sum_{j \in [N]} \mathbf{z}_j \cdot h(\mathbf{x}_j)^\top & \text{if } g(\mathbf{y}) = 0 \\ \perp & \text{if } g(\mathbf{y}) \neq 0 \end{cases}$$

fine-grained access control

Inner Product Functional Encryption (IPFE) [PKC:ABDP15, C:ALS16]

- setup: sample random matrices $\mathbf{A}$, $\mathbf{W}$ and define $\mathsf{mpk} = ([\mathbf{A}], [\mathbf{A}\mathbf{W}])$, $\mathsf{msk} = \mathbf{W}$

Inner Product Functional Encryption (IPFE) [PKC:ABDP15, C:ALS16]

- setup: sample random matrices $\mathbf{A}$, $\mathbf{W}$ and define $\mathsf{mpk} = ([\mathbf{A}], [\mathbf{AW}])$, $\mathsf{msk} = \mathbf{W}$
- encryption: to encrypt $\mathbf{z}$, sample random vector $\mathbf{s}$ and output $\boxed{\mathsf{ct} = \left(\underset{[c_1] :=}{[\mathbf{sA}]}, \underset{=: [c_2]}{[\mathbf{z} - \mathbf{sAW}]}\right)}$

Inner Product Functional Encryption (IPFE) [PKC:ABDP15, C:ALS16]

- **setup:** sample random matrices $\mathbf{A}, \mathbf{W}$ and define $\mathsf{mpk} = ([\mathbf{A}], [\mathbf{AW}]),\ \mathsf{msk} = \mathbf{W}$
- **encryption:** to encrypt $\mathbf{z}$, sample random vector $\mathbf{s}$ and output $\boxed{\mathsf{ct} = \big(\underbrace{[\mathbf{sA}]}_{[c_1] :=}, \underbrace{[\mathbf{z} - \mathbf{sAW}]}_{=: [c_2]}\big)}$
- **key generation:** to generate a key for $\mathbf{y}$, output $\mathsf{sk}_{\mathbf{y}} = \mathbf{d}^{\top} := \mathbf{W}\mathbf{y}^{\top}$
- **decryption:** output $[\mathbf{c}_1]\mathbf{d}^{\top} + [\mathbf{c}_2]\mathbf{y}^{\top} = [\mathbf{z}\mathbf{y}^{\top}]$

Inner Product Functional Encryption (IPFE) [PKC:ABDP15, C:ALS16]

- **setup:** sample random matrices $\mathbf{A}$, $\mathbf{W}$ and define $\mathsf{mpk} = ([\mathbf{A}], [\mathbf{AW}])$, $\mathsf{msk} = \mathbf{W}$
- **encryption:** to encrypt $\mathbf{z}$, sample random vector $\mathbf{s}$ and output $\boxed{\mathsf{ct} = \big([\mathbf{sA}], [\mathbf{z} - \mathbf{sAW}]\big)}$

  $[c_1] :=$ $=: [c_2]$
- **key generation:** to generate a key for $\mathbf{y}$, output $\mathsf{sk}_{\mathbf{y}} = \mathbf{d}^\top := \mathbf{W}\mathbf{y}^\top$

  or a matrix Y *(in which case the secret key is* $\boxed{\mathsf{sk}_{\mathbf{Y}} = \mathbf{D} := \mathbf{WY}}$*)*
- **decryption:** output $[\mathbf{c}_1]\mathbf{d}^\top + [\mathbf{c}_2]\mathbf{y}^\top = [\mathbf{z}\mathbf{y}^\top]$ *(or* $[\mathbf{c}_1]\mathbf{D} + [\mathbf{c}_2]\mathbf{Y} = [\mathbf{zY}]$*)*

Partial Garbling for 1AWS [ICALP:IW14]

- garbling: given an ABP h and public input $\mathbf{x}$, compute matrix $\mathbf{L_x}$, sample randomness $\mathbf{w}$, and output

$$\boxed{\mathrm{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L}_{\mathbf{x}})}$$

Partial Garbling for 1AWS [ICALP:IW14]

- garbling: given an ABP h and public input $\mathbf{x}$, compute matrix $\mathbf{L}_{\mathbf{x}}$, sample randomness $\mathbf{w}$, and output

$$\boxed{\mathrm{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L}_{\mathbf{x}})}$$

some subvector

Partial Garbling for 1AWS [ICALP:IW14]

- garbling: given an ABP h and public input $\mathbf{x}$, compute matrix $\mathbf{L}_{\mathbf{x}}$, sample randomness $\mathbf{w}$, and output

$$\boxed{\mathsf{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L}_{\mathbf{x}})}$$

some subvector

- reconstruction: given $(h, \mathbf{x})$, find vector $\mathbf{d}_{h,\mathbf{x}}$ such that

$$(\mathbf{p}_1, \mathbf{p}_2) \cdot \mathbf{d}_{h,\mathbf{x}}^{\top} = \mathbf{z} \cdot h(\mathbf{x})^{\top}$$

Partial Garbling for 1AWS [ICALP:IW14]

- garbling: given an ABP h and public input $\mathbf{x}$, compute matrix $\mathbf{L_x}$, sample randomness $\mathbf{w}$, and output

$$\boxed{\mathsf{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L_x})}$$

some subvector

- reconstruction: given $(h, \mathbf{x})$, find vector $\mathbf{d}_{h,\mathbf{x}}$ such that

$$(\mathbf{p}_1, \mathbf{p}_2) \cdot \mathbf{d}_{h,\mathbf{x}}^\top = \mathbf{z} \cdot h(\mathbf{x})^\top$$

- privacy: for random $\mathbf{w}$, the following distributions are indistinguishable

$$\{(\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L_x})\} \approx_s \{(-\underline{\mathbf{w}}, \mathbf{w}\mathbf{L_x} + \mathbf{z}h(\mathbf{x})^\top \cdot \mathbf{e}_1)\}$$

Combining the Two — Classical FE for 1AWS

$\mathsf{FE.ct}$ $\quad \big([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]\big)$

$\mathsf{FE.sk}_{h,\mathbf{x}}$ $\quad \mathbf{WL}_{\mathbf{x}}$

Reminder.

- ALS IFPE: $\boxed{\mathsf{ct} = \big([\mathbf{sA}], [\mathbf{z} - \mathbf{sAW}]\big)}$, $\boxed{\mathsf{sk}_{\mathbf{Y}} = \mathbf{D} := \mathbf{WY}}$
- partial garbling for 1AWS: $\boxed{\mathsf{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{wL}_{\mathbf{x}})}$

Combining the Two — Classical FE for 1AWS

$$\left.\begin{array}{ll} \mathsf{FE.ct} & \big([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]\big) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{WL_x} \end{array}\right\} \longrightarrow \Big(\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\overset{\text{“variable random pad” } w = sAW}{\mathbf{sAW}}\mathbf{L_x}]}_{=: [p_2]}\Big)$$

Reminder.

- ALS IFPE: $\boxed{\mathsf{ct} = ([\mathbf{sA}], [\mathbf{z} - \mathbf{sAW}])}$, $\boxed{\mathsf{sk}_\mathbf{Y} = \mathbf{D} := \mathbf{WY}}$
- partial garbling for 1AWS: $\boxed{\mathsf{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{wL_x})}$

Combining the Two — Classical FE for 1AWS

note: this is not the actual 1AWS functionality

“variable random pad” $w = sAW$

$$\left.\begin{array}{ll} \text{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]) \\ \text{FE.sk}_{h,\mathbf{x}} & \mathbf{W}\mathbf{L}_{\mathbf{x}} \end{array}\right\} \longrightarrow ([\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}], [\mathbf{sAWL}_{\mathbf{x}}])$$

$[p_1] :=$ $=: [p_2]$

Reminder.

- ALS IFPE: $\boxed{\mathsf{ct} = ([\mathbf{sA}], [\mathbf{z} - \mathbf{sAW}])}$, $\boxed{\mathsf{sk}_{\mathbf{Y}} = \mathbf{D} := \mathbf{WY}}$
- partial garbling for 1AWS: $\boxed{\mathsf{pgb}(h, \mathbf{x}, \mathbf{z}; \mathbf{w}) = (\mathbf{p}_1, \mathbf{p}_2) := (\mathbf{z} - \underline{\mathbf{w}}, \mathbf{w}\mathbf{L}_{\mathbf{x}})}$

RFE for a Single User

“variable random pad” $w = sAW$

$$\left.\begin{array}{ll} \mathsf{FE.ct} & \big([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]\big) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{WL_x} \end{array}\right\} \longrightarrow \Big(\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\overbrace{\mathbf{sAW}}^{w}\mathbf{L_x}]}_{=: [p_2]}\Big)$$

RFE for a Single User

crs $\quad (\underbrace{[\mathbf{A}], [\mathbf{A}\mathbf{W}]}_{FE.mpk})$

"variable random pad" $w = sAW$

$$\left.\begin{array}{ll} \mathsf{FE.ct} & ([\mathbf{s}\mathbf{A}], [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{W}\mathbf{L}_{\mathbf{x}} \end{array}\right\} \longrightarrow ([\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}], [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}}])$$

$[p_1] :=$ $\quad =: [p_2]$

RFE for a Single User

crs $\quad \big(\underbrace{[\mathbf{A}],[\mathbf{AW}]}_{FE.mpk}\big)$

(pk, sk) $\quad \big([\mathbf{AU}],\mathbf{U}\big)$ *(for a random matrix U)*

"variable random pad" $w = sAW$

$$\left.\begin{array}{ll} \text{FE.ct} & \big([\mathbf{sA}],[\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}]\big) \\ \text{FE.sk}_{h,\mathbf{x}} & \mathbf{WL_x} \end{array}\right\} \rightarrow \big(\underset{[p_1]:=}{[\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}]},\underset{=:[p_2]}{[\mathbf{sAWL_x}]}\big)$$

RFE for a Single User

crs $\big(\underbrace{[\mathbf{A}], [\mathbf{AW}]}_{\textit{FE.mpk}}\big)$

(pk, sk) $([\mathbf{AU}], \mathbf{U})$ *(for a random matrix U)*

mpk $([\mathbf{A}], [\mathbf{A}\underline{\mathbf{W}}], [\mathbf{AU} + \mathbf{AWL}_{\mathbf{x}}])$

ct $\big(\underbrace{[\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{\textit{FE.ct}}, \underbrace{[\mathbf{sAU} + \mathbf{sAWL}_{\mathbf{x}}]}_{\textit{Enc(pk, FE.sk}_{h,x}\textit{)}}\big)$

$$\left.\begin{array}{ll} \mathsf{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{WL}_{\mathbf{x}} \end{array}\right\} \longrightarrow \big(\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\overset{\text{“variable random pad” } w = sAW}{\mathbf{sAW}}\mathbf{L}_{\mathbf{x}}]}_{=: [p_2]}\big)$$

RFE for a Single User

crs $\underbrace{([\mathbf{A}],[\mathbf{AW}])}_{\textit{FE.mpk}}$

(pk, sk) $([\mathbf{AU}],\mathbf{U})$ *(for a random matrix U)*

mpk $([\mathbf{A}],[\mathbf{A}\underline{\mathbf{W}}],[\mathbf{AU}+\mathbf{AWL}_{\mathbf{x}}])$

ct $(\underbrace{[\mathbf{sA}],[\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}]}_{\textit{FE.ct}},\underbrace{[\mathbf{sAU}+\mathbf{sAWL}_{\mathbf{x}}]}_{\textit{Enc}(pk,\ \textit{FE.sk}_{h,x})})$

Security.

1) *sk=U is secret (i.e. user honest):*
-> nothing revealed under $MDDH_k$

$$\left.\begin{array}{ll} \text{FE.ct} & ([\mathbf{sA}],[\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}]) \\ \text{FE.sk}_{h,\mathbf{x}} & \mathbf{WL}_{\mathbf{x}} \end{array}\right\} \longrightarrow ([\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}],[\mathbf{sAWL}_{\mathbf{x}}])$$

"variable random pad" w = sAW

$[p_1] :=$

$=: [p_2]$

RFE for a Single User

crs $\quad (\underbrace{[\mathbf{A}], [\mathbf{AW}]}_{\textit{FE.mpk}})$

(pk, sk) $\quad ([\mathbf{AU}], \mathbf{U})$ *(for a random matrix U)*

mpk $\quad ([\mathbf{A}], [\mathbf{A\underline{W}}], [\mathbf{AU} + \mathbf{AWL_x}])$

ct $\quad (\underbrace{[\mathbf{sA}], [\mathbf{z} - \mathbf{sA\underline{W}}]}_{\textit{FE.ct}}, \underbrace{[\mathbf{sAU} + \mathbf{sAWL_x}]}_{\textit{Enc(pk, FE.sk}_{h,x}\textit{)}})$

Security.

1) *sk=U is secret (i.e. user honest):*
   -> nothing revealed under $MDDH_k$

2) *sk=U known to A (i.e. user corrupted):*
   -> only $zh(x)^T$ revealed under security of pgb

$$\left.\begin{array}{ll} \text{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA\underline{W}}]) \\ \text{FE.sk}_{h,\mathbf{x}} & \mathbf{WL_x} \end{array}\right\} \rightarrow (\underbrace{[\mathbf{z} - \mathbf{sA\underline{W}}]}_{[p_1] :=}, [\underbrace{\overset{\text{“variable random pad” } w = sAW}{\mathbf{sAW}}\mathbf{L_x}}_{=: [p_2]}])$$

RFE for Multiple Users

crs

$(\mathsf{pk}_i, \mathsf{sk}_i)$

mpk

ct

$$\left.\begin{array}{ll} \mathsf{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{W}\mathbf{L}_{\mathbf{x}} \end{array}\right\} \longrightarrow \Big(\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\overbrace{\mathbf{sAW}}^{\text{“variable random pad” } w = sAW}\mathbf{L}_{\mathbf{x}}]}_{=: [p_2]}\Big)$$

RFE for Multiple Users

crs $\quad \big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]}_{FE.mpk}\}_{i\in[L]}\big)$

$(\mathsf{pk}_i, \mathsf{sk}_i) \quad ([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ *(for random matrices U_i)*

mpk

ct

"variable random pad" $w = sAW$

$$\left.\begin{array}{ll} \mathsf{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{W}\mathbf{L}_{\mathbf{x}} \end{array}\right\} \longrightarrow \big(\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\mathbf{sAWL}_{\mathbf{x}}]}_{=: [p_2]}\big)$$

RFE for Multiple Users

crs $\big(\underbrace{[\mathbf{A}], \{[\mathbf{AW}_i]\}}_{FE.mpk}{}_{i\in[L]}\big)$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $([\mathbf{AU}_i], \mathbf{U}_i)$ *(for random matrices U_i)*

mpk $\big([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{AU}_i + \mathbf{AW}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

ct $\big([\mathbf{sA}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{sA}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{sAU}_i + \mathbf{sAW}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

sum of L independent 1-slot instances

"variable random pad" $w = sAW$

$$\left.\begin{array}{ll} \mathsf{FE.ct} & ([\mathbf{sA}], [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]) \\ \mathsf{FE.sk}_{h,\mathbf{x}} & \mathbf{WL}_\mathbf{x} \end{array}\right\} \longrightarrow (\underbrace{[\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]}_{[p_1] :=}, \underbrace{[\mathbf{sAWL}_\mathbf{x}]}_{=: [p_2]})$$

RFE for Multiple Users

crs $$\left([\mathbf{A}], \underbrace{\{[\mathbf{A}\mathbf{W}_i]\}}_{\textit{FE.mpk}}{}_{i\in[L]}\right)$$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $$([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$$ *(for random matrices U_i)*

mpk $$\left([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$$

ct $$\left([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$$

sum of L independent 1-slot instances

RFE for Multiple Users

crs $\Big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\textit{FE.mpk}}{}_{i\in[L]}\Big)$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ *(for random matrices U_i)*

mpk $\Big([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\Big)$

ct $\Big([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\Big)$

sum of L independent 1-slot instances

... how to decrypt? -> helper secret keys

RFE for Multiple Users

crs $\quad \big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\text{FE.mpk}}{}_{i\in[L]}\big)$

$(\mathsf{pk}_i, \mathsf{sk}_i) \quad ([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ *(for random matrices U_i)*

mpk $\quad \big([\mathbf{A}], \underline{\sum_{i\in[L]}}[\mathbf{A}\mathbf{W}_i], \underline{\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

ct $\quad \big([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\sum_{i\in[L]}} \mathbf{s}\mathbf{A}\mathbf{W}_i], \underline{\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

sum of L independent 1-slot instances *... how to decrypt? -> helper secret keys*

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = \big(\sum_{i\in[L]\setminus j} \mathbf{W}_i, \sum_{i\in[L]\setminus j} \mathbf{U}_i\big)$

RFE for Multiple Users

crs	$\left(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]}_{\text{FE.mpk}}\}_{i\in[L]}\right)$
$(\mathsf{pk}_i, \mathsf{sk}_i)$	$([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ (for random matrices U_i)
mpk	$\left([\mathbf{A}], \underline{\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$
ct	$\left([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\sum_{i\in[L]}} \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$

sum of L independent 1-slot instances

... how to decrypt? -> helper secret keys

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = \left(\sum_{i\in[L]\setminus j} \mathbf{W}_i, \sum_{i\in[L]\setminus j} \mathbf{U}_i\right)$
- problem 1: helper secret key contains scalar values

RFE for Multiple Users

crs	$\big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\textit{FE.mpk}}{}_{i\in[L]}\big)$
$(\mathsf{pk}_i, \mathsf{sk}_i)$	$([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ *(for random matrices U_i)*
mpk	$([\mathbf{A}], \underline{\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}_i}], \underline{\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}])$
ct	$([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\sum_{i\in[L]}}\, \mathbf{s}\mathbf{A}\underline{\mathbf{W}_i}], \underline{\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}])$

sum of L independent 1-slot instances

... how to decrypt? -> helper secret keys

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = (\sum_{i\in[L]\setminus j} \mathbf{W}_i, \sum_{i\in[L]\setminus j} \mathbf{U}_i)$
- problem 1: helper secret key contains scalar values
- solution 1: switch to pairing group with ciphertexts in $\mathbb{G}_1$ and helper secret keys in $\mathbb{G}_2$

RFE for Multiple Users

crs $\big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\textit{FE.mpk}}{}_{i \in [L]}\big)$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $\big([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i\big)$ *(for random matrices U_i)*

mpk $\big([\mathbf{A}], \underline{\sum_{i \in [L]}} [\mathbf{A}\underline{\mathbf{W}}_i], \underline{\sum_{i \in [L]}} [\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

ct $\big([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\sum_{i \in [L]}} \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\sum_{i \in [L]}} [\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

sum of L independent 1-slot instances *... how to decrypt? -> helper secret keys*

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = \big(\sum_{i \in [L] \setminus j} [\mathbf{W}_i]_2, \sum_{i \in [L] \setminus j} [\mathbf{U}_i]_2\big)$

RFE for Multiple Users

crs $$\underbrace{([\mathbf{A}], \{[\mathbf{AW}_i]\}}_{\text{FE.mpk}}{}_{i\in[L]})$$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $$([\mathbf{AU}_i], \mathbf{U}_i)$$ *(for random matrices U_i)*

mpk $$([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{AU}_i + \mathbf{AW}_i\mathbf{L}_{i,\mathbf{x}}])$$

ct $$([\mathbf{sA}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{sA}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{sAU}_i + \mathbf{sAW}_i\mathbf{L}_{i,\mathbf{x}}])$$

sum of L independent 1-slot instances

... how to decrypt? -> helper secret keys

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = (\sum_{i\in[L]\setminus j}[\mathbf{W}_i]_2, \sum_{i\in[L]\setminus j}[\mathbf{U}_i]_2)$
- problem 2: masking terms for different users are correlated

RFE for Multiple Users

crs $$\left(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\text{FE.mpk}}{}_{i\in[L]}\right)$$

$(\mathsf{pk}_i, \mathsf{sk}_i)$ $$([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$$ (for random matrices U_i)

mpk $$\left([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$$

ct $$\left([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\right)$$

sum of L independent 1-slot instances

... how to decrypt? -> helper secret keys

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = (\sum_{i\in[L]\setminus j}[\mathbf{W}_i]_2, \sum_{i\in[L]\setminus j}[\mathbf{U}_i]_2)$
- problem 2: masking terms for different users are correlated
- (partial) solution 2: user-specific re-randomization of helper secret keys

RFE for Multiple Users

crs $\quad \Big(\underbrace{[\mathbf{A}], \{[\mathbf{A}\mathbf{W}_i]\}}_{\textit{FE.mpk}}{}_{i\in[L]}\Big)$

$(\mathsf{pk}_i, \mathsf{sk}_i) \quad ([\mathbf{A}\mathbf{U}_i], \mathbf{U}_i)$ *(for random matrices U_i)*

mpk $\quad \big([\mathbf{A}], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{A}\mathbf{U}_i + \mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

ct $\quad \big([\mathbf{s}\mathbf{A}], [\mathbf{z} - \underline{\textstyle\sum_{i\in[L]}}\, \mathbf{s}\mathbf{A}\underline{\mathbf{W}}_i], \underline{\textstyle\sum_{i\in[L]}}[\mathbf{s}\mathbf{A}\mathbf{U}_i + \mathbf{s}\mathbf{A}\mathbf{W}_i\mathbf{L}_{i,\mathbf{x}}]\big)$

sum of L independent 1-slot instances *... how to decrypt? -> helper secret keys*

Intuition.

- user j could decrypt given $\mathsf{hsk}_j = (\sum_{i\in[L]\setminus j}[\mathbf{W}_i]_2, \sum_{i\in[L]\setminus j}[\mathbf{U}_i]_2)$
- problem 2: masking terms for different users are correlated
- (partial) solution 2: user-specific re-randomization of helper secret keys

$$\mathsf{hsk}_j = ([\mathbf{B}\mathbf{r}_j^\top]_2, \textstyle\sum_{i\in[L]\setminus j}[\mathbf{W}_i\mathbf{B}\mathbf{r}_j^\top]_2, \sum_{i\in[L]\setminus j}[\mathbf{U}_i\mathbf{B}\mathbf{r}_j^\top]_2)$$

Pad Re-Randomization

$$\overset{\text{ciphertext}}{[\mathbf{p}_1]_1} \cdot \overset{\text{helper secret key}}{[\underline{\mathbf{R}}]_2} = [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}} \cdot \mathbf{R}]_t$$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix

Pad Re-Randomization

ciphertext *helper secret key* **problem 1:** *input vector changes*

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{sA}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x} \cdot \mathbf{R}]_t$$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix

Pad Re-Randomization

ciphertext *helper secret key* *problem 1: input vector changes*

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{sA}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding

should be $sAWR \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix

Pad Re-Randomization

ciphertext *helper secret key* *problem 1: input vector changes*

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{sA}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{sA}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{sAWL_x} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding should be $sAWR \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix
- solution 1: $\mathbf{R} = (\mathbf{I} \otimes \mathbf{r}^\top)$ for $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^k$ (tensored ALS encodings)

Pad Re-Randomization

ciphertext *helper secret key* *problem 1: input vector changes*

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_\mathbf{x}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_\mathbf{x} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding

should be $sAWR \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix
- solution 1: $\mathbf{R} = (\mathbf{I} \otimes \mathbf{r}^\top)$ for $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^k$ (**tensored** ALS encodings)

mixed-product property:

$(A \otimes B)(C \otimes D) = (AC \otimes BD)$

Pad Re-Randomization

ciphertext

helper secret key

problem 1: input vector changes

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding

should be $sAWR \cdot L_x$

-> $sAWL_x \cdot (I \otimes r^T) = sAW(I \otimes r^T) \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix
- solution 1: $\mathbf{R} = (\mathbf{I} \otimes \mathbf{r}^\top)$ for $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^k$ (**tensored** ALS encodings)

mixed-product property:

$(A \otimes B)(C \otimes D) = (AC \otimes BD)$

Pad Re-Randomization

ciphertext *helper secret key* *problem 1: input vector changes* *-> encode $z \otimes sA$ and decode in new basis sAr^T*

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_\mathbf{x}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_\mathbf{x} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding should be $sAWR \cdot L_x$

-> $sAWL_x \cdot (I \otimes r^T) = sAW(I \otimes r^T) \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix
- solution 1: $\mathbf{R} = (\mathbf{I} \otimes \mathbf{r}^\top)$ for $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^k$ (**tensored** ALS encodings)

mixed-product property:

$(A \otimes B)(C \otimes D) = (AC \otimes BD)$

Pad Re-Randomization

ciphertext

helper secret key

problem 1: input vector changes

-> encode $z \otimes sA$ and decode in new basis sAR

$$[\mathbf{p}_1]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}}]_1 \cdot [\underline{\mathbf{R}}]_2 = [\mathbf{z} \cdot \underline{\mathbf{R}} - \mathbf{s}\mathbf{A}\underline{\mathbf{W}} \cdot \underline{\mathbf{R}}]_t$$

$$[\mathbf{p}_2]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}}]_1 \cdot [\mathbf{R}]_2 = [\mathbf{s}\mathbf{A}\mathbf{W}\mathbf{L}_{\mathbf{x}} \cdot \mathbf{R}]_t$$

problem 2: correctly randomized encoding

should be $sAWR \cdot L_x$

-> $sAWL_x \cdot (I \otimes r^T) = sAW(I \otimes r^T) \cdot L_x$

Question: how to choose $\mathbf{R}$?

- naive approach: a random (uniform) matrix
- solution 1: $\mathbf{R} = (\mathbf{I} \otimes \mathbf{r}^\top)$ for $\mathbf{r} \leftarrow_\$ \mathbb{Z}_p^k$ (**tensored** ALS encodings)
- solution 2: use different ALS keys (**nested** ALS encodings)

mixed-product property:

$(A \otimes B)(C \otimes D) = (AC \otimes BD)$

Solution 2: Nested ALS Encodings

$$\left.\begin{aligned}
[\mathbf{p}_{1,\text{in}}]_t &= \overset{\text{ciphertext}}{[\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}_{\text{in}}]_1} \cdot \overset{\text{helper secret key}}{[\mathbf{I}]_2} = [\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}_{\text{in}}]_t \\
[\mathbf{p}_{1,\text{out}}]_t &= [\mathbf{sA}]_1 \cdot [\underline{\mathbf{W}}_{\text{in}} - \underline{\mathbf{W}}_{\text{out}}]_2 = [\mathbf{sA}\underline{\mathbf{W}}_{\text{in}} - \mathbf{sA}\underline{\mathbf{W}}_{\text{out}}]_t \\
[\mathbf{p}_2]_t &= [\mathbf{sA}]_1 \cdot [\mathbf{W}_{\text{out}}\mathbf{L}_{\mathbf{x}}]_2 = [\mathbf{sAW}_{\text{out}}\mathbf{L}_{\mathbf{x}}]_t
\end{aligned}\right\} \longrightarrow
\begin{aligned}
[\mathbf{p}_1]_t &= [\mathbf{p}_{1,\text{in}}]_t + [\mathbf{p}_{1,\text{out}}]_t = [\mathbf{z}-\mathbf{sA}\underline{\mathbf{W}}_{\text{out}}]_t \\
[\mathbf{p}_2]_t &= [\mathbf{sAW}_{\text{out}}\mathbf{L}_{\mathbf{x}}]_t
\end{aligned}$$

Conclusion

- **classical FE** provides security against malicious user but needs to trust authority
- **registered FE** circumvents the need for trusted authority

Conclusion

- **classical FE** provides security against malicious user but needs to trust authority
- **registered FE** circumvents the need for trusted authority

- **this work:**
  - RFE for **1AWS** with adaptive security using **tensored ALS encodings**
  - RFE for **AB-AWS** with selective security using **nested ALS encodings**

Conclusion

- **classical FE** provides security against malicious user but needs to trust authority
- **registered FE** circumvents the need for trusted authority

- **this work:**
  - RFE for **1AWS** with adaptive security using **tensored ALS encodings**
  - RFE for **AB-AWS** with selective security using **nested ALS encodings**

- **follow-up work:**
  - **modular framework** (pre-constrained IP-RFE + garbling scheme)
  - **new functionalities** (AB-AWS and AB-QF for log-space TMs)

Conclusion

- **classical FE** provides security against malicious user but needs to trust authority
- **registered FE** circumvents the need for trusted authority

- **this work:**
  - RFE for **1AWS** with adaptive security using **tensored ALS encodings**
  - RFE for **AB-AWS** with selective security using **nested ALS encodings**

- **follow-up work:**
  - **modular framework** (pre-constrained IP-RFE + garbling scheme)
  - **new functionalities** (AB-AWS and AB-QF for log-space TMs)

- **open problems:**
  - adaptive security
  - compression of CRS

Conclusion

- **classical FE** provides security against malicious user but needs to trust authority
- **registered FE** circumvents the need for trusted authority

- **this work:**
  - RFE for **1AWS** with adaptive security using **tensored ALS encodings**
  - RFE for **AB-AWS** with selective security using **nested ALS encodings**

- **follow-up work:**
  - **modular framework** (pre-constrained IP-RFE + garbling scheme)
  - **new functionalities** (AB-AWS and AB-QF for log-space TMs)

- **open problems:**
  - adaptive security
  - compression of CRS

Thank you!!! :)