PKC 2025

Røros, Norway

Dynamic Decentralized Functional Encryptions from Pairings in the Standard Model

Duy Nguyen

Telecom Paris, IPP

Open questions in FH-IP DDFE

Open questions in FH-IP DDFE

- **Standard-model security** for FH-IP-DDFE(raised by [Shi and Vanjani-PKC23]) ?

Open questions in FH-IP DDFE

- **Standard-model security** for FH-IP-DDFE(raised by [Shi and Vanjani-PKC23]) ?

- **Adaptive security** for FH-IP DDFE from **any single-user FH-IP FE**?

Open questions in FH-IP DDFE

- **Standard-model security** for FH-IP-DDFE(raised by [Shi and Vanjani-PKC23]) ? ☑
  - Solution: pseudorandom zero-sharing with RO-free share updatability.

- **Adaptive security** for FH-IP DDFE from **any single-user FH-IP FE**? ☑
  - Solution: proof strategy without complexity leveraging.

Open questions in FH-IP DDFE

- **Standard-model security** for FH-IP-DDFE(raised by [Shi and Vanjani-PKC23]) ? ☑
  - Solution: pseudorandom zero-sharing with <u>RO-free share updatability.</u>

- **Adaptive security** for FH-IP DDFE from **any single-user FH-IP FE**? ☑
  - Solution: proof strategy <u>without complexity leveraging.</u>

↓

2-in-1: **modular construction** for FH-IP DDFE with **both security**.

Results and Related Works

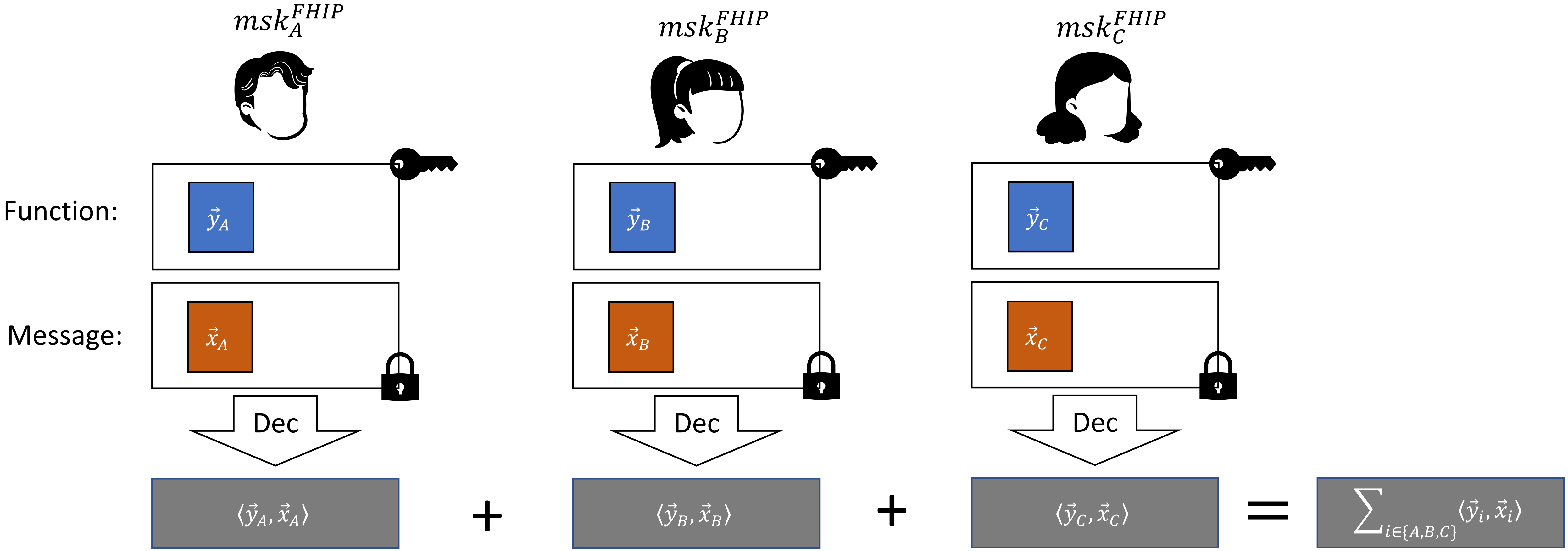
Scheme	Function class	Function hiding	(Dynamic) Decentralized	Without RO	Adaptive key/message queries	Unbounded key/message tag repetitions	Adaptive user-corruption queries	Assumption + (any FE as building block)	Per-client CT size
[CDG+18]	IP	✗	✓	✗	✓	✗	✗	SXDH	$O_\lambda(d)$
[ABG19]	IP	✗	✓	✓	✓	✓	✓	IPFE	$O_\lambda(d \cdot n)$
[LT19]	IP	✗	✓	✓	✓	✗	✗	LWE	$O_\lambda(d)$
[CDSG+20]	IP	✗	✓	✗	✗	✓	✗	DDH + IPFE	$O_\lambda(d)$
[ABM+20]	IP	✗	✓	✗	✓	✓	✗	DCR	$O_\lambda(d)$
[AGT21b]	IP	✓	✓	✗	✗	✗	✗	SXDH +FH-IPFE	$O_\lambda(d)$
[SV23]	IP	✓	✗	✓	✓	✓	✗	DLin	$O_\lambda(d)$
[NPS24]	IP	✓	✓	✗	✓	✗	✗	SXDH	$O_\lambda(d \cdot J)$
Our FH-IP DDFE	IP	✓	✓	✓	✓	✓	✗	SXDH +FH-IPFE	$O_\lambda(d + n)$
[ATY23]	AWS	✗	✓	✗	✗	✓	✗	MDDH +AWSw/IP-FE	$O_\lambda(N(kn_0 + n_1))$
Our AWS DDFE	AWS	✗	✓	✓	✗	✓	✗	SXDH + AWSw/IP-FE	$O_\lambda(N(kn_0 + n_1 + n))$

Comparison with prior (Decentralized) MCFE schemes.

Technique 1: Zero-Sharing for Standard-Model Security

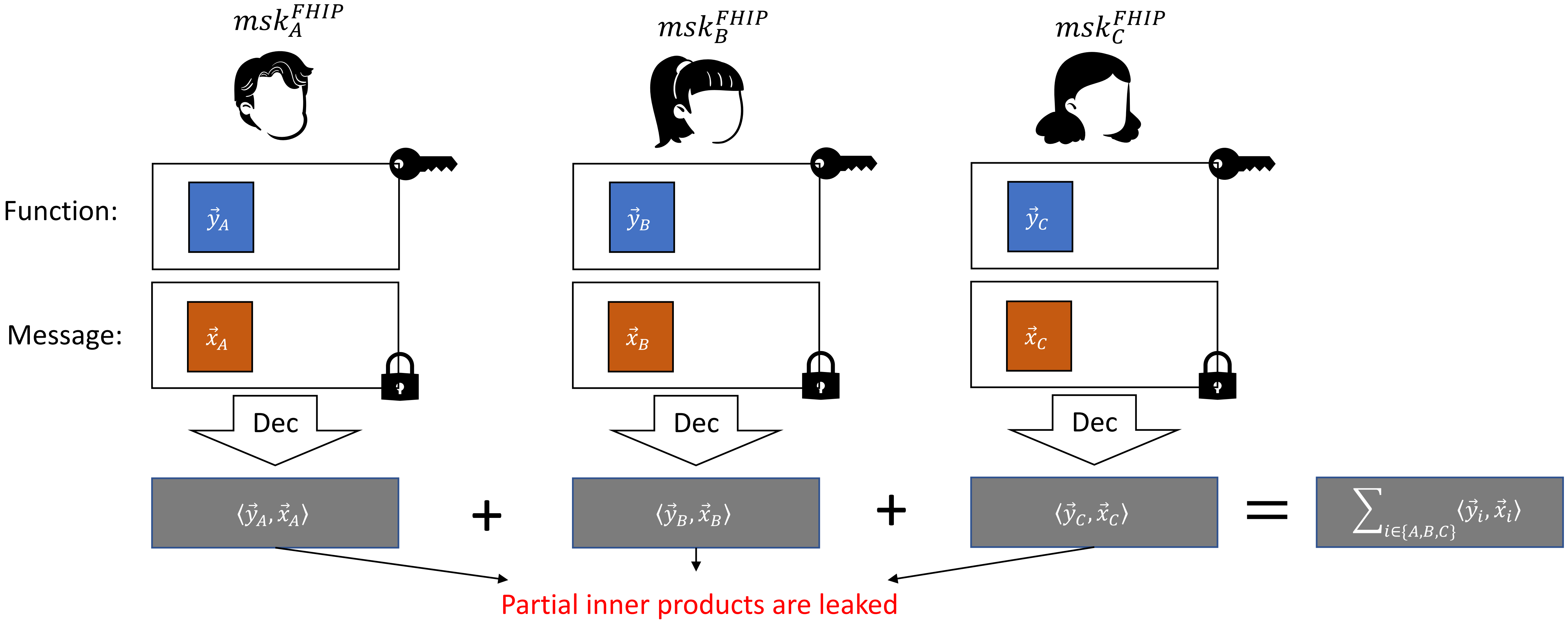
Technique 1: Zero-Sharing for Standard-Model Security

Why zero sharing in Decentralized FE?

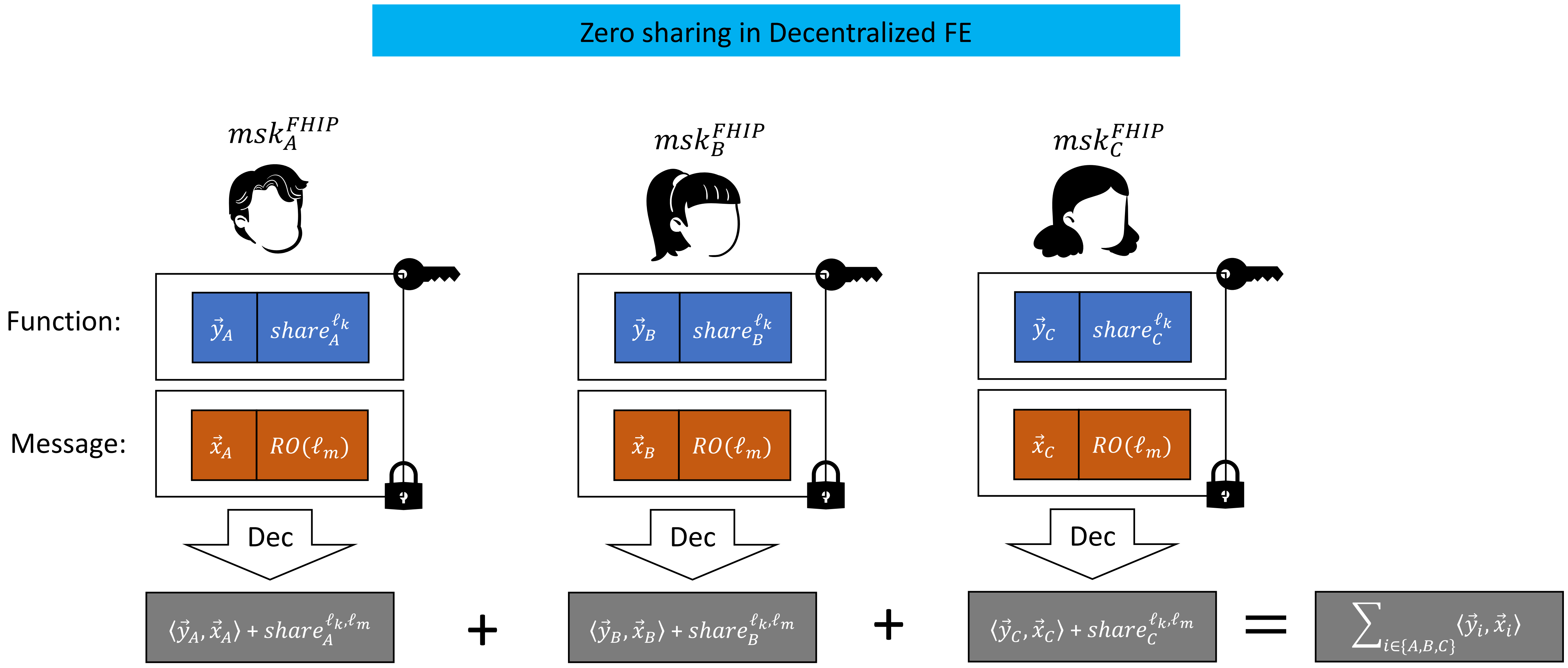


Technique 1: Zero-Sharing for Standard-Model Security

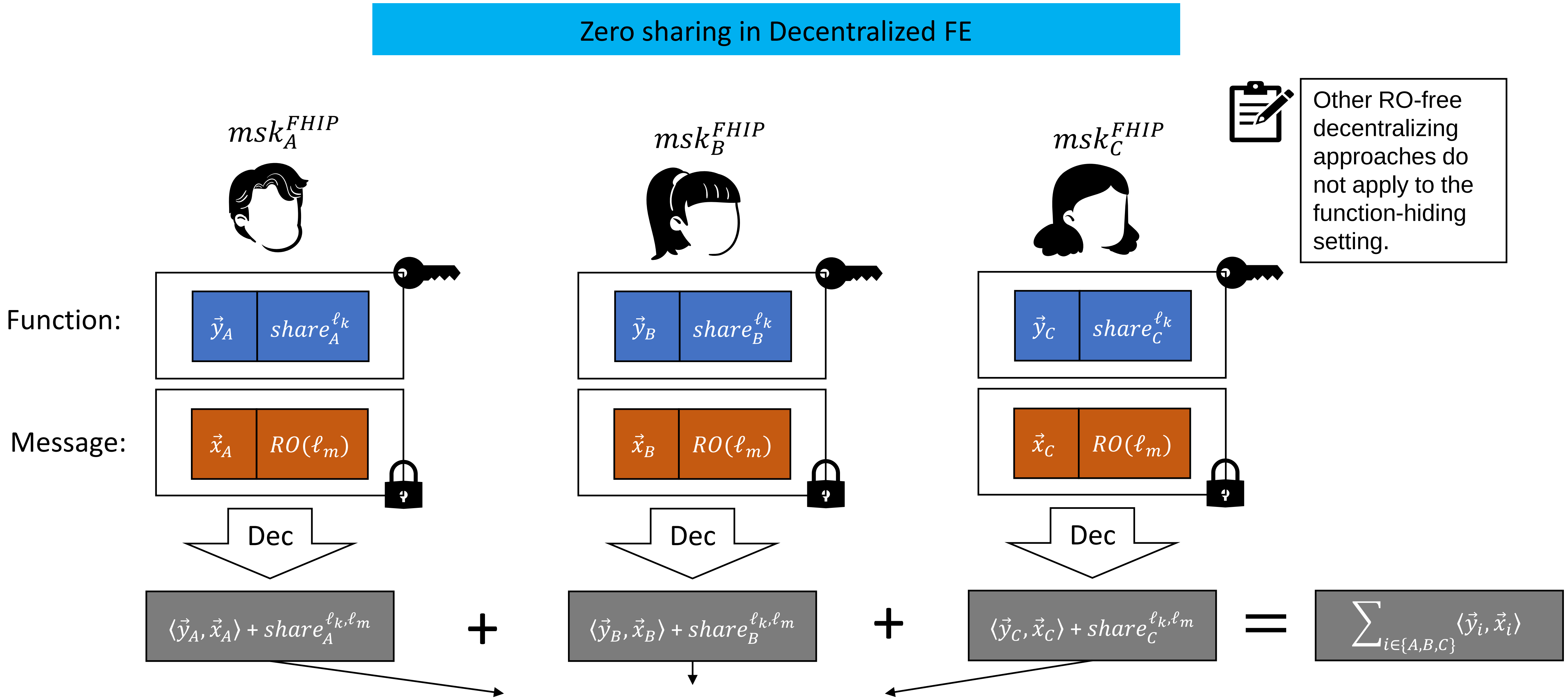
Why zero sharing in Decentralized FE?



Technique 1: Zero-Sharing for Standard-Model Security



Technique 1: Zero-Sharing for Standard-Model Security



RO is required to update share label [Agrawal, Goyal, Tomida-TCC21]

Technique 1: Zero-Sharing for Standard-Model Security

The algebraic form of zero shares

Technique 1: Zero-Sharing for Standard-Model Security

The algebraic form of zero shares

K_{AB}, K_{AC}

$share_A^{\ell_k} = 0 + PRF_{K_{AB}}(\ell_k) + PRF_{K_{AC}}(\ell_k)$

K_{BA}, K_{BC}

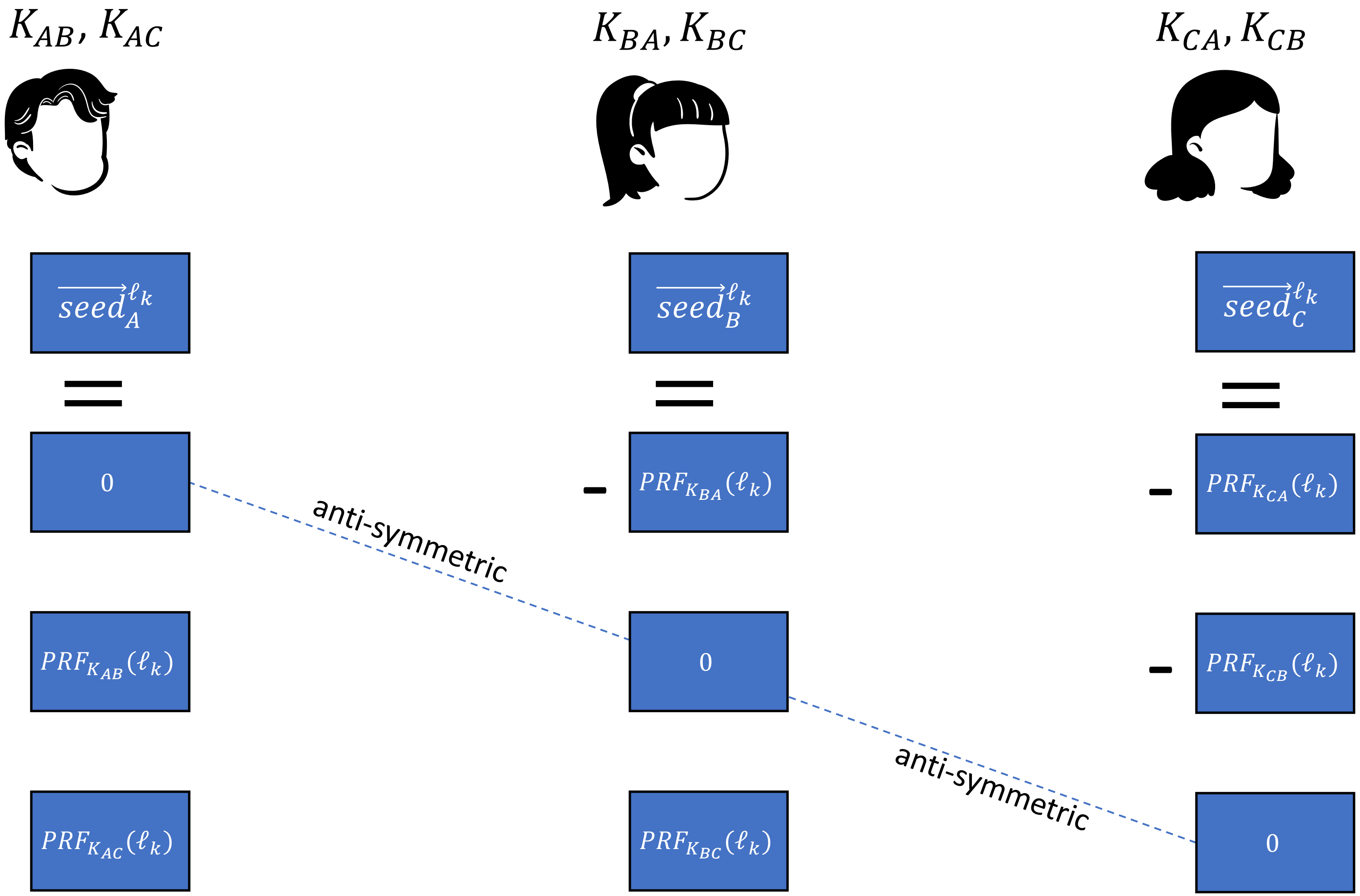
$share_B^{\ell_k} = -PRF_{K_{BA}}(\ell_k) + 0 + PRF_{K_{BC}}(\ell_k)$

K_{CA}, K_{CB}

$share_C^{\ell_k} = -PRF_{K_{CA}}(\ell_k) + -PRF_{K_{CB}}(\ell_k) + 0$

Technique 1: Zero-Sharing for Standard-Model Security

The algebraic form of zero shares



Technique 1: Zero-Sharing for Standard-Model Security

Updating zero shares without RO

K_{AB}, K_{AC}

$$\overrightarrow{seed}_A^{\ell_k} \odot \overrightarrow{update}_A^{\ell_m}$$
$$=$$
$$0 \times 0$$
$$PRF_{K_{AB}}(\ell_k) \times PRF'_{K_{AB}}(\ell_m)$$
$$PRF_{K_{AC}}(\ell_k) \times PRF'_{K_{AC}}(\ell_m)$$

K_{BA}, K_{BC}

$$\overrightarrow{seed}_B^{\ell_k} \odot \overrightarrow{update}_B^{\ell_m}$$
$$=$$
$$-\, PRF_{K_{BA}}(\ell_k) \times PRF'_{K_{BA}}(\ell_m)$$
$$0 \times 0$$
$$PRF_{K_{BC}}(\ell_k) \times PRF'_{K_{BC}}(\ell_m)$$

K_{CA}, K_{CB}

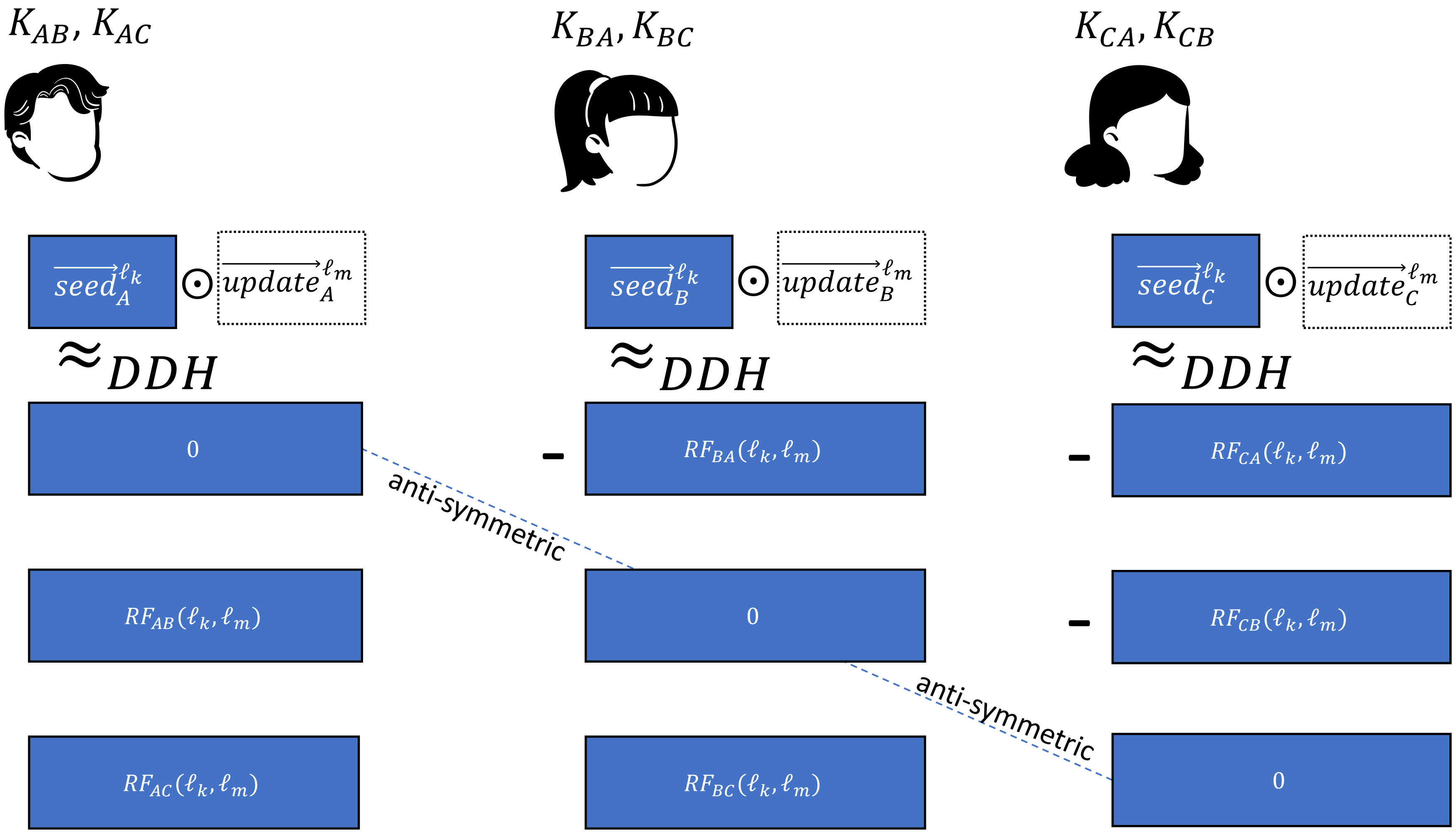
$$\overrightarrow{seed}_C^{\ell_k} \odot \overrightarrow{update}_C^{\ell_m}$$
$$=$$
$$-\, PRF_{K_{CA}}(\ell_k) \times PRF'_{K_{CA}}(\ell_m)$$
$$-\, PRF_{K_{CB}}(\ell_k) \times PRF'_{K_{CB}}(\ell_m)$$
$$0 \times 0$$

anti-symmetric

anti-symmetric

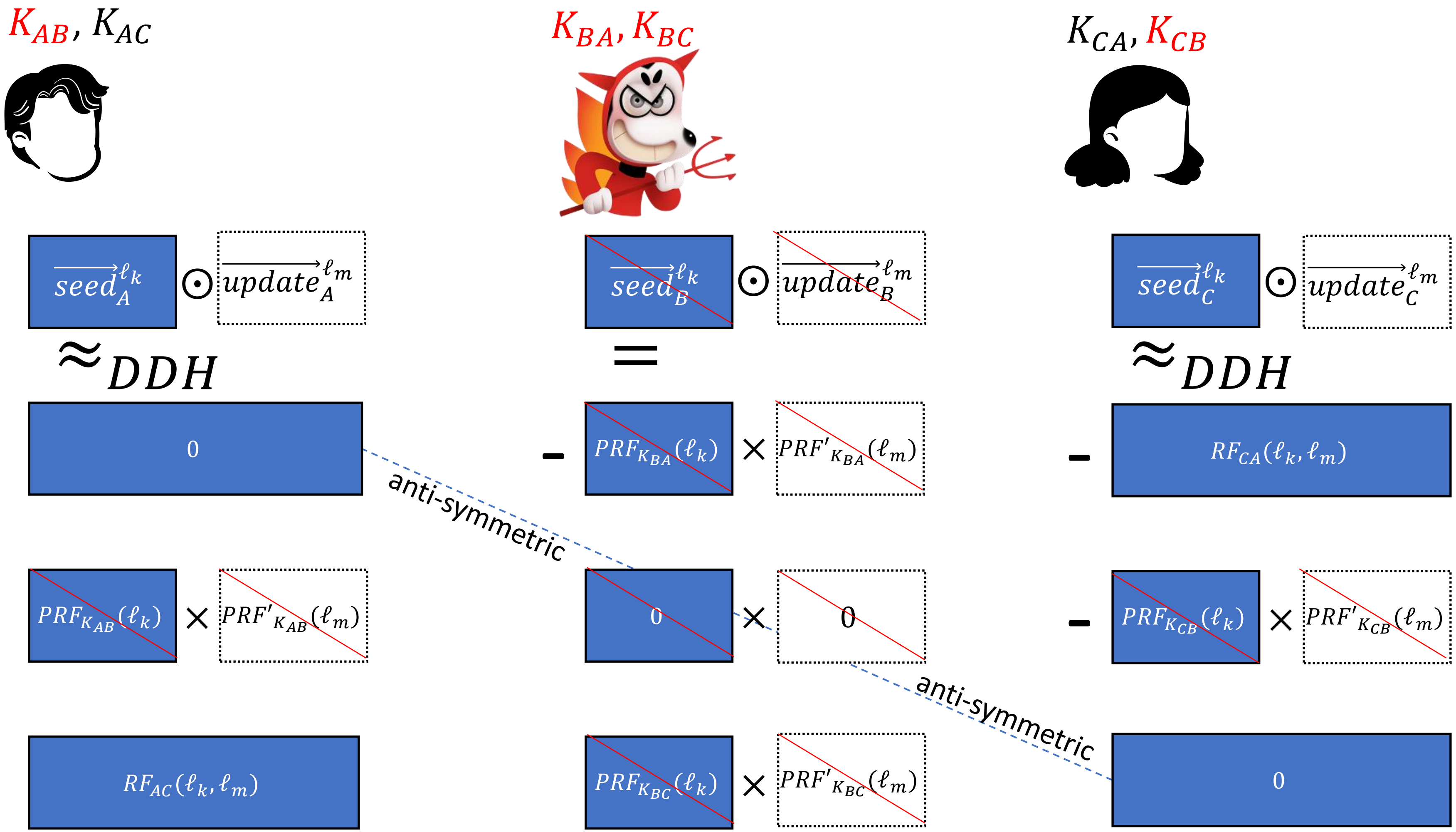
Technique 1: Zero-Sharing for Standard-Model Security

Security of the updated shares



Technique 1: Zero-Sharing for Standard-Model Security

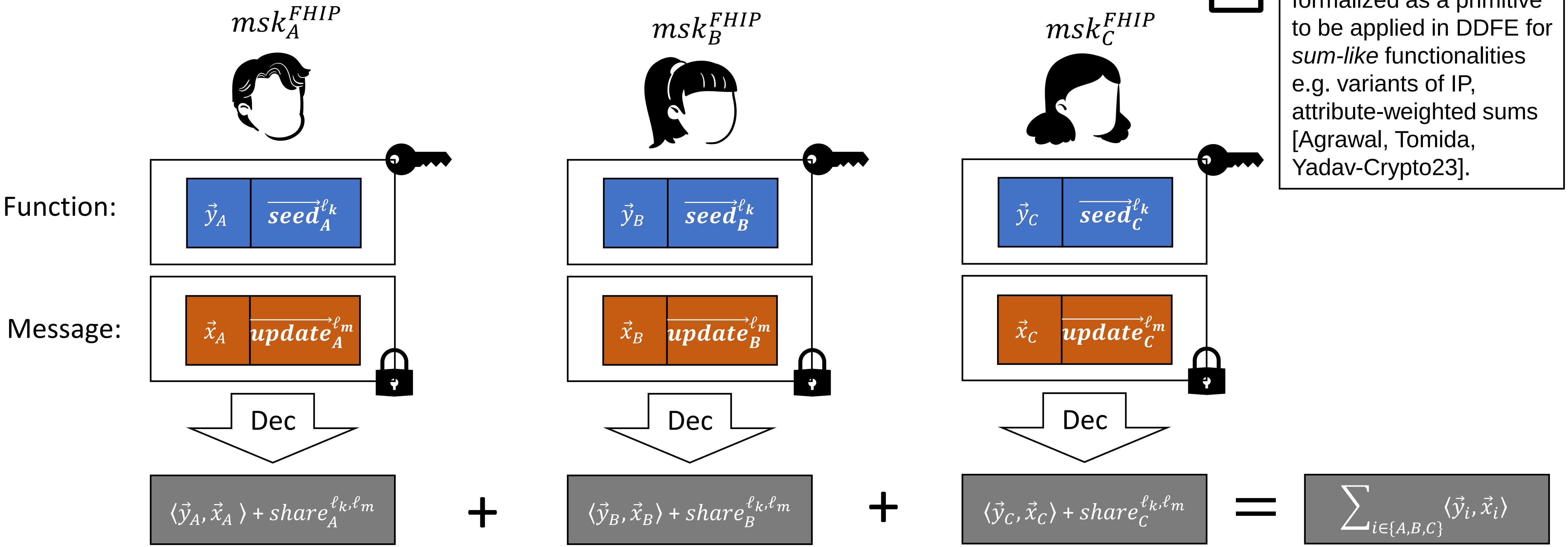
Security of the updated shares



Technique 1: Zero-Sharing for Standard-Model Security

Decentralized FE with RO-free Updatable Zero-Sharing

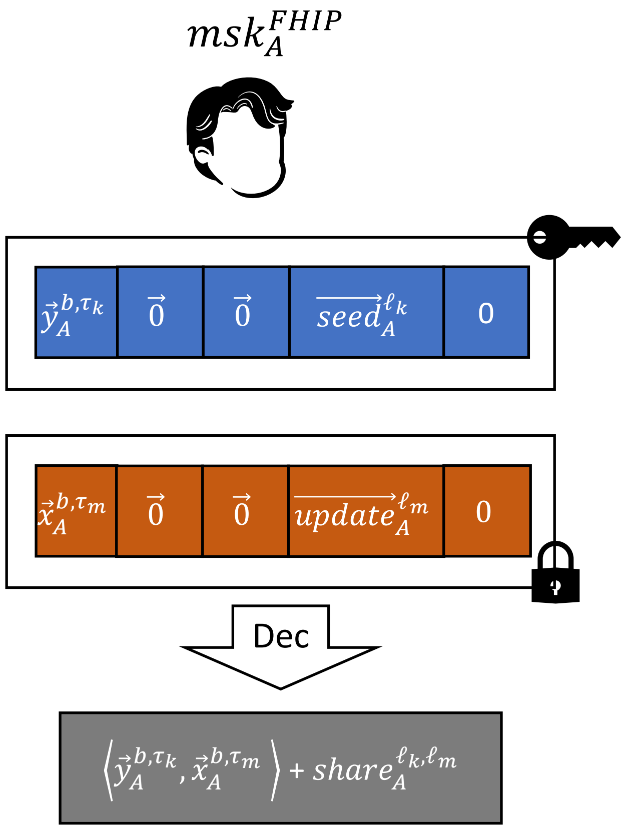
The technique is formalized as a primitive to be applied in DDFE for *sum-like* functionalities e.g. variants of IP, attribute-weighted sums [Agrawal, Tomida, Yadav-Crypto23].



Technique 2: Proof for Adaptive Security Preserving

Technique 2: Proof for Adaptive Security Preserving

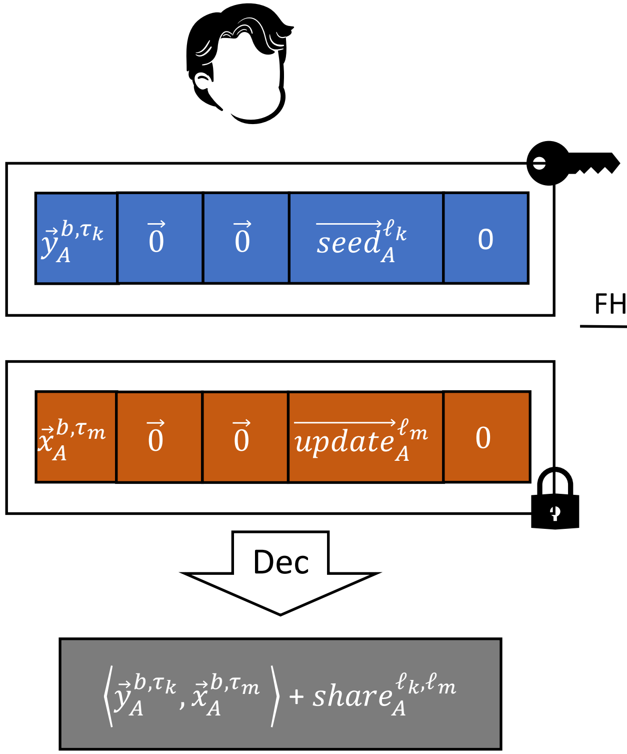
Scheme in game-based security



Technique 2: Proof for Adaptive Security Preserving

Semi-functional mode

msk_A^{FHIP}



FH-IPFE, UZS $\rightarrow$

all keys:

$\vec{y}_A^{b,\tau_k}$	$\vec{y}_A^{0,\tau_k}$	$\vec{y}_A^{0,\tau_k}$	$\overrightarrow{seed}_A^{\ell_k}$	$share_A^{\ell_k,\ell_m}$

ciphertexts under the target label:

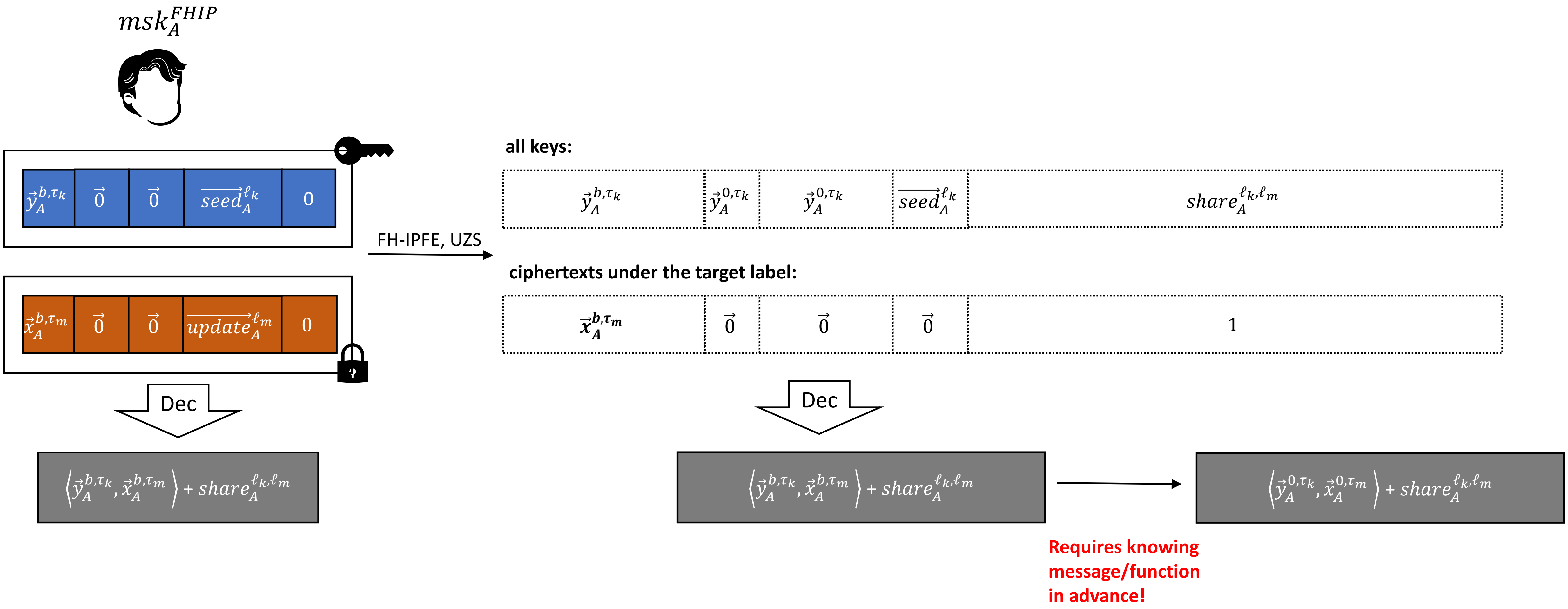
$\vec{x}_A^{b,\tau_m}$	$\vec{0}$	$\vec{0}$	$\vec{0}$	1

Dec

$\left\langle \vec{y}_A^{b,\tau_k}, \vec{x}_A^{b,\tau_m} \right\rangle + share_A^{\ell_k,\ell_m}$

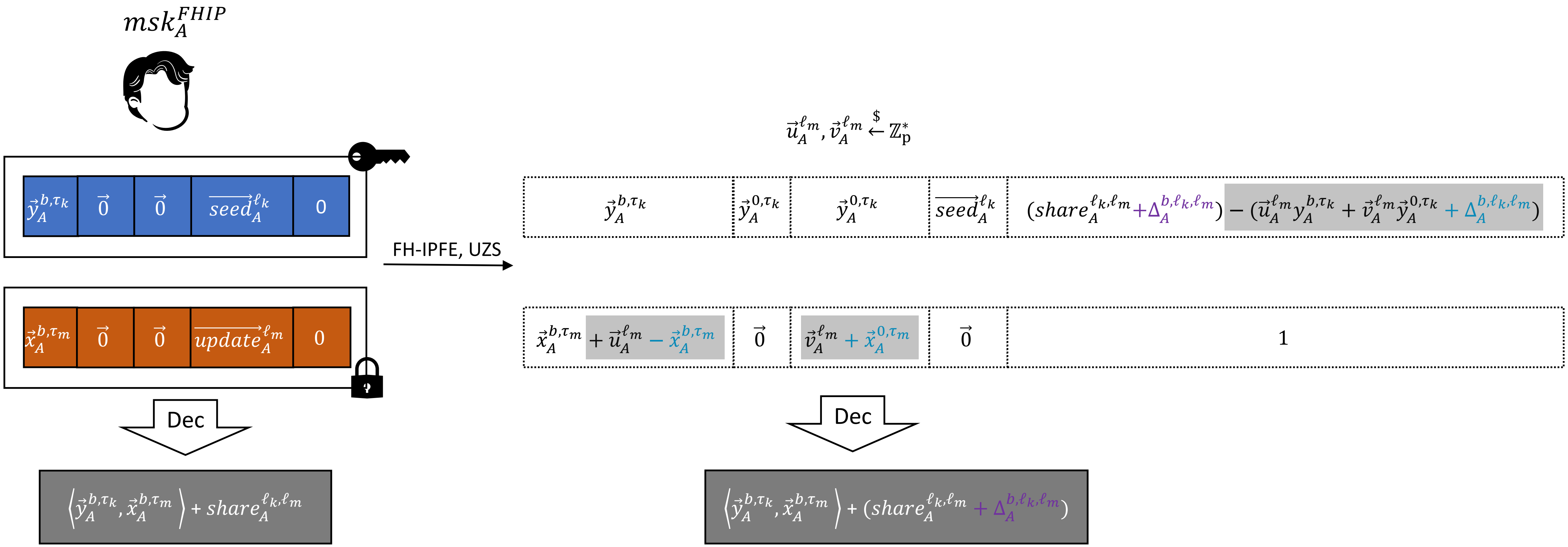
Technique 2: Proof for Adaptive Security Preserving

Semi-functional mode



Technique 2: Proof for Adaptive Security Preserving

Novel Padding Strategy in Security Proof



For all non-corrupted users A, the difference $\Delta_A^{b,\ell_m,\ell_k} = \left\langle \vec{y}_A^{0,\tau_k}, \vec{x}_A^{0,\tau_m} \right\rangle - \left\langle \vec{y}_A^{b,\tau_k}, \vec{x}_A^{b,\tau_m} \right\rangle$

1. *is **invariant** across all repetition index (τ_m, τ_k);*
2. *forms **zero shares** with differences of other non-corrupted users.*

Technique 2: Proof for Adaptive Security Preserving

Novel Padding Strategy in Security Proof

msk_A^{FHIP}

$\vec{y}_A^{b,\tau_k}$	$\vec{0}$	$\vec{0}$	$\overrightarrow{seed}_A^{\ell_k}$	0

$\vec{x}_A^{b,\tau_m}$	$\vec{0}$	$\vec{0}$	$\overrightarrow{update}_A^{\ell_m}$	0

Dec

$\left\langle \vec{y}_A^{b,\tau_k}, \vec{x}_A^{b,\tau_m} \right\rangle + share_A^{\ell_k,\ell_m}$

FH-IPFE, UZS

$\vec{u}_A^{\ell_m}, \vec{v}_A^{\ell_m} \xleftarrow{\$} \mathbb{Z}_p^*$

$\vec{y}_A^{b,\tau_k}$	$\vec{y}_A^{0,\tau_k}$	$\vec{y}_A^{0,\tau_k}$	$\overrightarrow{seed}_A^{\ell_k}$	$share_A^{\ell_k,\ell_m} - (\vec{u}_A^{\ell_m} \vec{y}_A^{b,\tau_k} + \vec{v}_A^{\ell_m} \vec{y}_A^{0,\tau_k})$

$\vec{u}_A^{\ell_m}$	$\vec{0}$	$\vec{v}_A^{\ell_m} + \boldsymbol{\vec{x}_A^{0,\tau_m}}$	$\vec{0}$	1

Dec

$\left\langle \vec{y}_A^{b,\tau_k}, \vec{x}_A^{b,\tau_m} \right\rangle + (share_A^{\ell_k,\ell_m} + \Delta_A^{b,\ell_k,\ell_m})$ $=$ $\left\langle \vec{y}_A^{0,\tau_k}, \vec{x}_A^{0,\tau_m} \right\rangle + share_A^{\ell_k,\ell_m}$

For all non-corrupted users A, the difference $\Delta_A^{b,\ell_m,\ell_k} = \left\langle \vec{y}_A^{0,\tau_k}, \vec{x}_A^{0,\tau_m} \right\rangle - \left\langle \vec{y}_A^{b,\tau_k}, \vec{x}_A^{b,\tau_m} \right\rangle$

*1. is **invariant** across all repetition index (τ_m, τ_k);*
*2. forms **zero shares** with differences of other non-corrupted users.*

Conclusion

	DDFE: Generic Constructions with Strong Security	DDFE from Pairings in the Standard Model
Settings	1. Standard assumptions 2. Possibly in ROM 3. With/without pairings	1. Standard assumptions 2. Standard model 3. With pairings
Goals	1. Generic compiler to obtain DDFE from DMCFE 2. DDFE for various concrete functionalities: IP, AWS, with/without function hiding, with/without access control	1. Standard model for DDFE, including FH-IP and AWS functionalities 2. Adaptive security for FH-IP DDFE from any FH-IP FE
Framework	From Decentralized MCFE with structural properties	From single-user FH-IPFE and AWS/IP-FE in pairing groups
Challenges	Adaptive security with repetitions on both message and key tags, with potentially illegitimate keys in case of access control	Zero-sharing for standard model, and adaptive security (also with repetitions) without complexity leveraging from any FH-IP FE

Conclusion

	DDFE: Generic Constructions with Strong Security	DDFE from Pairings in the Standard Model
Settings	1. Standard assumptions 2. Possibly in ROM 3. With/without pairings	1. Standard assumptions 2. Standard model 3. With pairings
Goals	1. Generic compiler to obtain DDFE from DMCFE 2. DDFE for various concrete functionalities: IP, AWS, with/without function hiding, with/without access control	1. Standard model for DDFE, including FH-IP and AWS functionalities 2. Adaptive security for FH-IP DDFE from any FH-IP FE
Framework	From Decentralized MCFE with structural properties	From single-user FH-IPFE and AWS/IP-FE in pairing groups
Challenges	Adaptive security with repetitions on both message and key tags, with potentially illegitimate keys in case of access control	Zero-sharing for standard model, and adaptive security (also with repetitions) without complexity leveraging from any FH-IP FE

Thank you!