Multi-Key Homomorphic Secret Sharing

TPMPC 2025

Geoffroy Couteau

CNRS, IRIF
Universitè Paris Citè

Lalita Devadas

MIT

Aditya Hegde

JHU

Sacha
Servan-Schreiber

MIT

Abhishek Jain

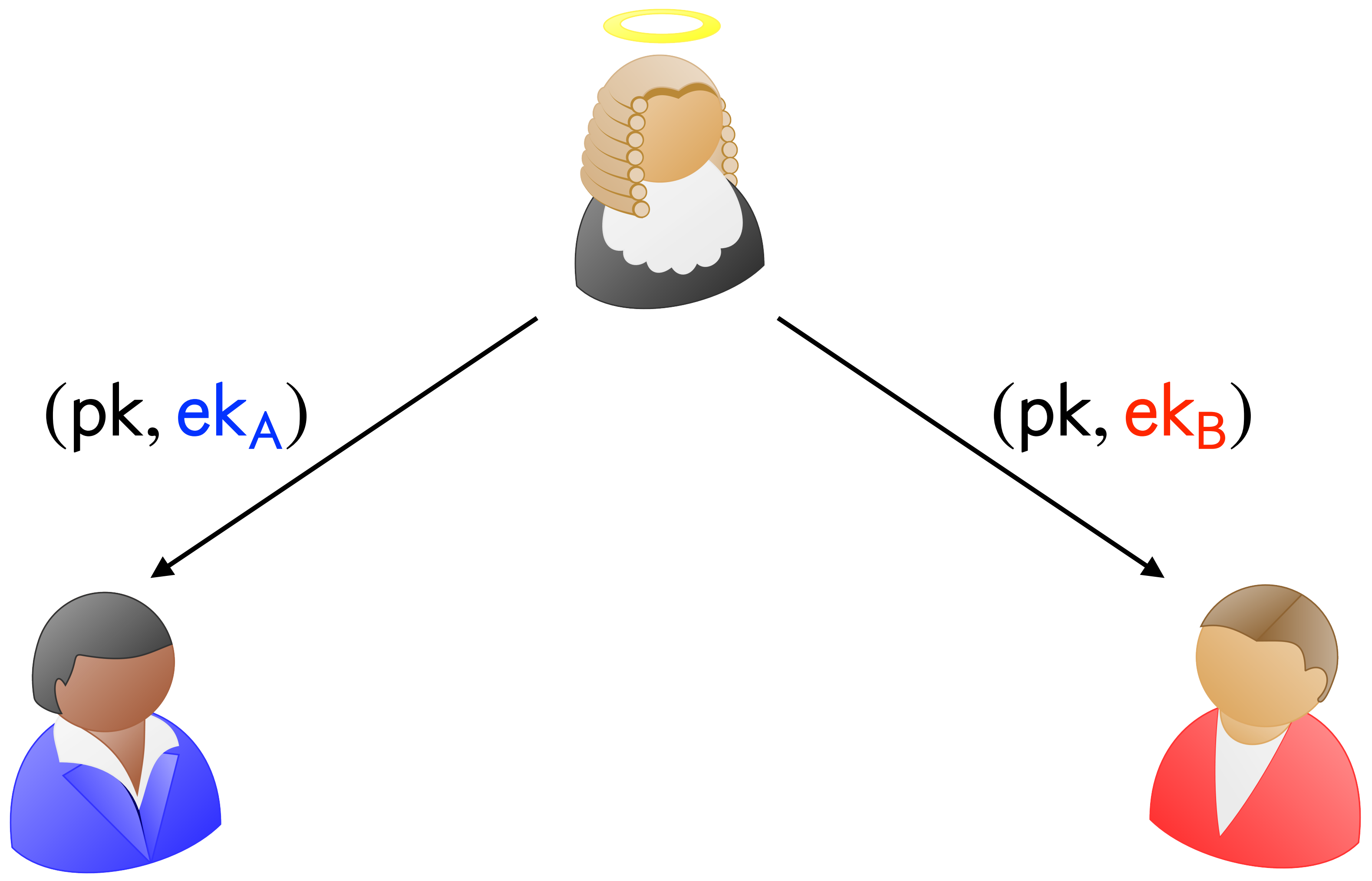
NTT Research
JHU

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai’16]

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai'16]

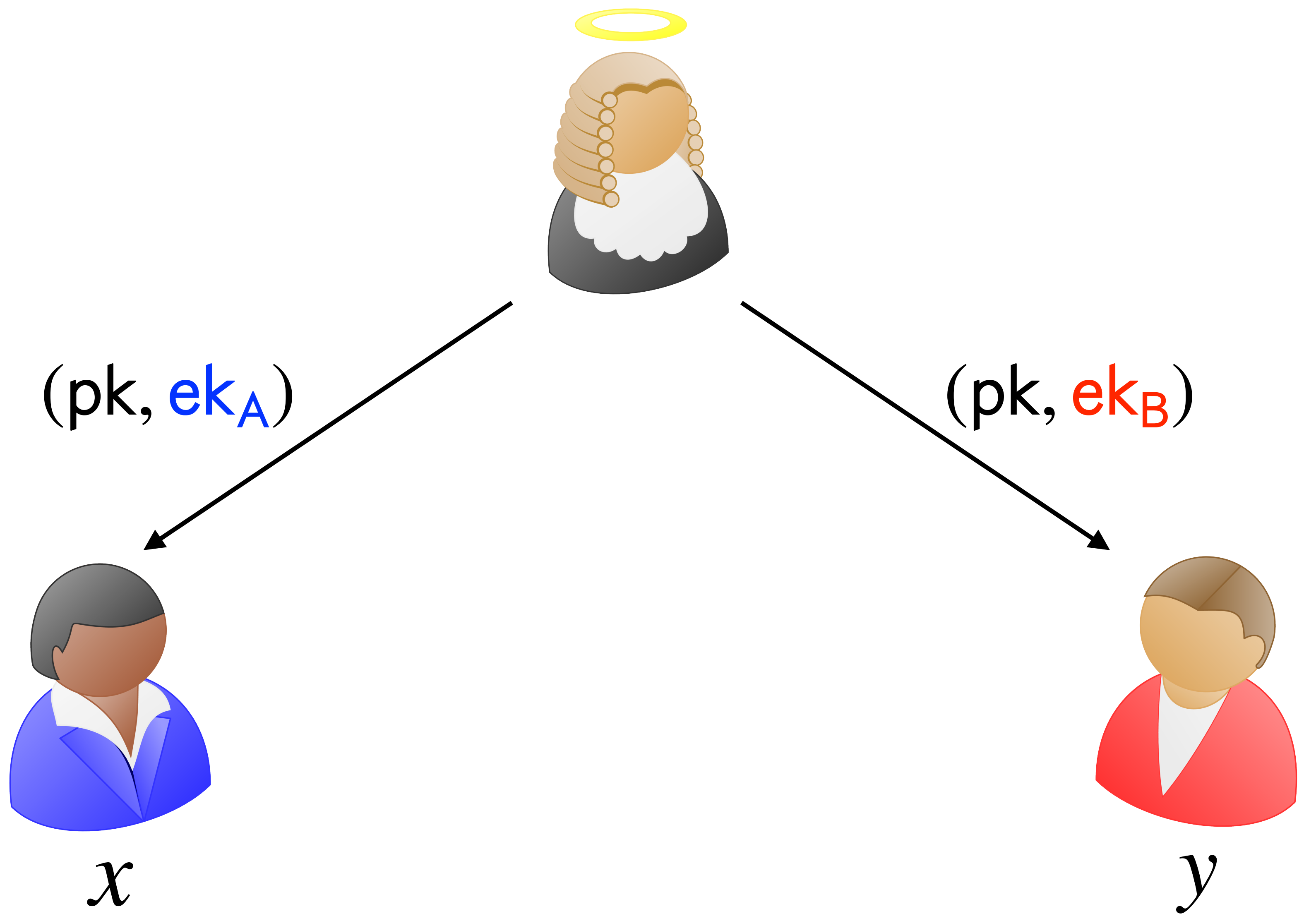


$(\mathrm{pk}, \mathrm{ek}_A)$

$(\mathrm{pk}, \mathrm{ek}_B)$

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai'16]



$(\mathsf{pk}, \mathsf{ek}_A)$

$(\mathsf{pk}, \mathsf{ek}_B)$

x

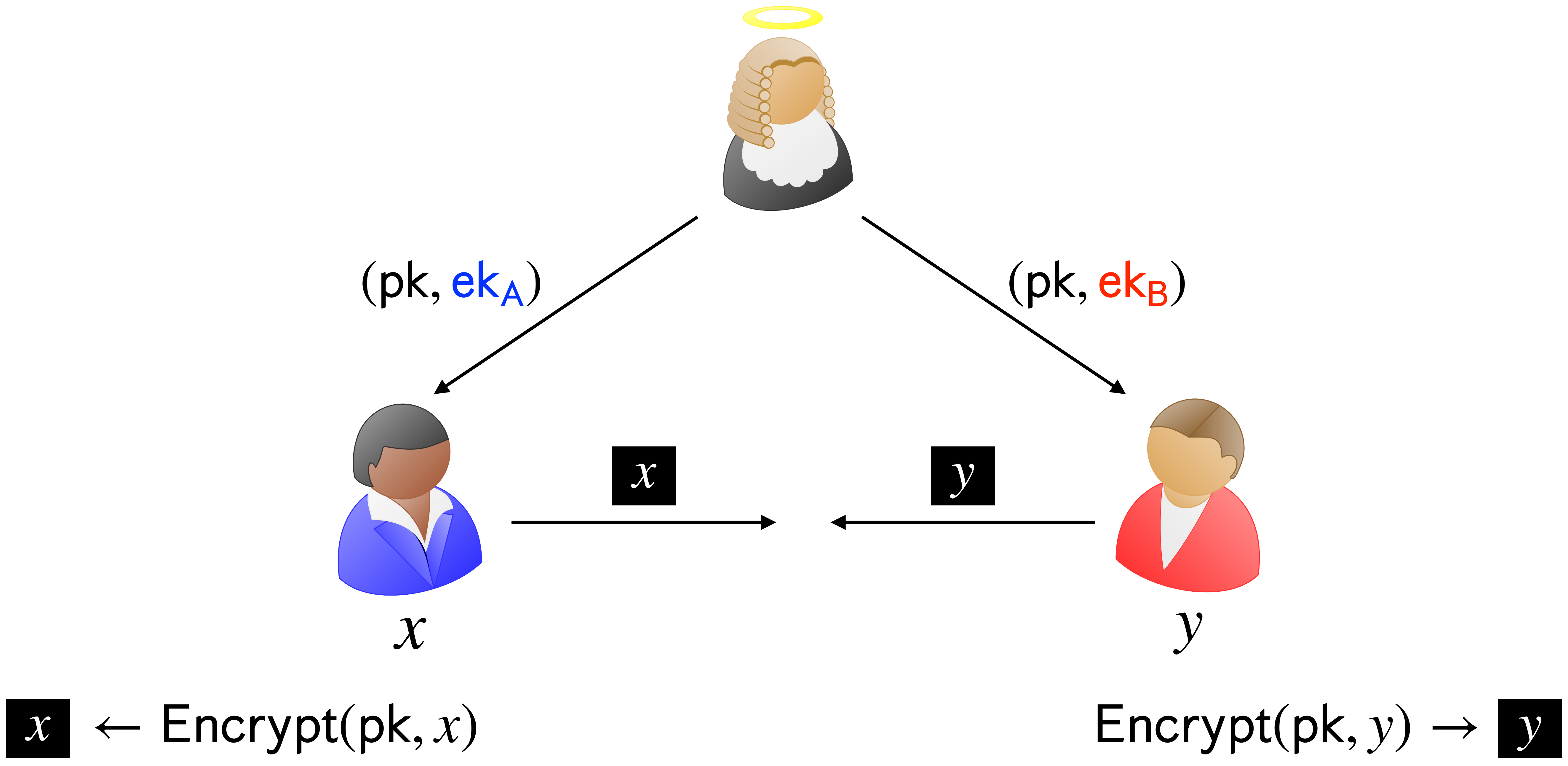
y

$\boxed{x} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)$

$\mathsf{Encrypt}(\mathsf{pk}, y) \rightarrow \boxed{y}$

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai’16]



(pk, ek_A)

(pk, ek_B)

$\boxed{x}$

$\boxed{y}$

x

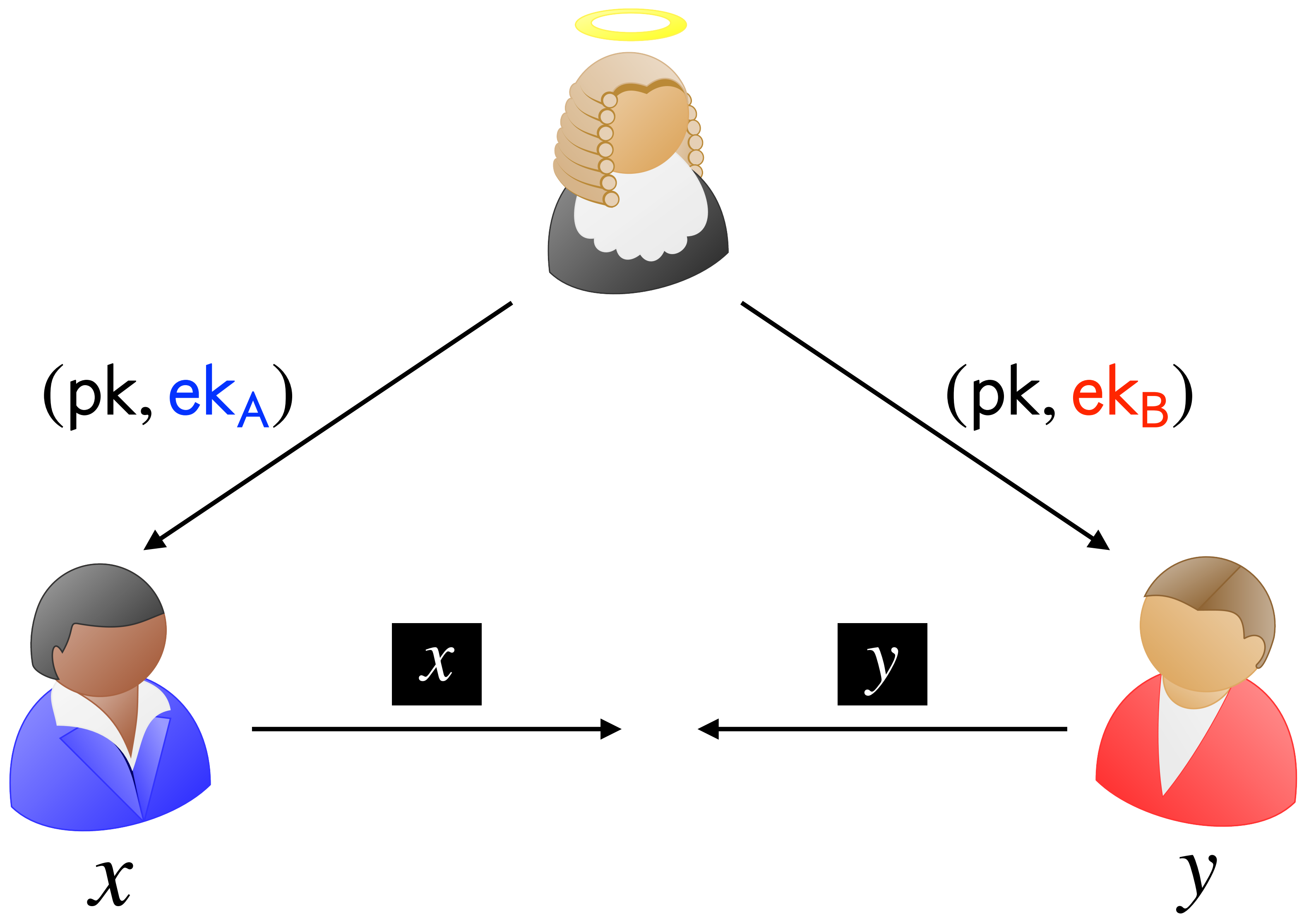
y

$\boxed{x} \leftarrow \text{Encrypt}(\text{pk}, x)$

$\text{Encrypt}(\text{pk}, y) \rightarrow \boxed{y}$

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai'16]



$(\mathsf{pk}, \mathsf{ek}_A)$

$(\mathsf{pk}, \mathsf{ek}_B)$

$\boxed{x}$

$\boxed{y}$

x

y

$\boxed{x} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)$

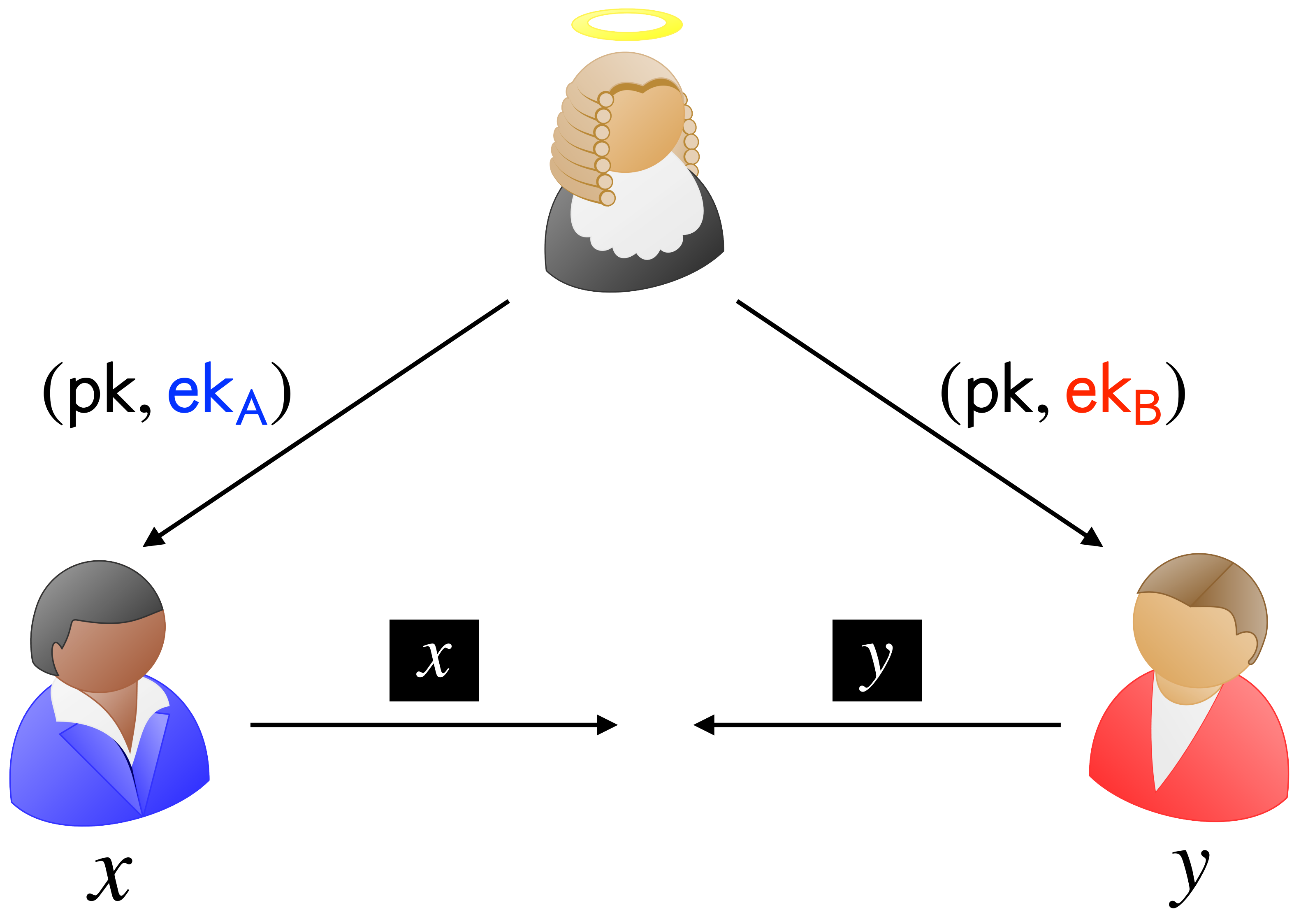
$\mathsf{Encrypt}(\mathsf{pk}, y) \rightarrow \boxed{y}$

$z_A \leftarrow \mathsf{Eval}(\mathsf{ek}_A\ , C\ , \boxed{x}\ , \boxed{y}\ )$

$\mathsf{Eval}(\mathsf{ek}_B\ , C\ , \boxed{x}\ , \boxed{y}\ ) \rightarrow z_B$

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai'16]



(pk, ek_A)

(pk, ek_B)

$\boxed{x}$ $\boxed{y}$

x y

$\boxed{x} \leftarrow \text{Encrypt}(\text{pk}, x)$

$\text{Encrypt}(\text{pk}, y) \rightarrow \boxed{y}$

$z_A \leftarrow \text{Eval}(\text{ek}_A, C, \boxed{x}, \boxed{y})$

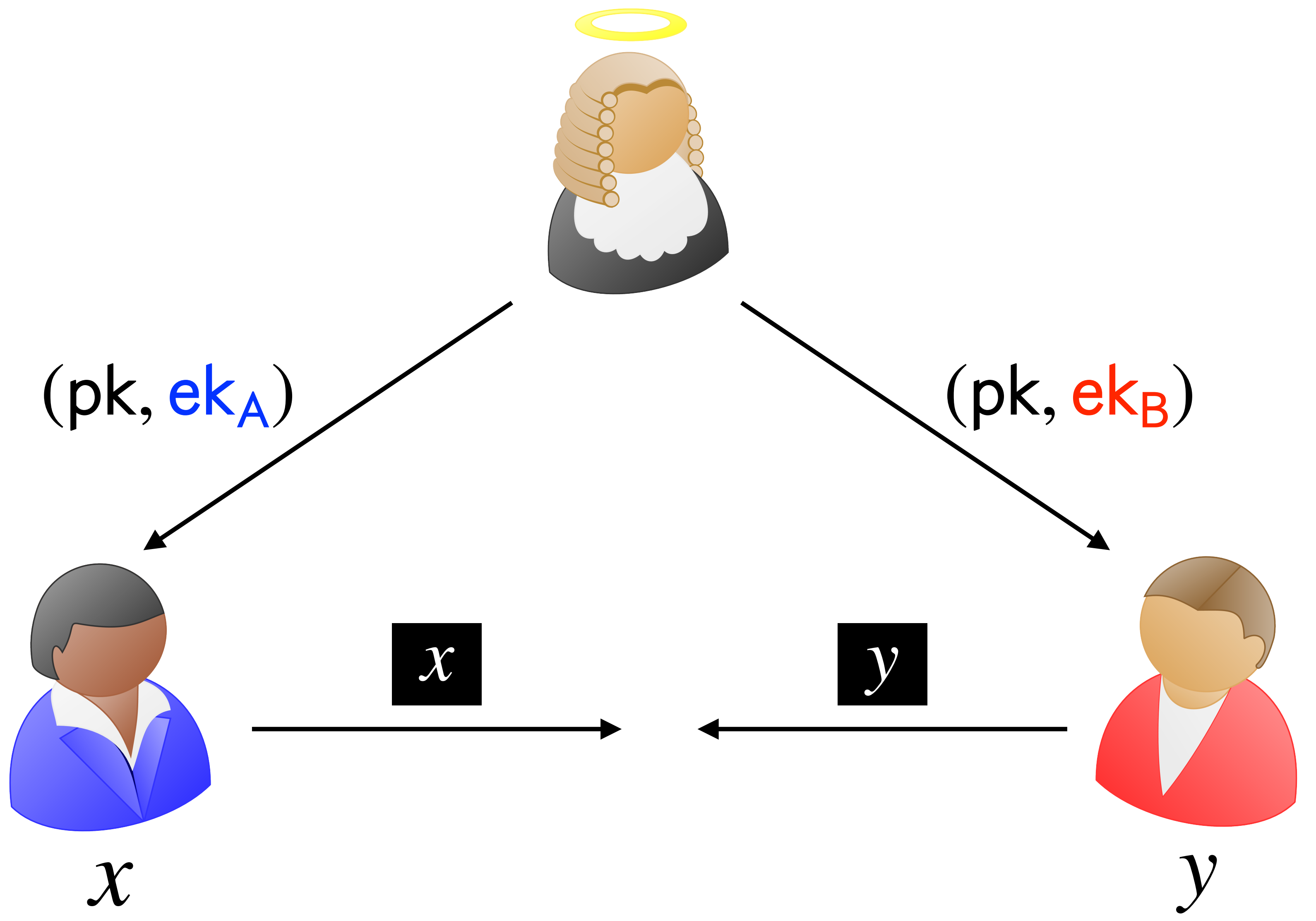
$\text{Eval}(\text{ek}_B, C, \boxed{x}, \boxed{y}) \rightarrow z_B$

Correctness

$$z_A + z_B = C(x, y)$$

Homomorphic Secret Sharing

[Boyle-Gilboa-Ishai’16]



$(\text{pk}, \text{ek}_{\text{A}})$

$(\text{pk}, \text{ek}_{\text{B}})$

$\boxed{x}$ $\boxed{y}$

x y

$\boxed{x} \leftarrow \text{Encrypt}(\text{pk}, x)$

$\text{Encrypt}(\text{pk}, y) \rightarrow \boxed{y}$

$z_{\text{A}} \leftarrow \text{Eval}(\text{ek}_{\text{A}}, C, \boxed{x}, \boxed{y})$

$\text{Eval}(\text{ek}_{\text{B}}, C, \boxed{x}, \boxed{y}) \rightarrow z_{\text{B}}$

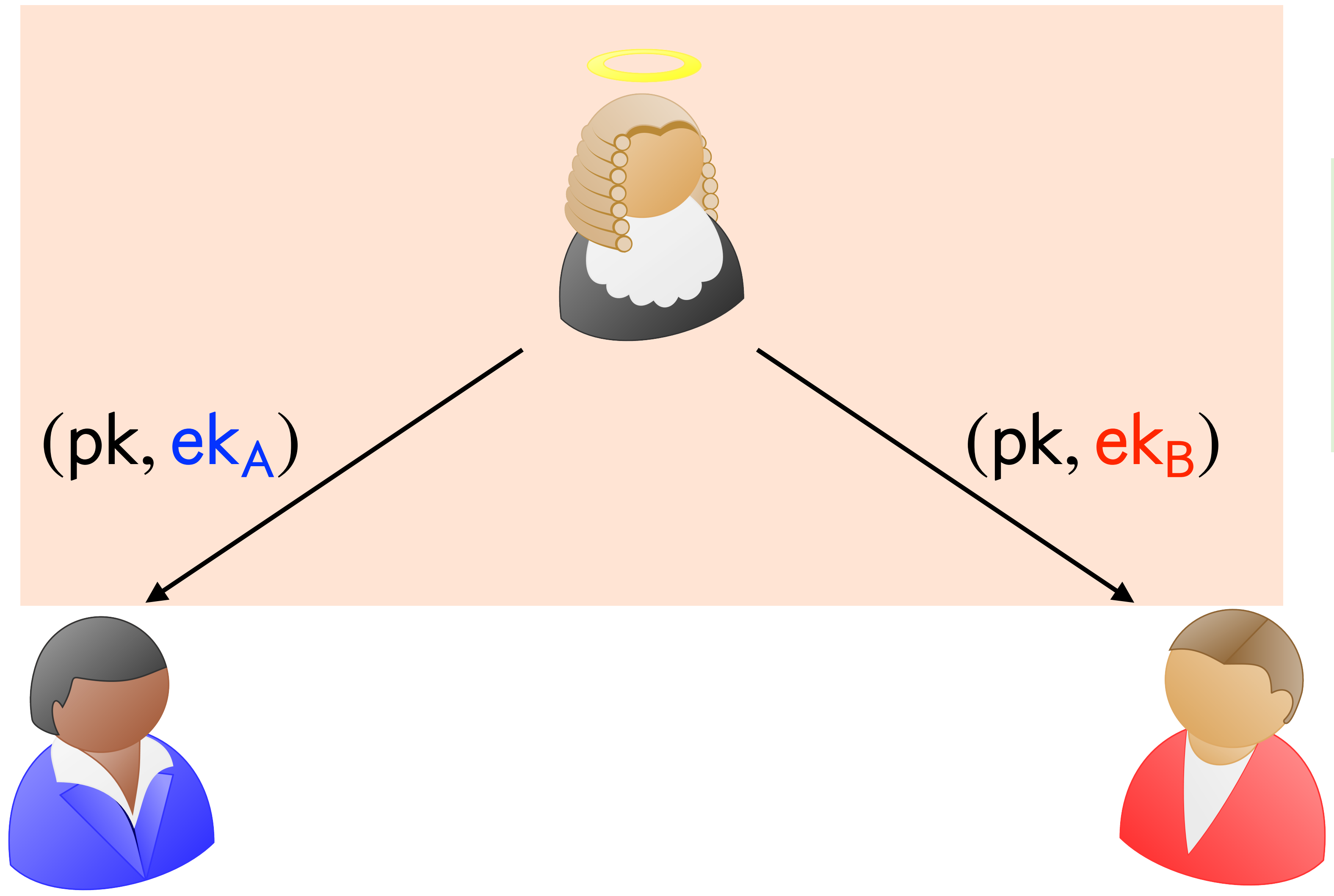
Correctness $\quad z_{\text{A}} + z_{\text{B}} = C(x, y)$

Security

$\boxed{x}$ ensures privacy of x

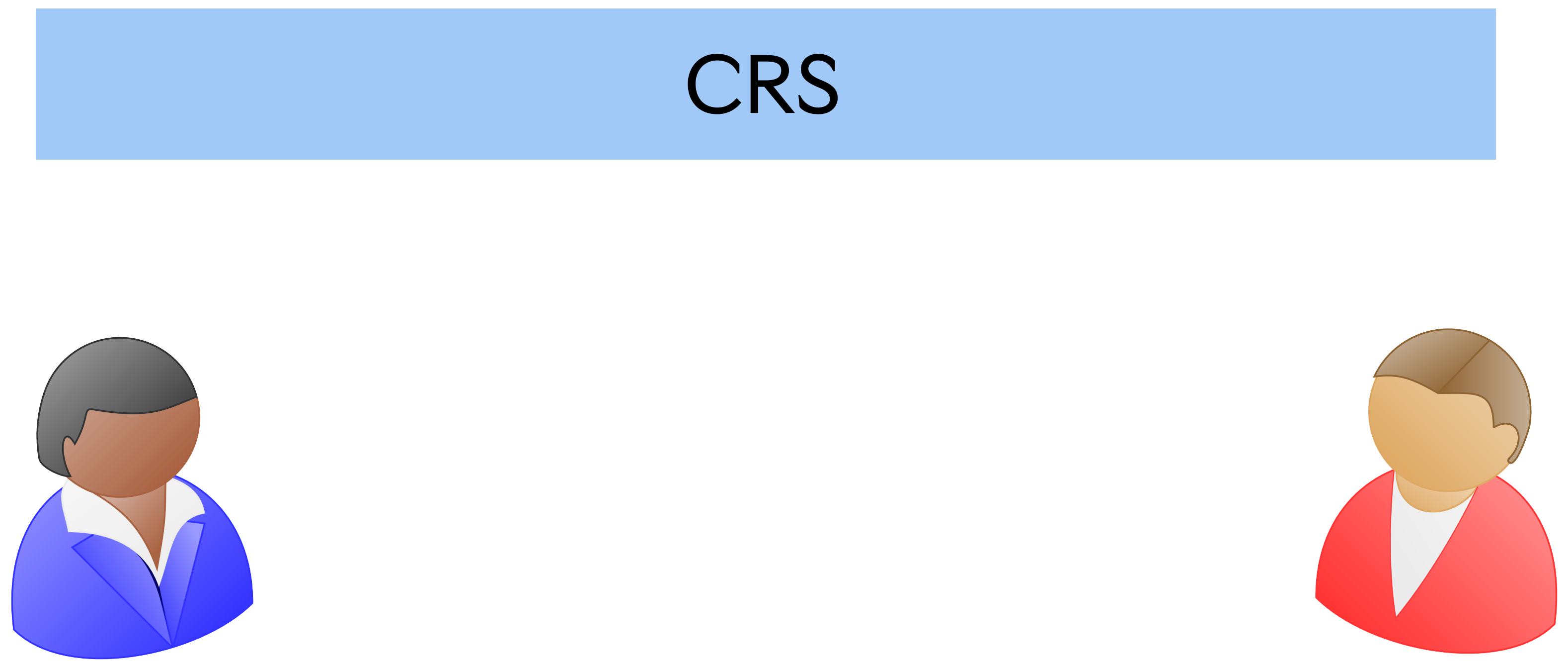
$\boxed{y}$ ensures privacy of y

Multi-Key Homomorphic Secret Sharing


$(\mathrm{pk}, \mathrm{ek}_A)$
$(\mathrm{pk}, \mathrm{ek}_B)$

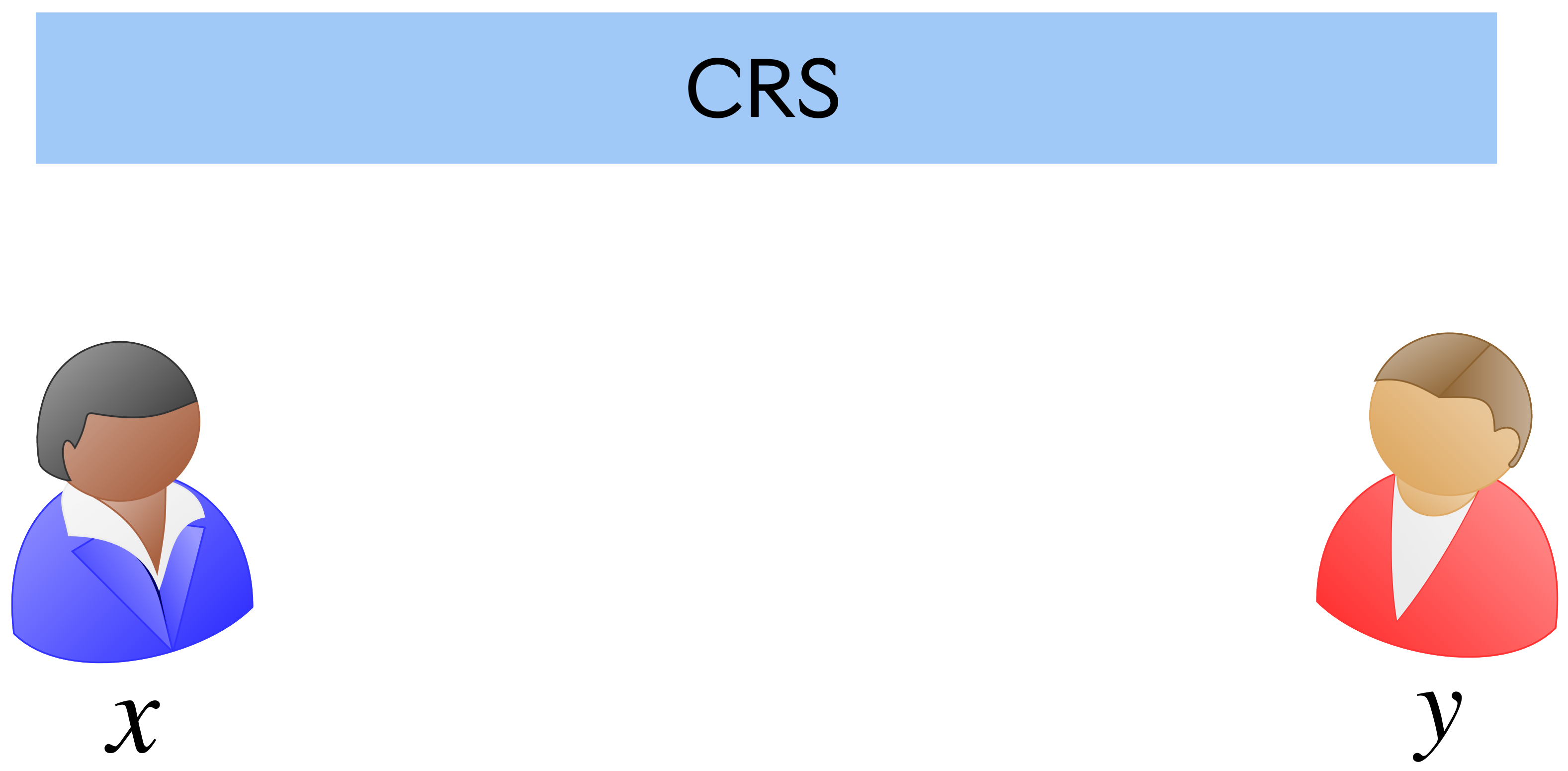
Replace correlated setup with CRS

Multi-Key Homomorphic Secret Sharing



CRS

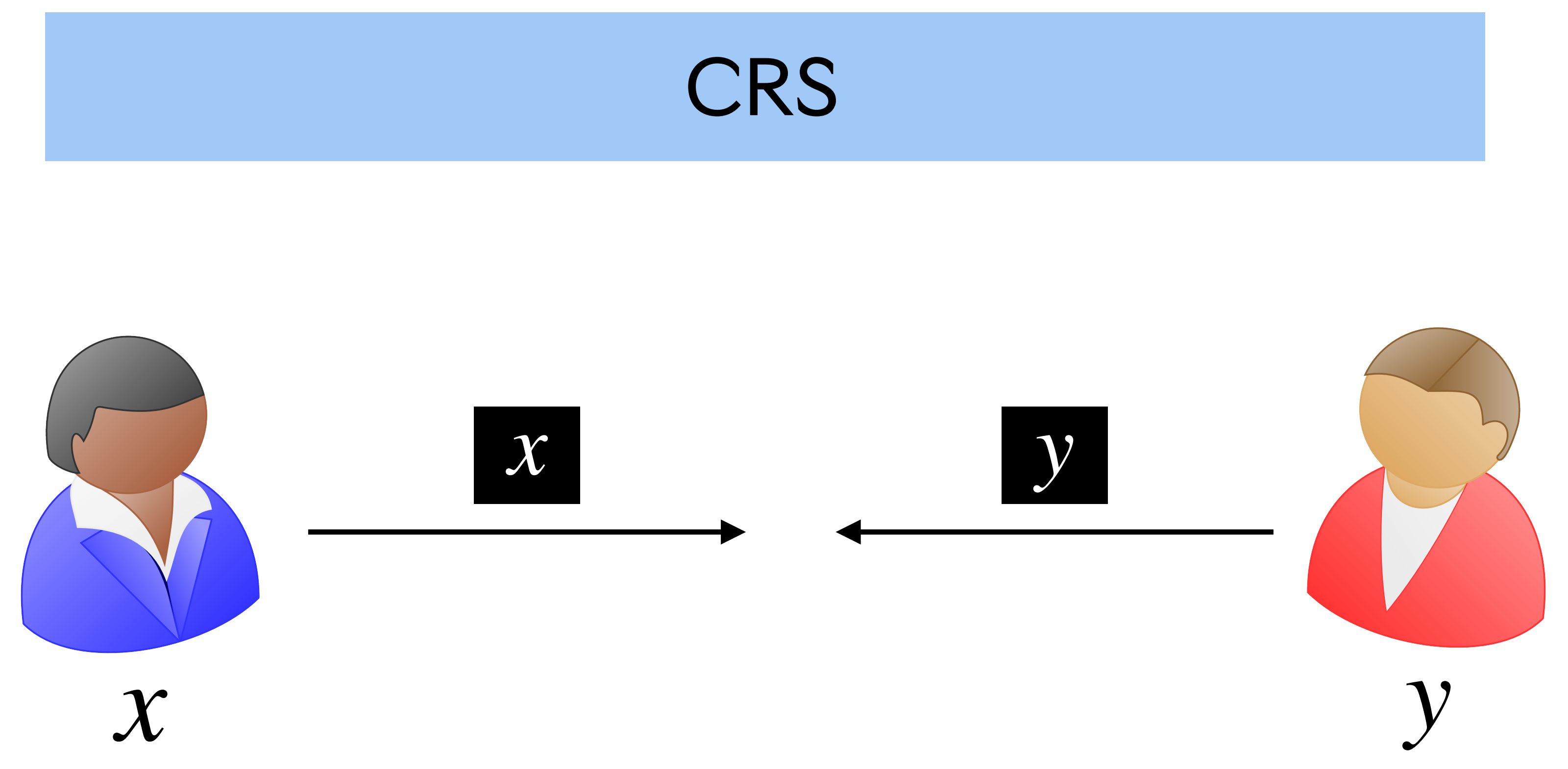
Multi-Key Homomorphic Secret Sharing



$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

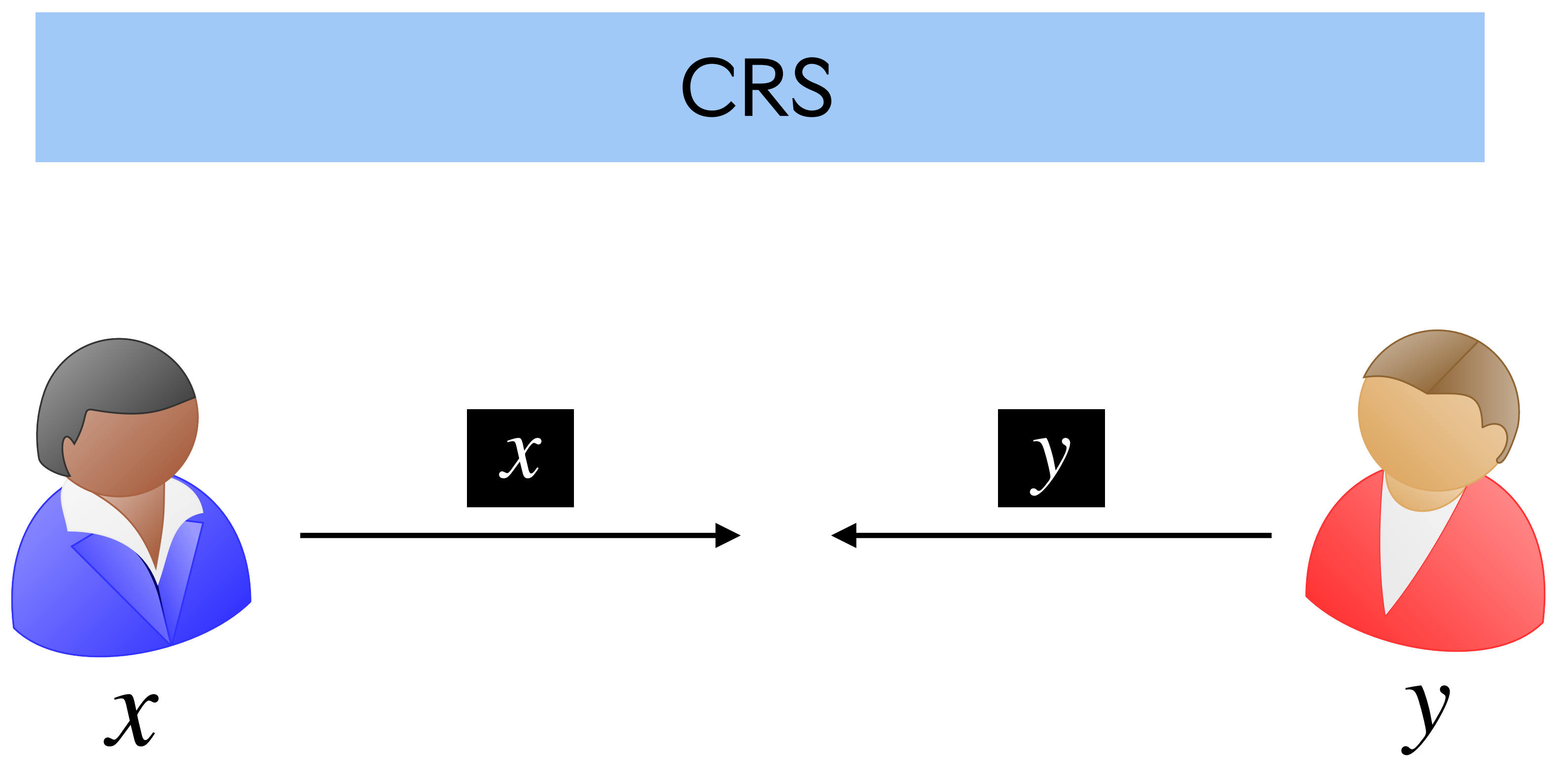
Multi-Key Homomorphic Secret Sharing



$(\boxed{x} , \mathrm{st}_A) \leftarrow \mathrm{Encode}(x)$

$\mathrm{Encode}(y) \rightarrow (\boxed{y} , \mathrm{st}_B)$

Multi-Key Homomorphic Secret Sharing



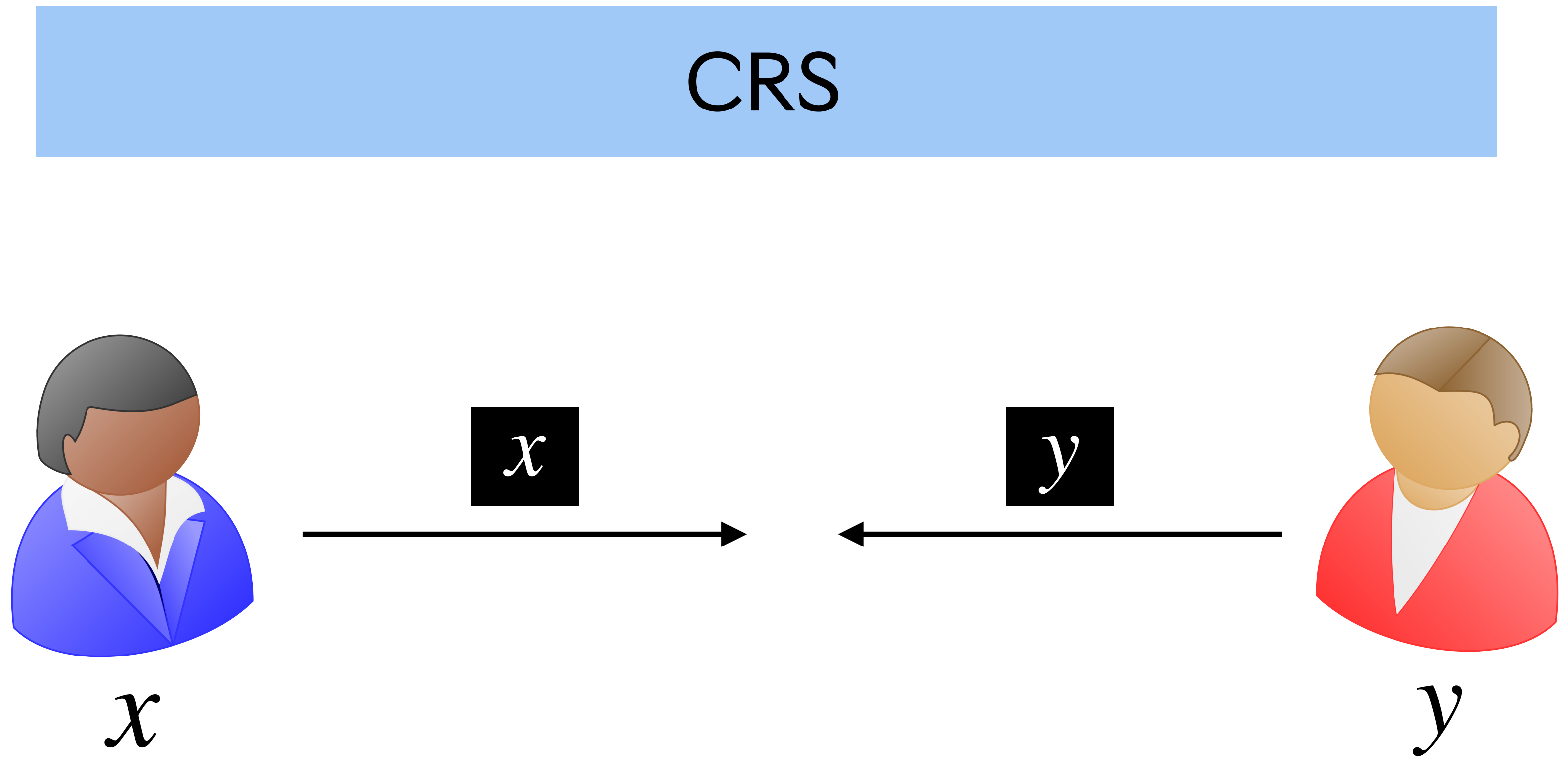
$(\ \boxed{x}\ , \mathsf{st}_{\mathsf{A}}\ ) \leftarrow \mathsf{Encode}(\ x\ )$

$\mathsf{Encode}(\ y\ ) \rightarrow (\ \boxed{y}\ , \mathsf{st}_{\mathsf{B}}\ )$

$z_{\mathsf{A}} \leftarrow \mathsf{Eval}(\ C\ , \mathsf{st}_{\mathsf{A}}\ , \boxed{y}\ )$

$\mathsf{Eval}(\ C\ , \mathsf{st}_{\mathsf{B}}\ , \boxed{x}\ ) \rightarrow z_{\mathsf{B}}$

Multi-Key Homomorphic Secret Sharing



CRS

$\boxed{x}$

$\boxed{y}$

x

y

$(\boxed{x}, \mathsf{st}_\mathsf{A}) \leftarrow \mathsf{Encode}(x)$

$\boxed{z_\mathsf{A}} \leftarrow \mathsf{Eval}(C, \mathsf{st}_\mathsf{A}, \boxed{y})$

$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_\mathsf{B})$

$\mathsf{Eval}(C, \mathsf{st}_\mathsf{B}, \boxed{x}) \rightarrow \boxed{z_\mathsf{B}}$

Correctness

$$\boxed{z_\mathsf{A}} + \boxed{z_\mathsf{B}} = C(x, y)$$

Security

$\boxed{x}$ ensures privacy of x

$\boxed{y}$ ensures privacy of y

Outline

Applications

Our Results

Constructing Multi-Key HSS

Outline

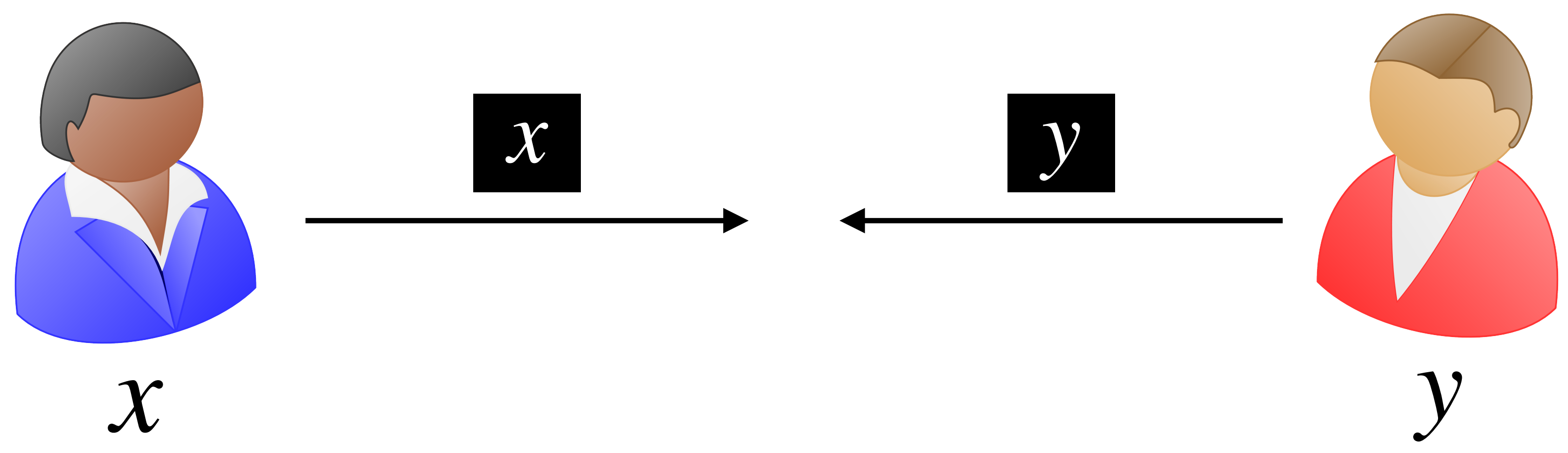
Applications

Our Results

Constructing Multi-Key HSS

Key Properties of Multi-Key HSS

CRS



$(\ \boxed{x}\ ,\ \mathsf{st}_A\ ) \leftarrow$ Encode$(\ x\ )$

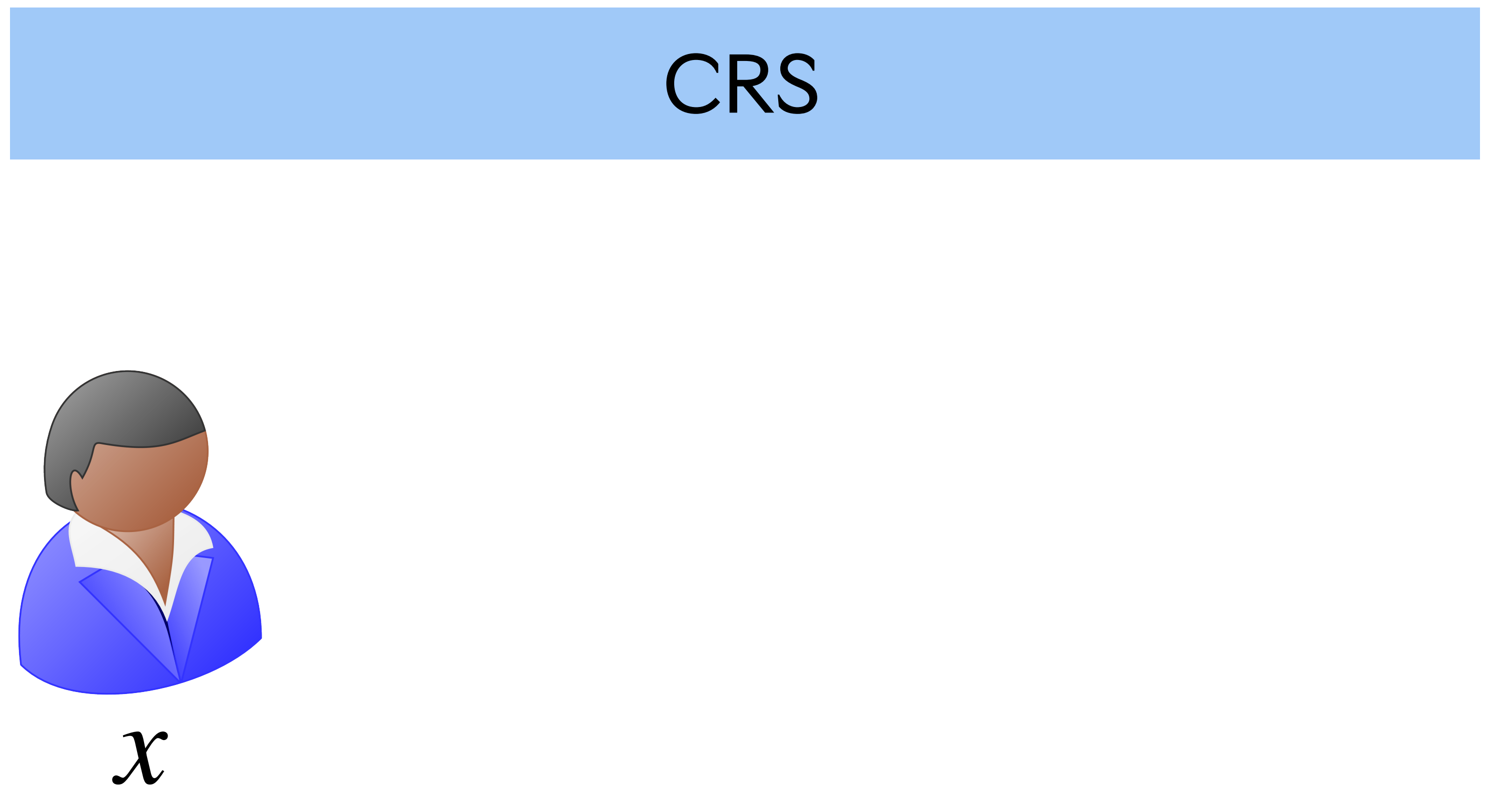
$z_A \leftarrow$ Eval$(\ C\ ,\ \mathsf{st}_A\ ,\ \boxed{y}\ )$

Encode$(\ y\ ) \rightarrow (\ \boxed{y}\ ,\ \mathsf{st}_B\ )$

Eval$(\ C\ ,\ \mathsf{st}_B\ ,\ \boxed{x}\ ) \rightarrow z_B$

Reduces round complexity by avoiding correlated setup

Key Properties of Multi-Key HSS



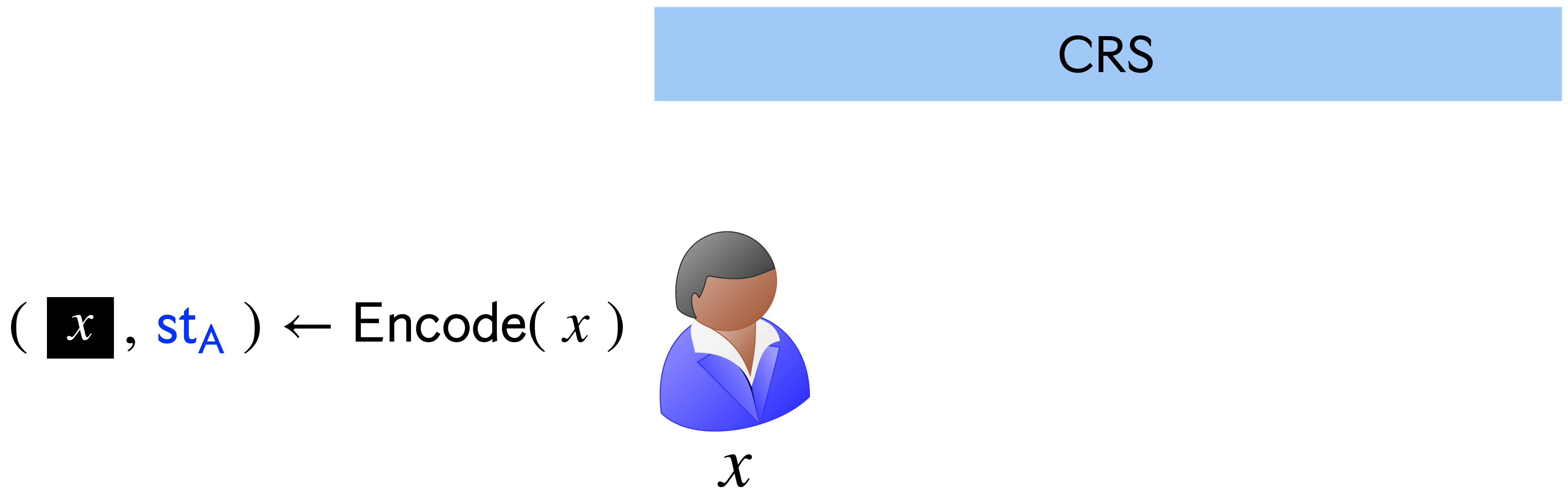
CRS

x

Reduces round complexity by avoiding correlated setup

Reusability of input encodings

Key Properties of Multi-Key HSS



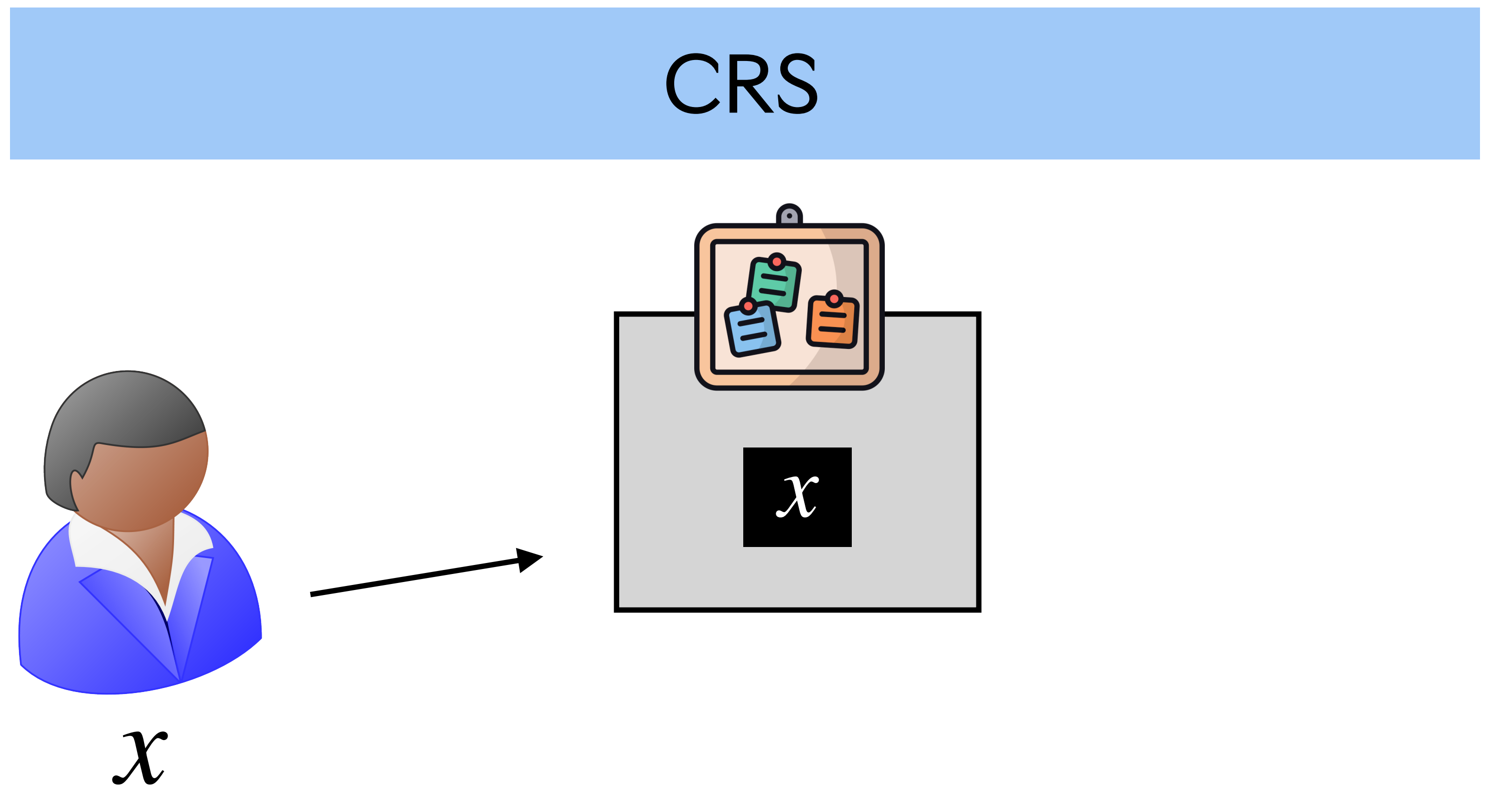
$(\boxed{x}, \mathrm{st}_A) \leftarrow \mathrm{Encode}(x)$

Reduces round complexity by avoiding correlated setup

Reusability of input encodings

Key Properties of Multi-Key HSS

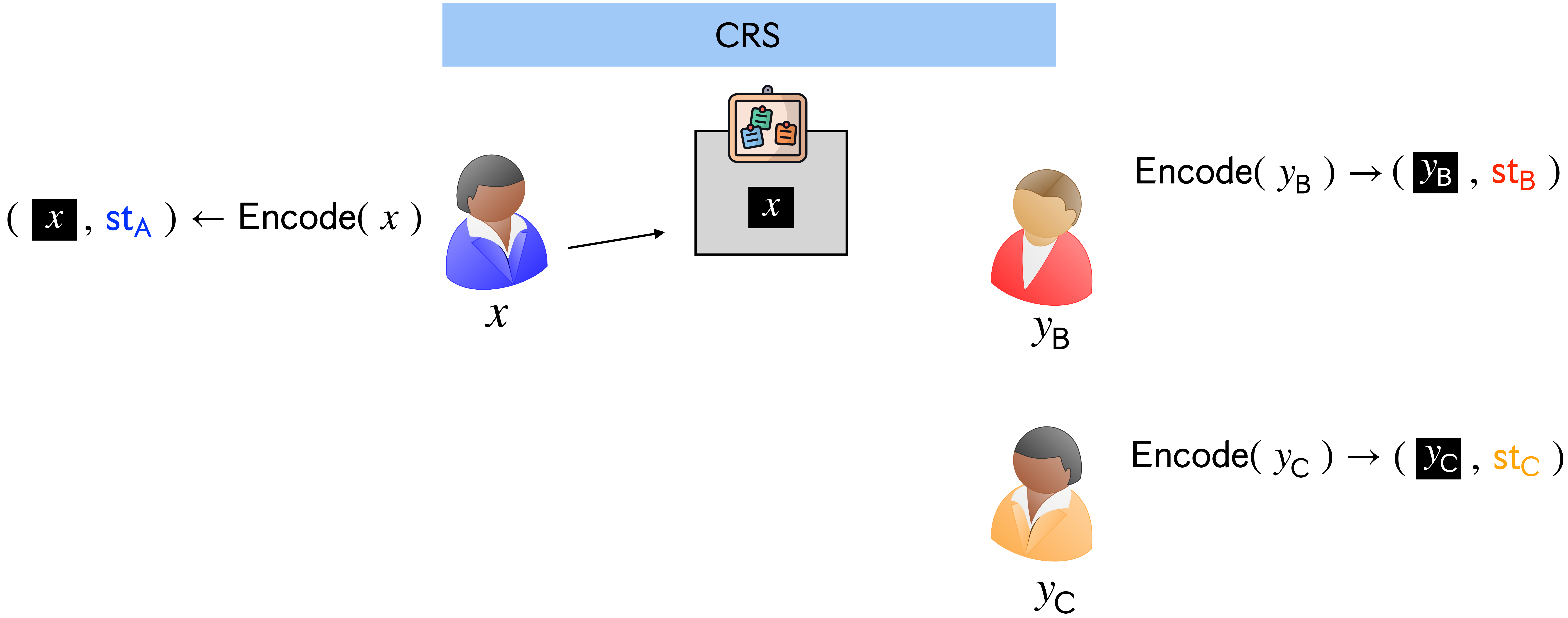
$(\boxed{x} , \mathrm{st}_A) \leftarrow \mathrm{Encode}(x)$



Reduces round complexity by avoiding correlated setup

Reusability of input encodings

Key Properties of Multi-Key HSS



CRS

$(\boxed{x} , \mathrm{st}_A) \leftarrow$ Encode(x)

$\boxed{x}$

x

Encode(y_B) $\rightarrow (\boxed{y_B} , \mathrm{st}_B)$

y_B

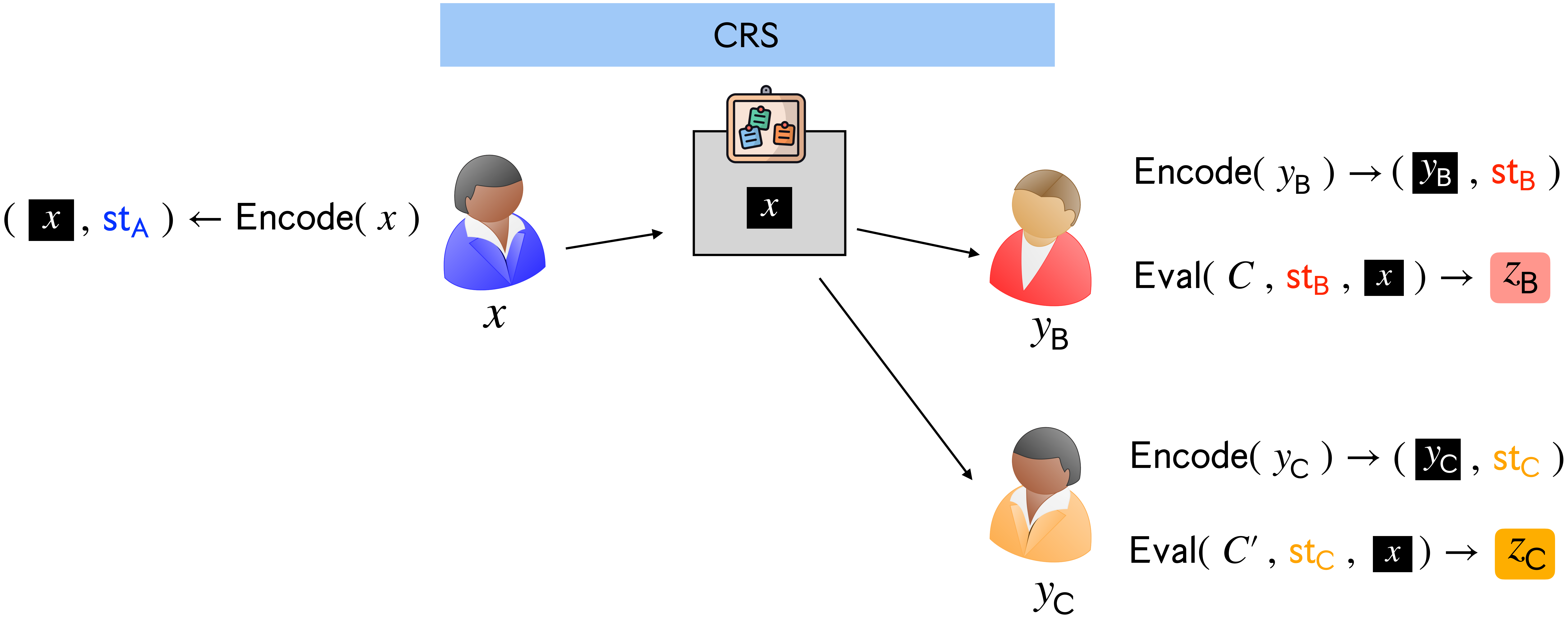
Encode(y_C) $\rightarrow (\boxed{y_C} , \mathrm{st}_C)$

y_C

Reduces round complexity by avoiding correlated setup

Reusability of input encodings

Key Properties of Multi-Key HSS



$(\boxed{x}, \mathsf{st}_A) \leftarrow \text{Encode}(x)$

$\text{Encode}(y_B) \rightarrow (\boxed{y_B}, \mathsf{st}_B)$

$\text{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow z_B$

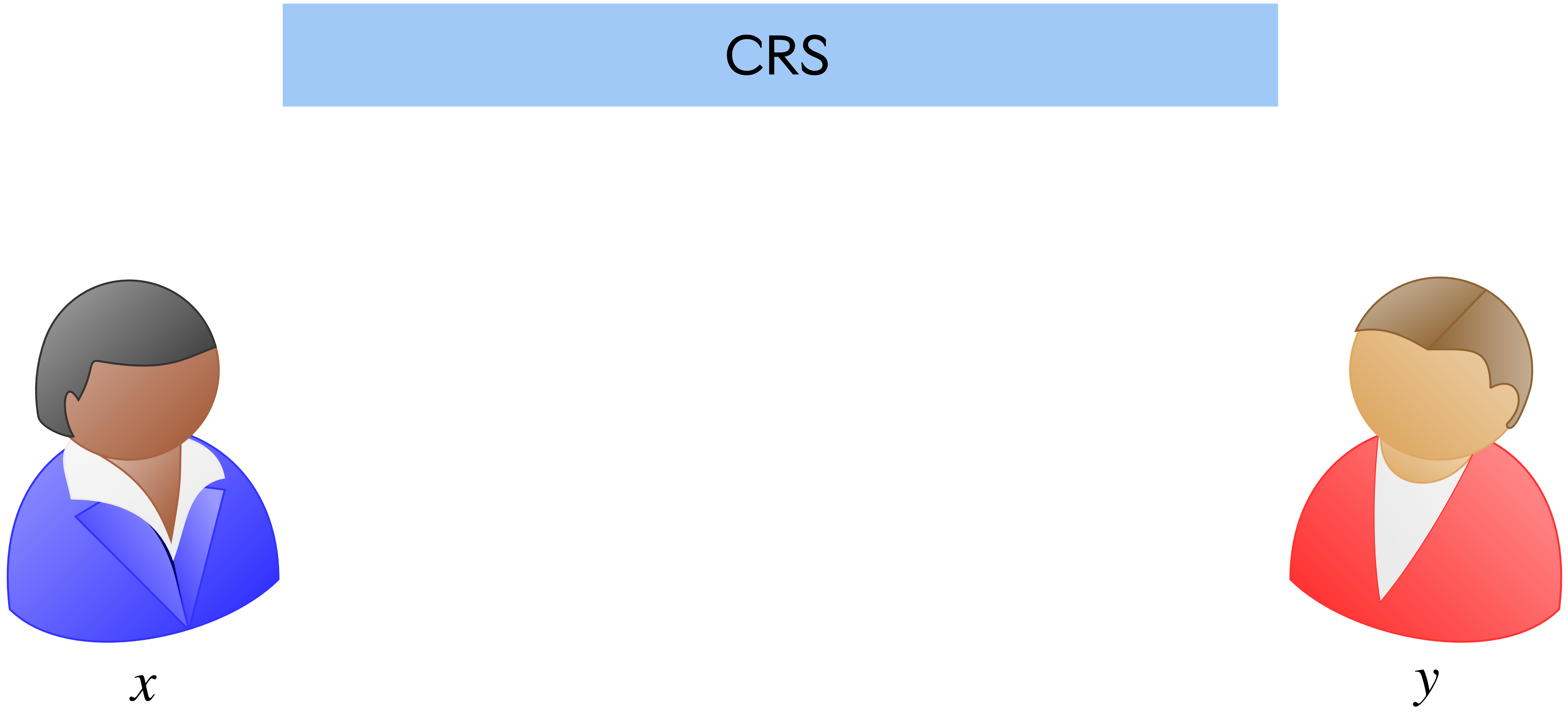
$\text{Encode}(y_C) \rightarrow (\boxed{y_C}, \mathsf{st}_C)$

$\text{Eval}(C', \mathsf{st}_C, \boxed{x}) \rightarrow z_C$

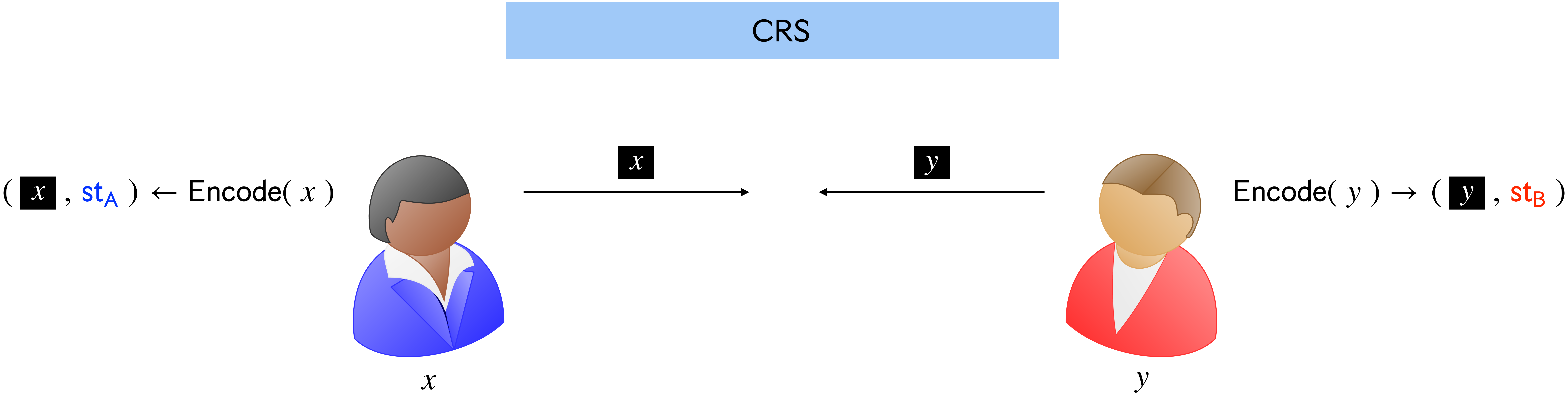
Reduces round complexity by avoiding correlated setup

Reusability of input encodings

Application 1: Two-Round Sublinear Secure Computation



Application 1: Two-Round Sublinear Secure Computation



CRS

$(\boxed{x}, \mathrm{st}_A) \leftarrow \mathrm{Encode}(x)$

x

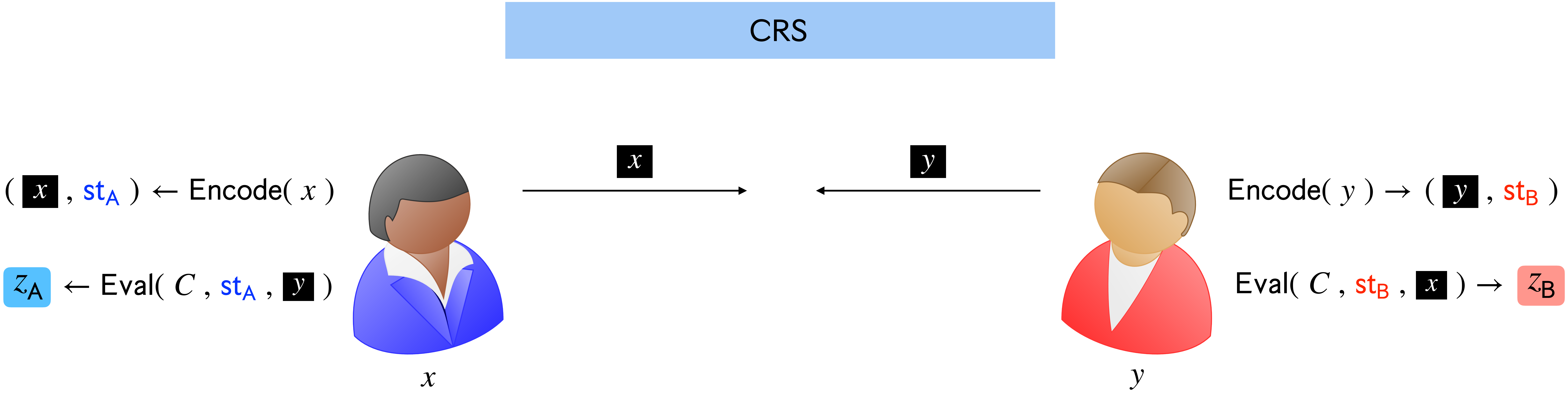
$\boxed{x}$

$\boxed{y}$

y

$\mathrm{Encode}(y) \rightarrow (\boxed{y}, \mathrm{st}_B)$

Application 1: Two-Round Sublinear Secure Computation



CRS

$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$z_A \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

x

$\xrightarrow{\boxed{x}}$

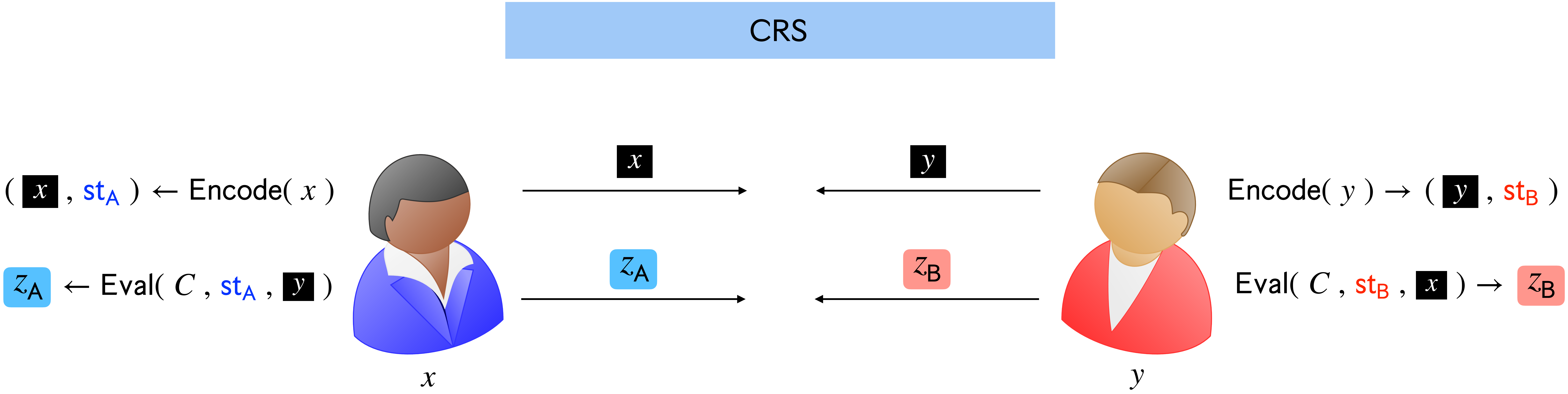
$\xleftarrow{\boxed{y}}$

$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow z_B$

y

Application 1: Two-Round Sublinear Secure Computation



CRS

$(\boxed{x} , \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$z_A \leftarrow \mathsf{Eval}(C , \mathsf{st}_A , \boxed{y})$

$\mathsf{Encode}(y) \rightarrow (\boxed{y} , \mathsf{st}_B)$

$\mathsf{Eval}(C , \mathsf{st}_B , \boxed{x}) \rightarrow z_B$

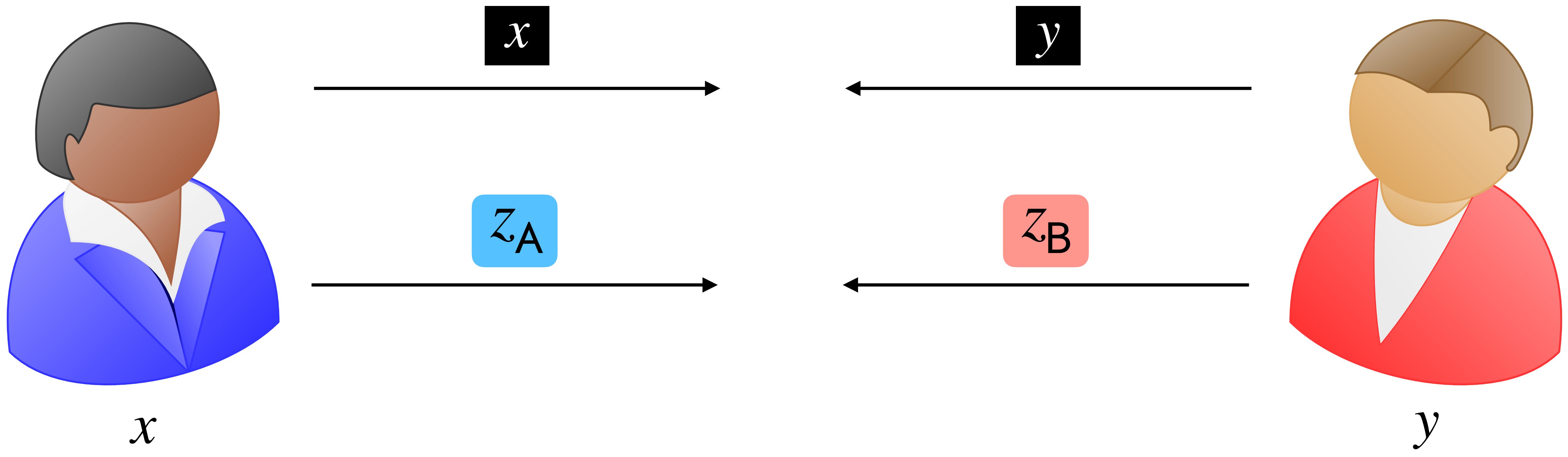
Application 1: Two-Round Sublinear Secure Computation

CRS

$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$\boxed{z_A} \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

$z = \boxed{z_A} + \boxed{z_B}$



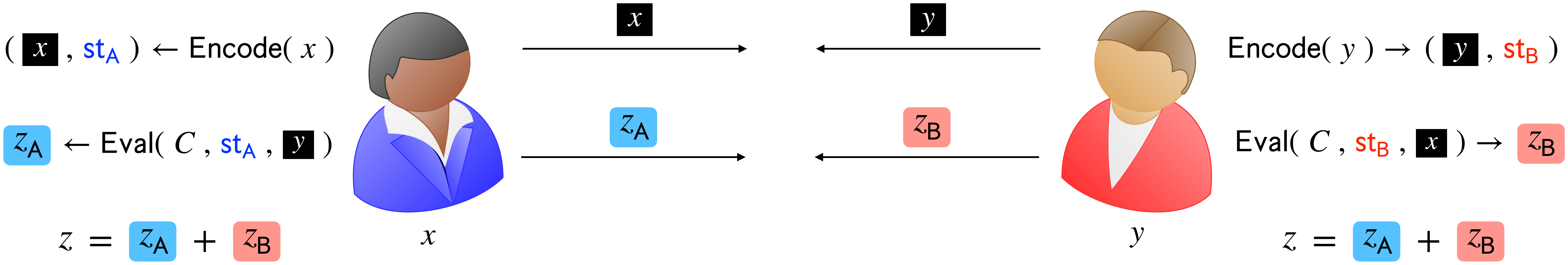
$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow \boxed{z_B}$

$z = \boxed{z_A} + \boxed{z_B}$

Application 1: Two-Round Sublinear Secure Computation

CRS



$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$z_A \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

$z = z_A + z_B$

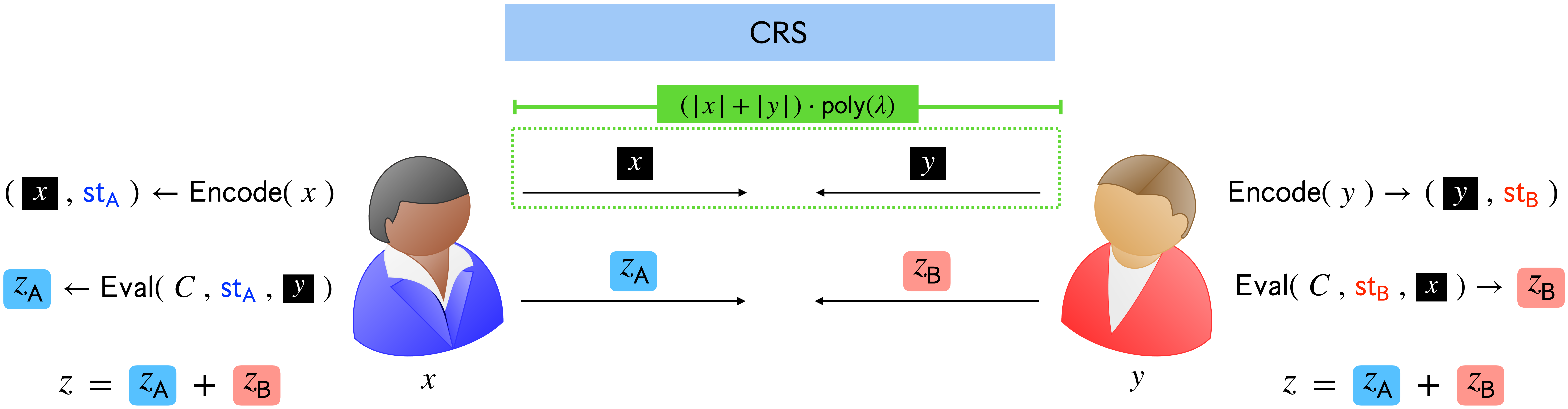
$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow z_B$

$z = z_A + z_B$

Sublinear communication in size of the circuit

Application 1: Two-Round Sublinear Secure Computation



CRS

$(|x| + |y|) \cdot \mathrm{poly}(\lambda)$

$\boxed{x}$ $\boxed{y}$

z_A z_B

$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$z_A \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

$z = z_A + z_B$

x

y

$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow z_B$

$z = z_A + z_B$

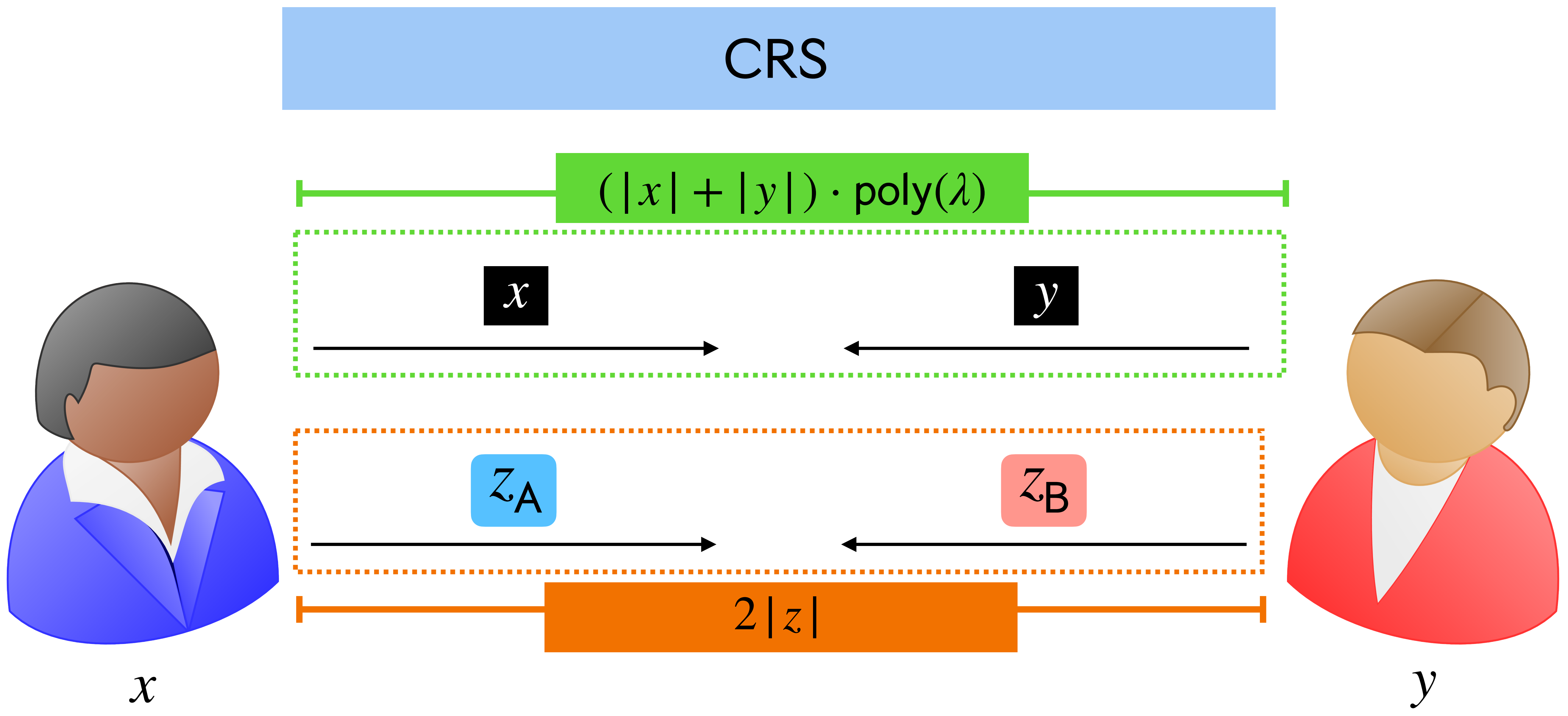
Sublinear communication in size of the circuit

Application 1: Two-Round Sublinear Secure Computation

$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$\boxed{z_A} \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

$z = \boxed{z_A} + \boxed{z_B}$



$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow \boxed{z_B}$

$z = \boxed{z_A} + \boxed{z_B}$

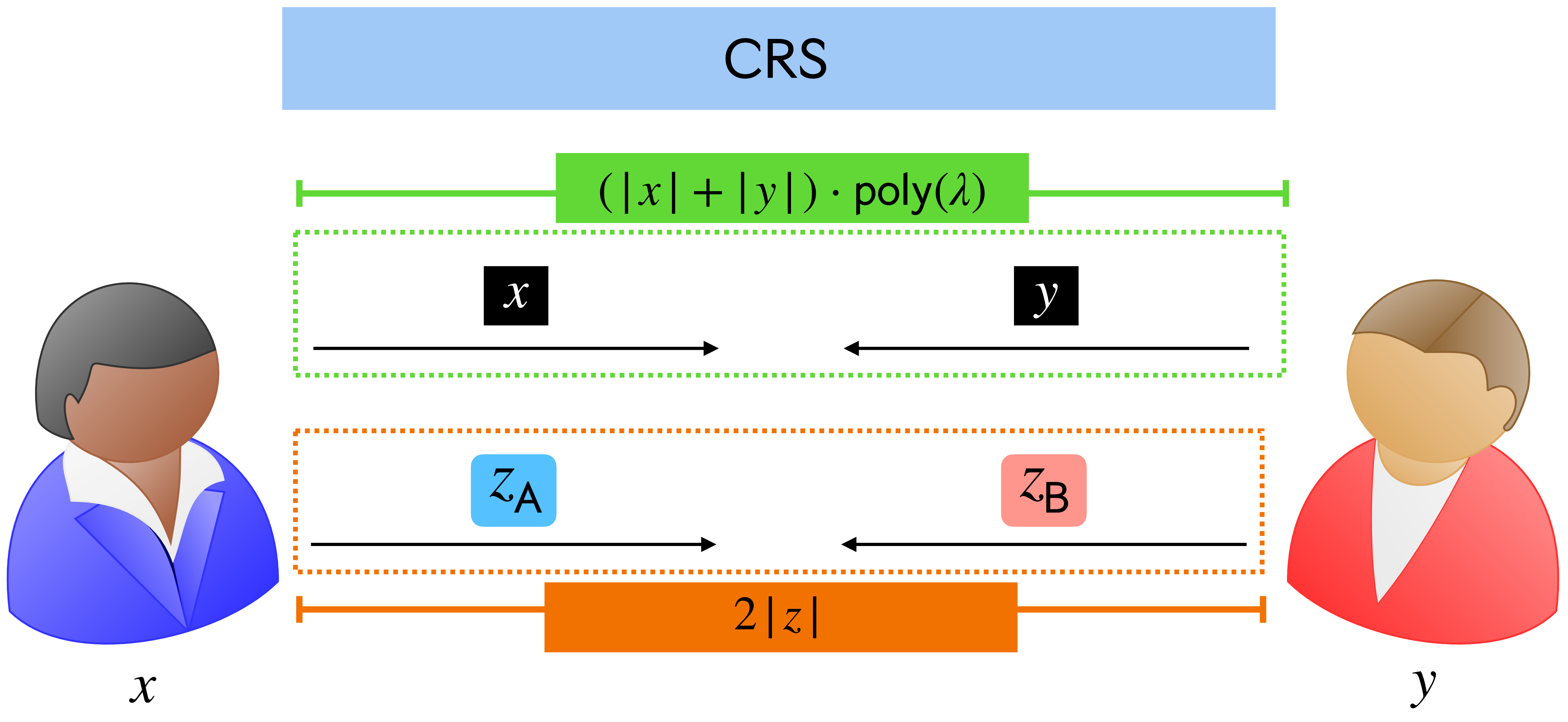
Sublinear communication in size of the circuit

Application 1: Two-Round Sublinear Secure Computation

$(\boxed{x}, \mathsf{st}_A) \leftarrow \mathsf{Encode}(x)$

$z_A \leftarrow \mathsf{Eval}(C, \mathsf{st}_A, \boxed{y})$

$z = z_A + z_B$



$\mathsf{Encode}(y) \rightarrow (\boxed{y}, \mathsf{st}_B)$

$\mathsf{Eval}(C, \mathsf{st}_B, \boxed{x}) \rightarrow z_B$

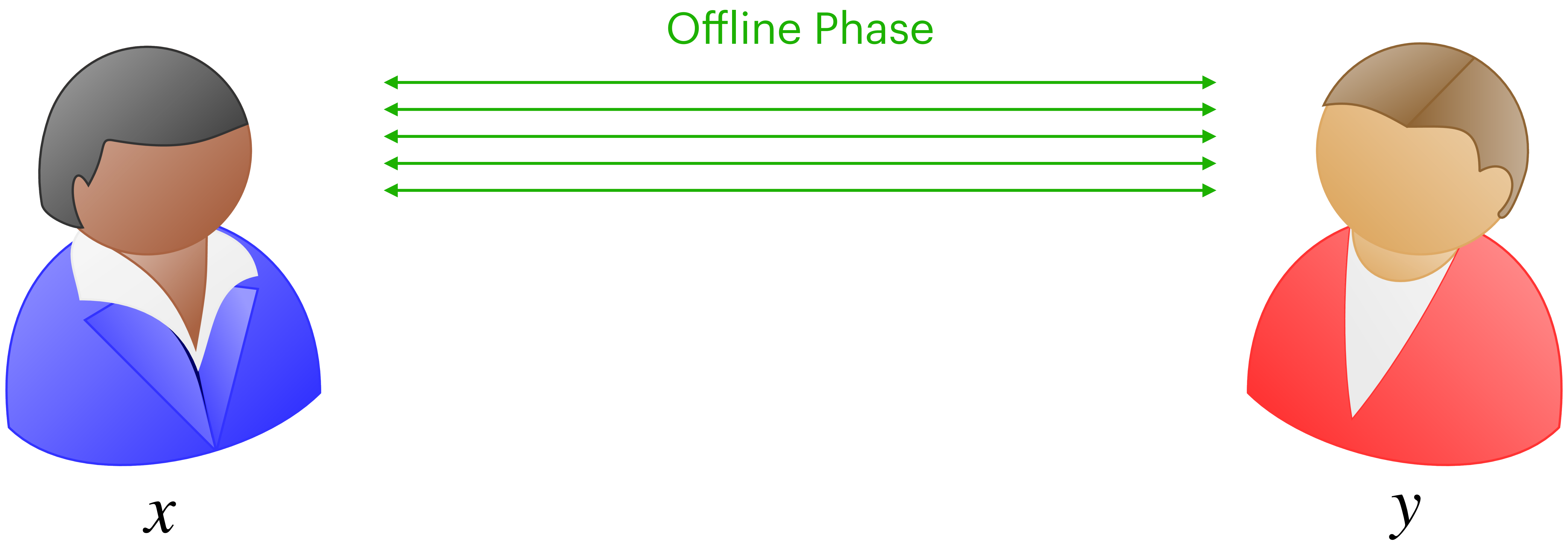
$z = z_A + z_B$

Sublinear communication in size of the circuit

Two-round protocol in the CRS model

Preprocessing Model for Secure Computation

Preprocessing Model for Secure Computation

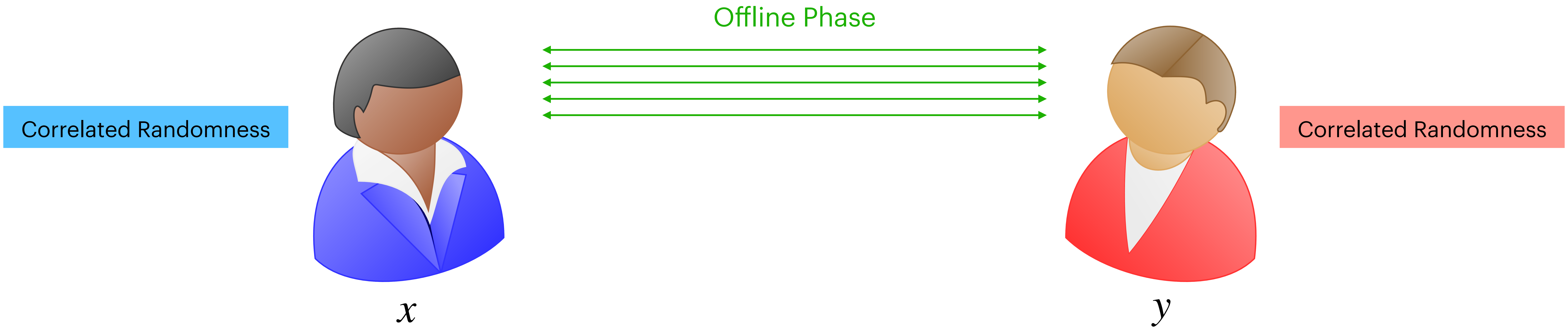


Offline Phase

x

y

Preprocessing Model for Secure Computation



Correlated Randomness

Offline Phase

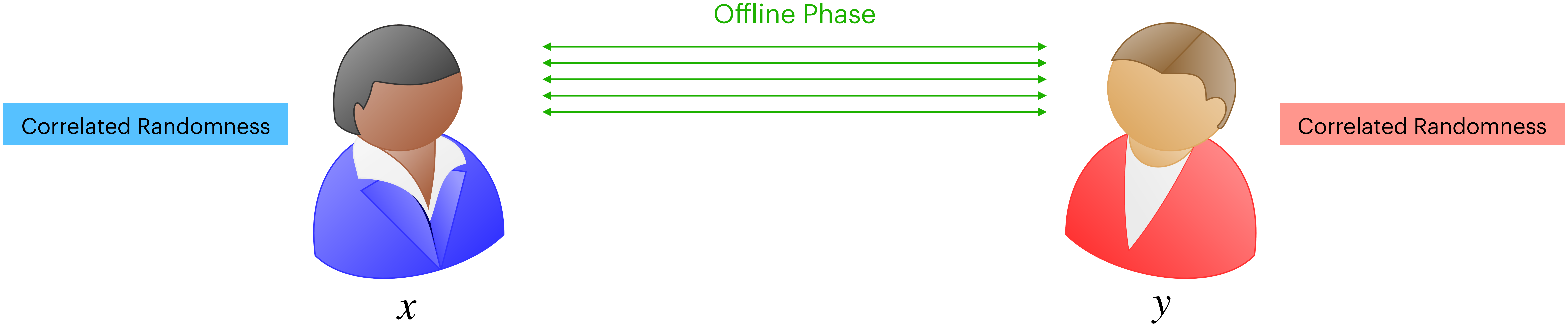
Correlated Randomness

x

y

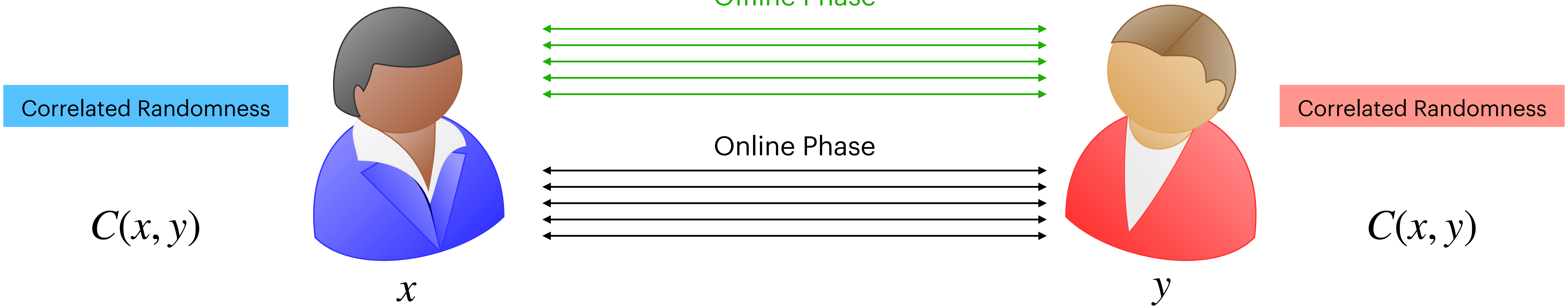
Preprocessing Model for Secure Computation

Offline phase is independent of inputs and evaluated circuit



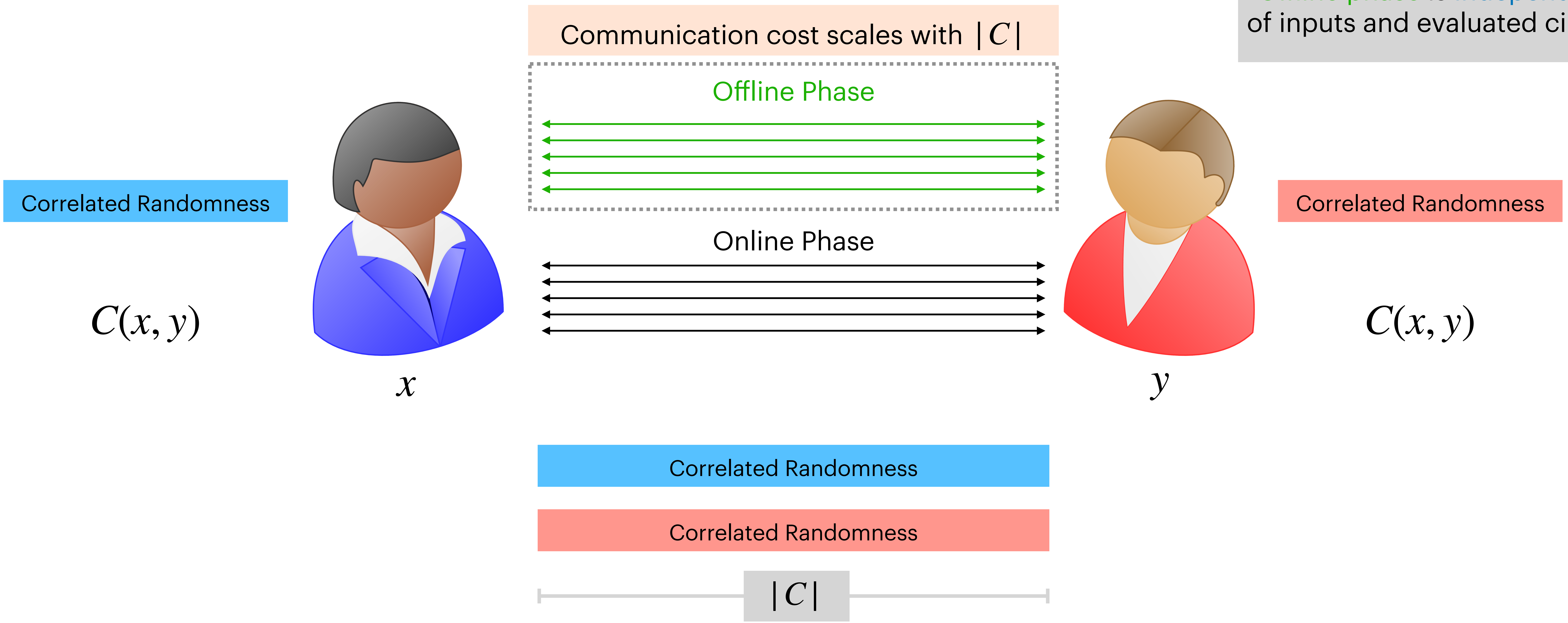
Preprocessing Model for Secure Computation

Offline phase is independent of inputs and evaluated circuit



Preprocessing Model for Secure Computation

Offline phase is independent of inputs and evaluated circuit



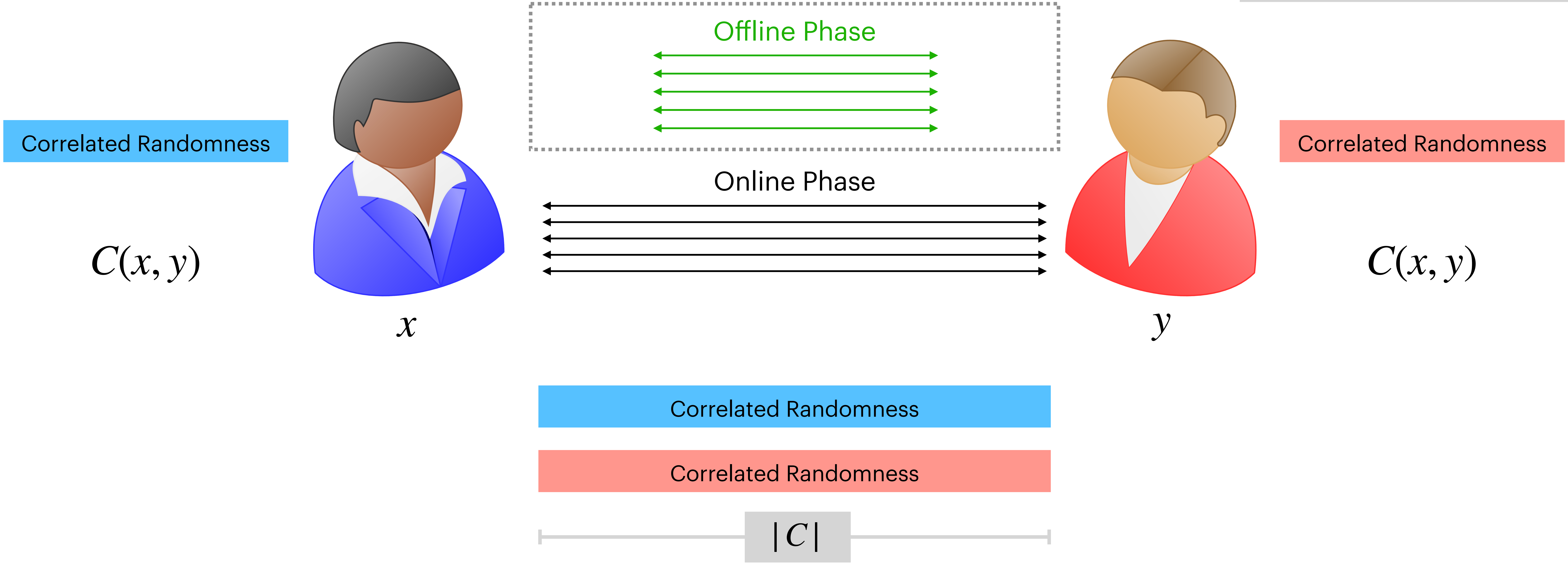
Communication cost scales with $|C|$

Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]

Public-key PCF: Generate preprocessing with low offline communication cost

Offline phase is independent of inputs and evaluated circuit



Offline Phase

Online Phase

Correlated Randomness

$C(x, y)$

x

Correlated Randomness

$C(x, y)$

y

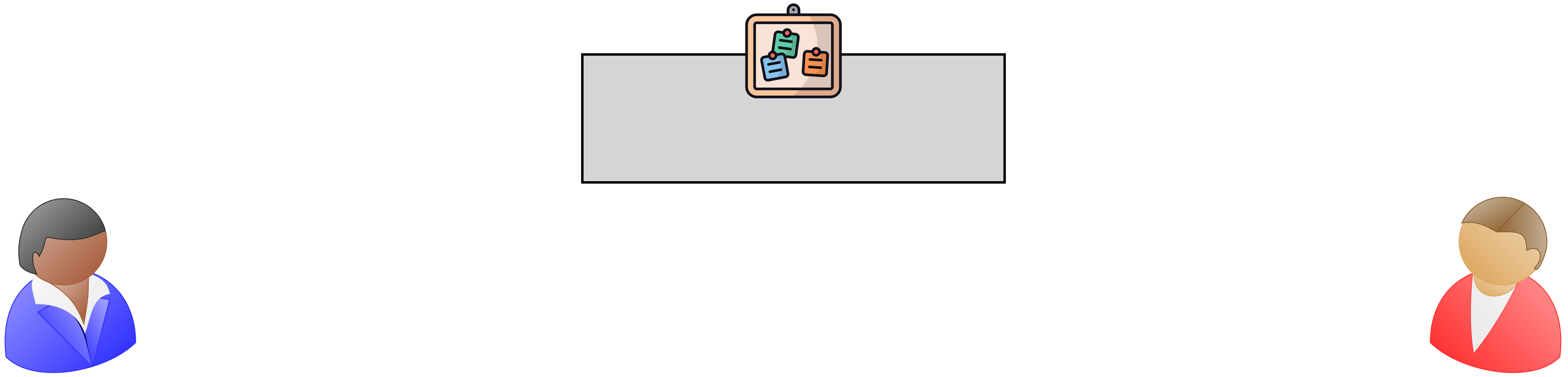
Correlated Randomness

Correlated Randomness

$|C|$

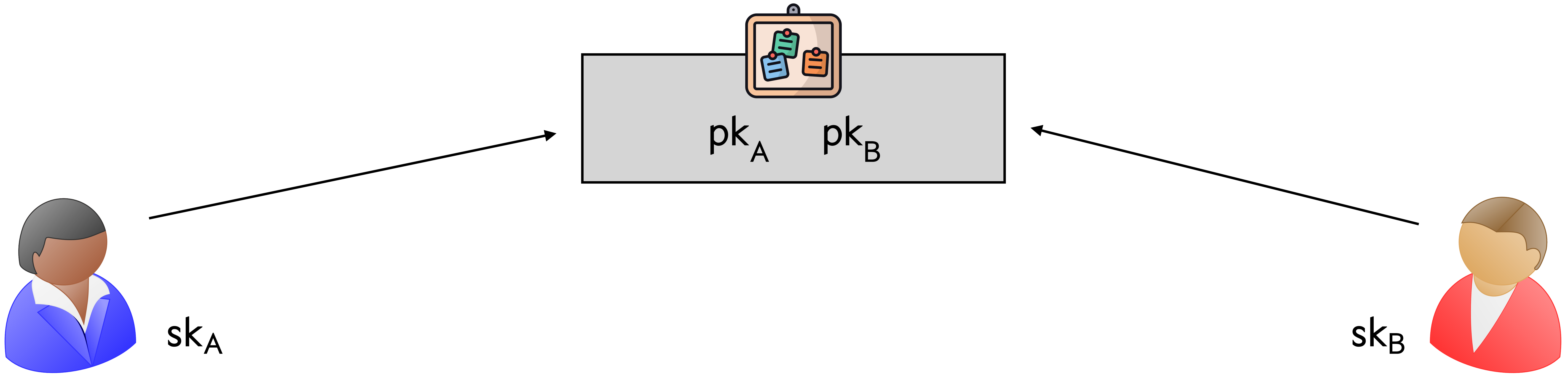
Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]



Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]



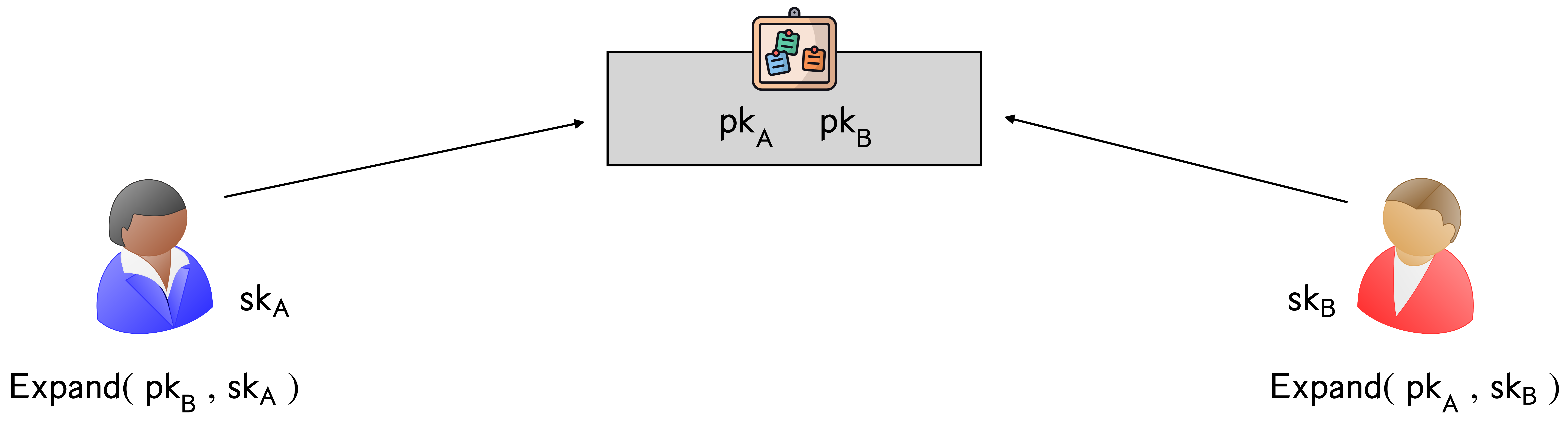
pk_A pk_B

sk_A

sk_B

Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]



pk_A pk_B

sk_A

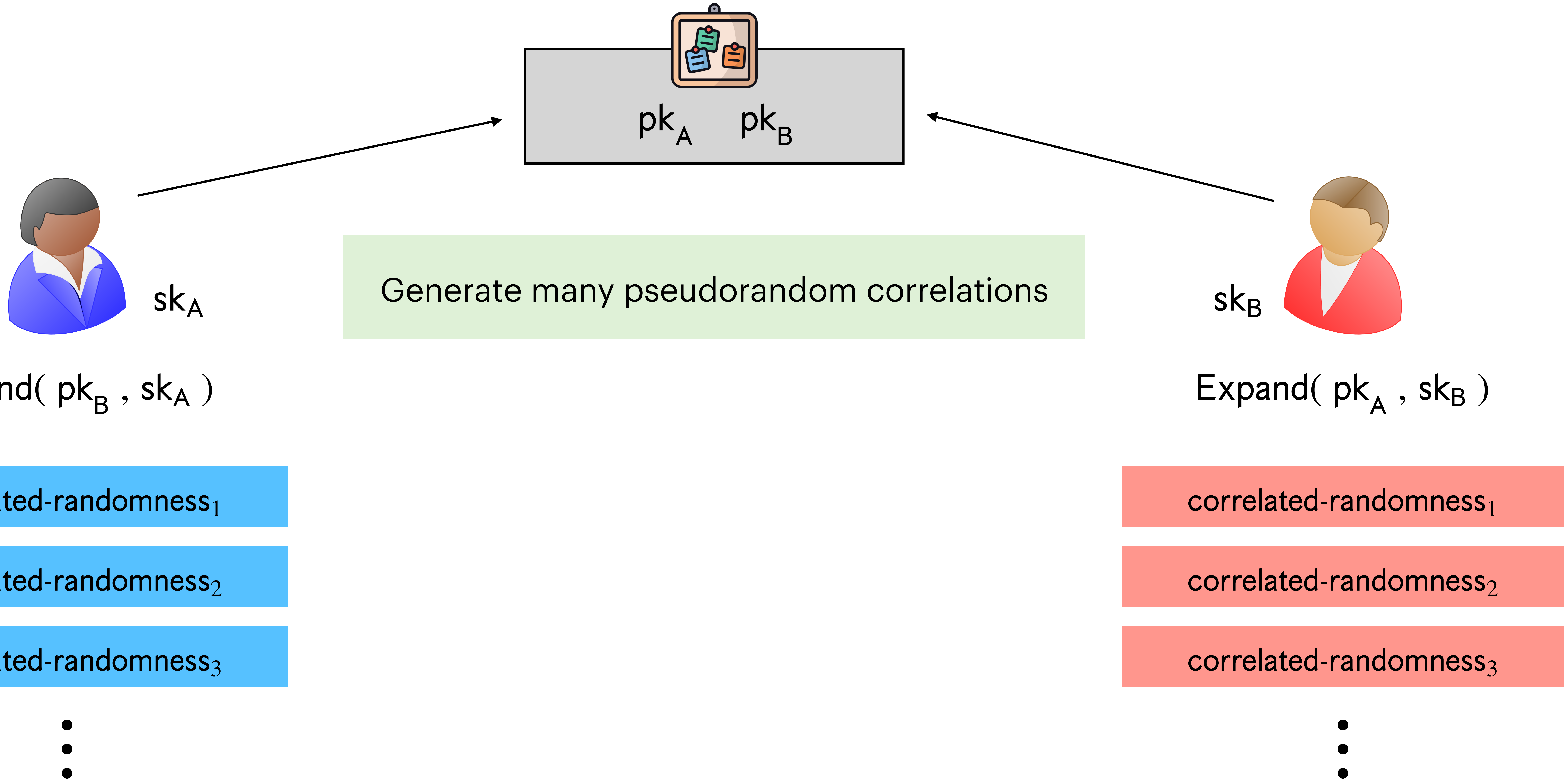
sk_B

Expand(pk_B , sk_A)

Expand(pk_A , sk_B)

Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]



pk_A pk_B

sk_A

Generate many pseudorandom correlations

sk_B

Expand(pk_B , sk_A)

Expand(pk_A , sk_B)

correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

correlated-randomness$_1$

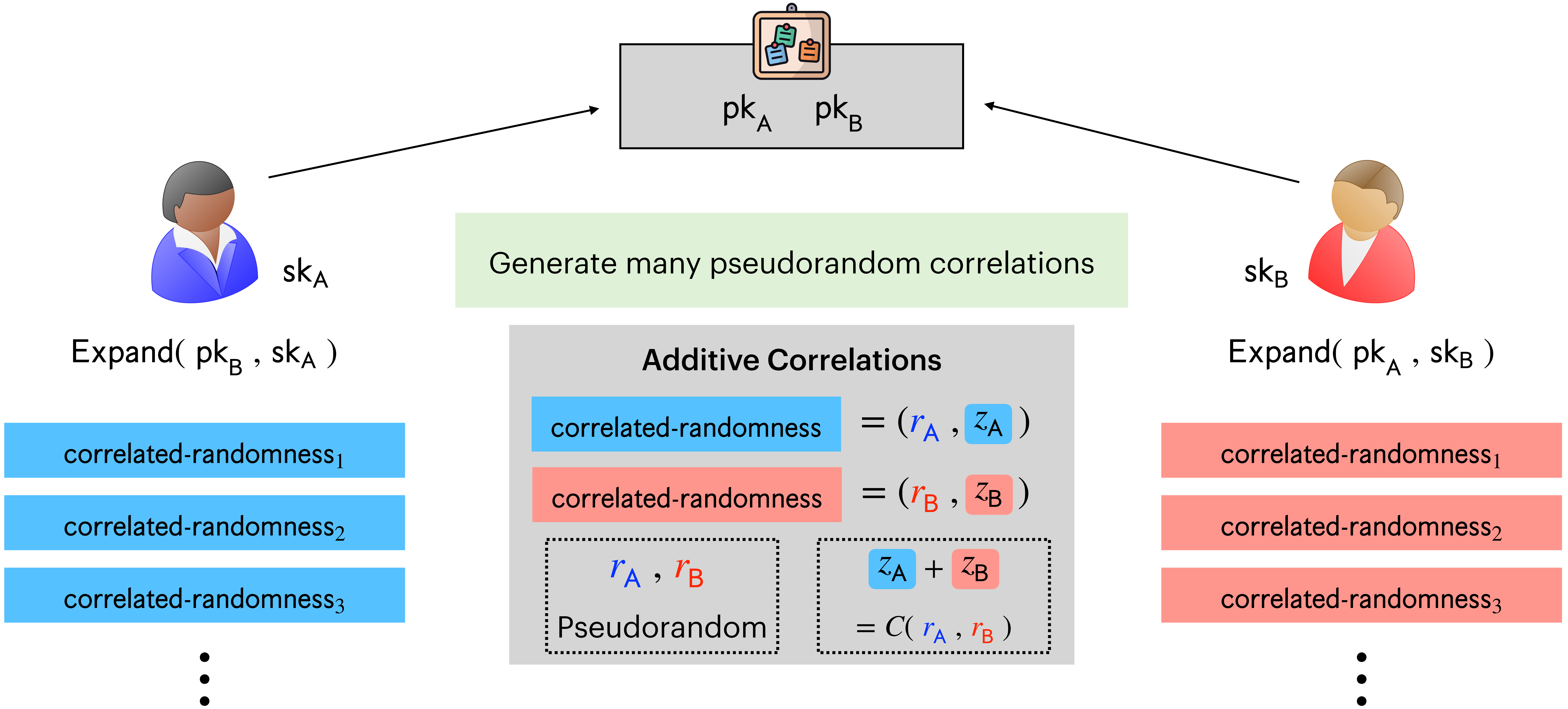
correlated-randomness$_2$

correlated-randomness$_3$

⋮

Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov’21] [Bui-Couteau-Meyer-Passalègue-Riahinia’24]



pk_A pk_B

sk_A

sk_B

Generate many pseudorandom correlations

Expand(pk_B , sk_A)

Expand(pk_A , sk_B)

correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

Additive Correlations

correlated-randomness $= (r_A , z_A)$

correlated-randomness $= (r_B , z_B)$

r_A , r_B

Pseudorandom

$z_A + z_B$

$= C(r_A , r_B)$

correlated-randomness$_1$

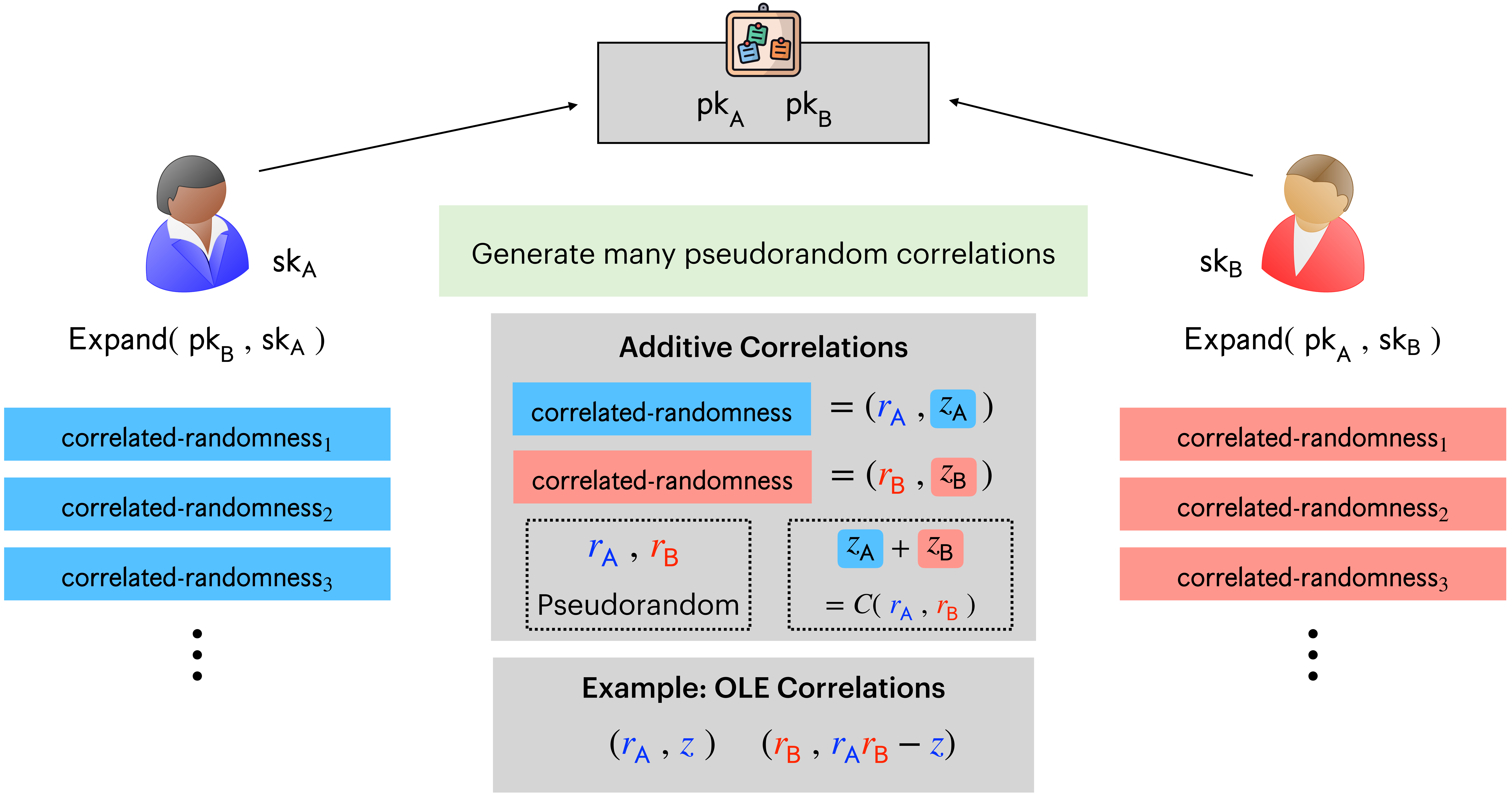
correlated-randomness$_2$

correlated-randomness$_3$

⋮

Public-Key Pseudorandom Correlation Functions

[Orlandi-Scholl-Yakoubov'21] [Bui-Couteau-Meyer-Passalègue-Riahinia'24]



pk_A pk_B

sk_A

Generate many pseudorandom correlations

sk_B

Expand(pk_B , sk_A)

correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

Additive Correlations

correlated-randomness $= (r_A , z_A)$

correlated-randomness $= (r_B , z_B)$

r_A , r_B

Pseudorandom

$z_A + z_B$

$= C(r_A , r_B)$

Example: OLE Correlations

(r_A , z) $(r_B , r_A r_B - z)$

Expand(pk_A , sk_B)

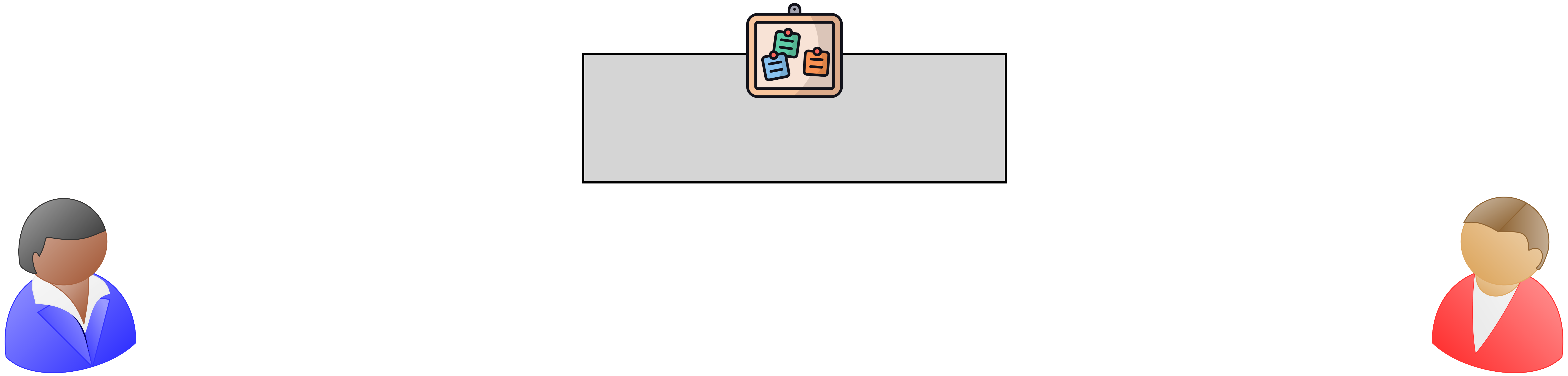
correlated-randomness$_1$

correlated-randomness$_2$

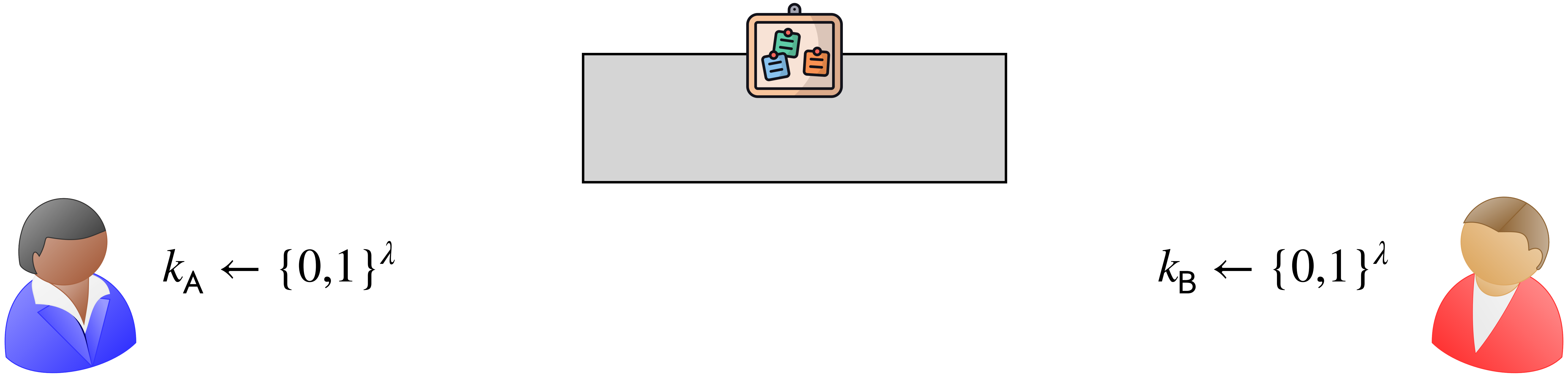
correlated-randomness$_3$

⋮

Application 2: Public-Key PCF for Additive Correlations



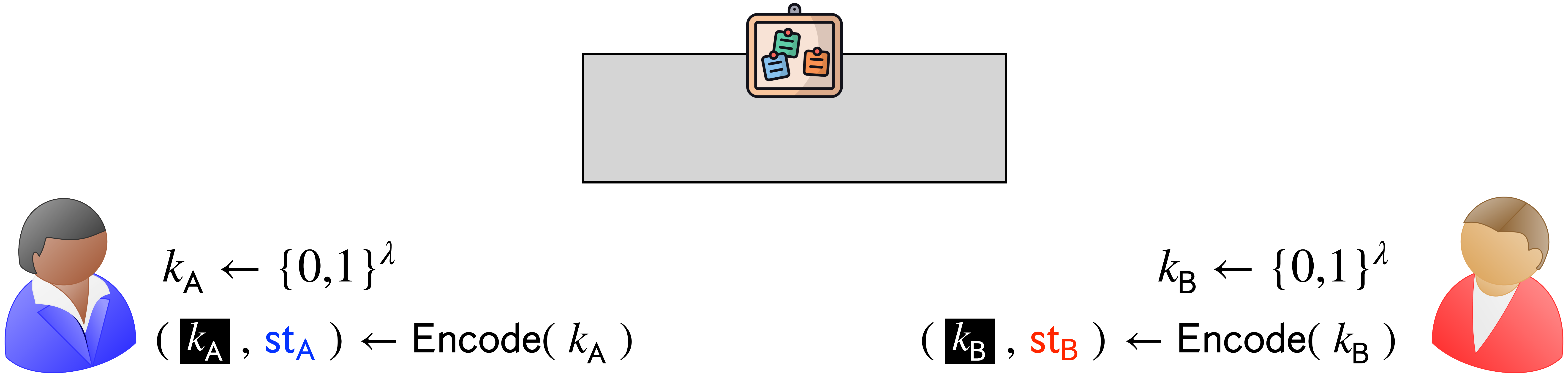
Application 2: Public-Key PCF for Additive Correlations



$k_{\mathsf{A}} \leftarrow \{0,1\}^{\lambda}$

$k_{\mathsf{B}} \leftarrow \{0,1\}^{\lambda}$

Application 2: Public-Key PCF for Additive Correlations



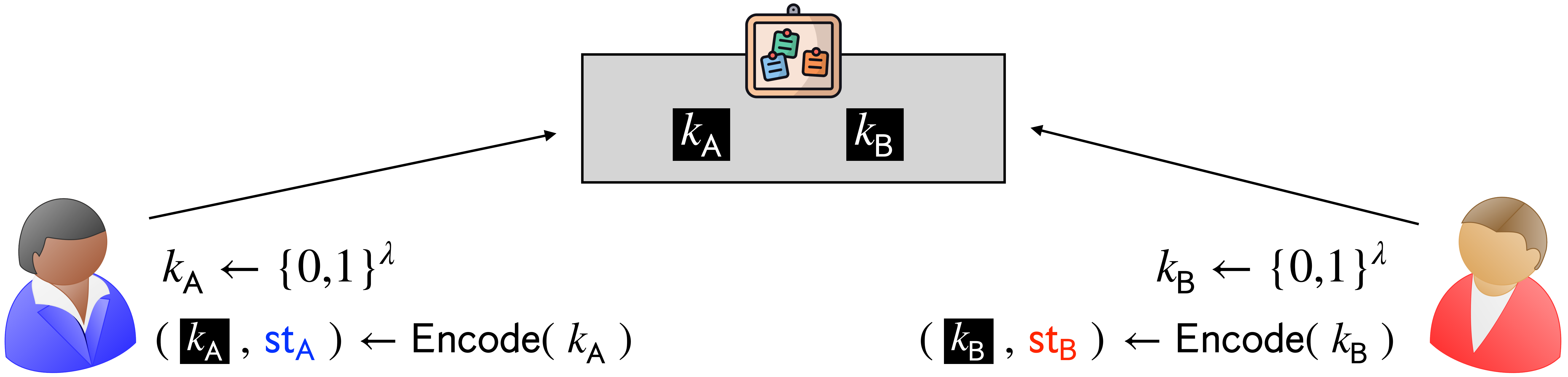
$k_{\mathsf{A}} \leftarrow \{0,1\}^{\lambda}$

$(k_{\mathsf{A}} , \mathsf{st}_{\mathsf{A}}) \leftarrow \mathsf{Encode}(k_{\mathsf{A}})$

$k_{\mathsf{B}} \leftarrow \{0,1\}^{\lambda}$

$(k_{\mathsf{B}} , \mathsf{st}_{\mathsf{B}}) \leftarrow \mathsf{Encode}(k_{\mathsf{B}})$

Application 2: Public-Key PCF for Additive Correlations



$\boxed{k_A}$ $\boxed{k_B}$

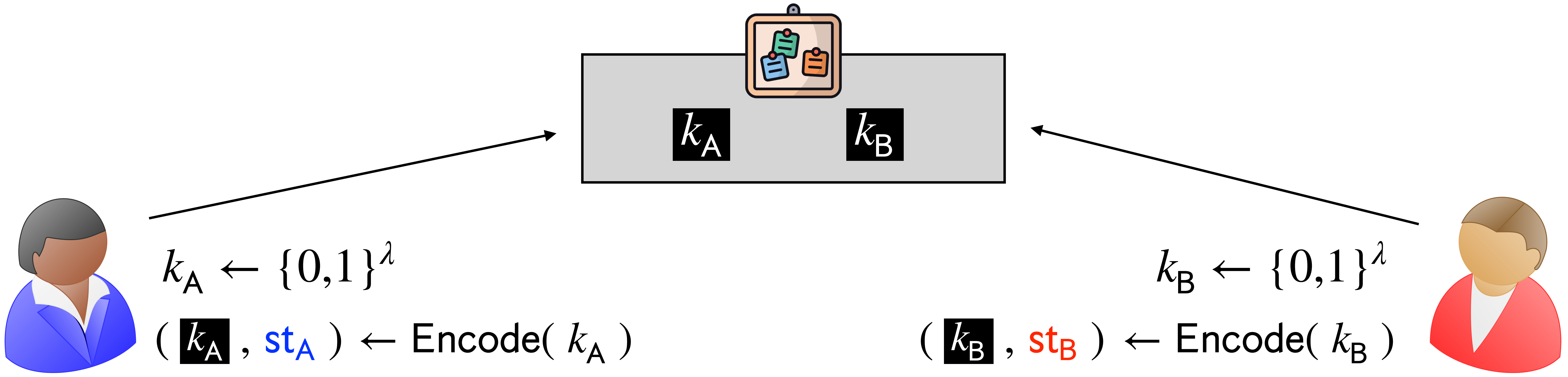
$k_A \leftarrow \{0,1\}^\lambda$

$(\ \boxed{k_A}\ ,\ \mathsf{st}_A\ ) \leftarrow \mathsf{Encode}(\ k_A\ )$

$k_B \leftarrow \{0,1\}^\lambda$

$(\ \boxed{k_B}\ ,\ \mathsf{st}_B\ ) \leftarrow \mathsf{Encode}(\ k_B\ )$

Application 2: Public-Key PCF for Additive Correlations



k_{A} k_{B}

$k_{\mathsf{A}} \leftarrow \{0,1\}^{\lambda}$

$(\ k_{\mathsf{A}}\ ,\ \mathsf{st}_{\mathsf{A}}\ ) \leftarrow \mathsf{Encode}(\ k_{\mathsf{A}}\ )$

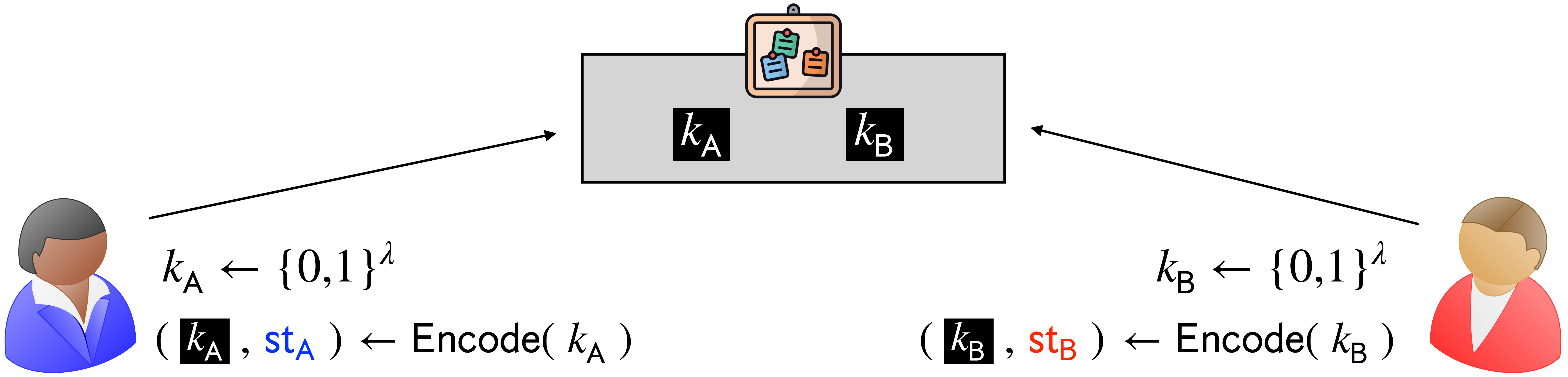
$k_{\mathsf{B}} \leftarrow \{0,1\}^{\lambda}$

$(\ k_{\mathsf{B}}\ ,\ \mathsf{st}_{\mathsf{B}}\ ) \leftarrow \mathsf{Encode}(\ k_{\mathsf{B}}\ )$

$r_{\mathsf{A}} = \mathsf{PRF}(\ k_{\mathsf{A}}\ ,\ i\ )$

$r_{\mathsf{B}} = \mathsf{PRF}(\ k_{\mathsf{B}}\ ,\ i\ )$

Application 2: Public-Key PCF for Additive Correlations



k_{A} k_{B}

$k_{\mathsf{A}} \leftarrow \{0,1\}^{\lambda}$

$(k_{\mathsf{A}} , \mathsf{st}_{\mathsf{A}}) \leftarrow \mathsf{Encode}(k_{\mathsf{A}})$

$k_{\mathsf{B}} \leftarrow \{0,1\}^{\lambda}$

$(k_{\mathsf{B}} , \mathsf{st}_{\mathsf{B}}) \leftarrow \mathsf{Encode}(k_{\mathsf{B}})$

$r_{\mathsf{A}} = \mathsf{PRF}(k_{\mathsf{A}} , i)$

$z_{\mathsf{A}} \leftarrow \mathsf{Eval}(C_i^* , \mathsf{st}_{\mathsf{A}} , k_{\mathsf{B}})$

$C_i^*(k_{\mathsf{A}} , k_{\mathsf{B}})$

$r_{\mathsf{A}} = \mathsf{PRF}(k_{\mathsf{A}} , i)$

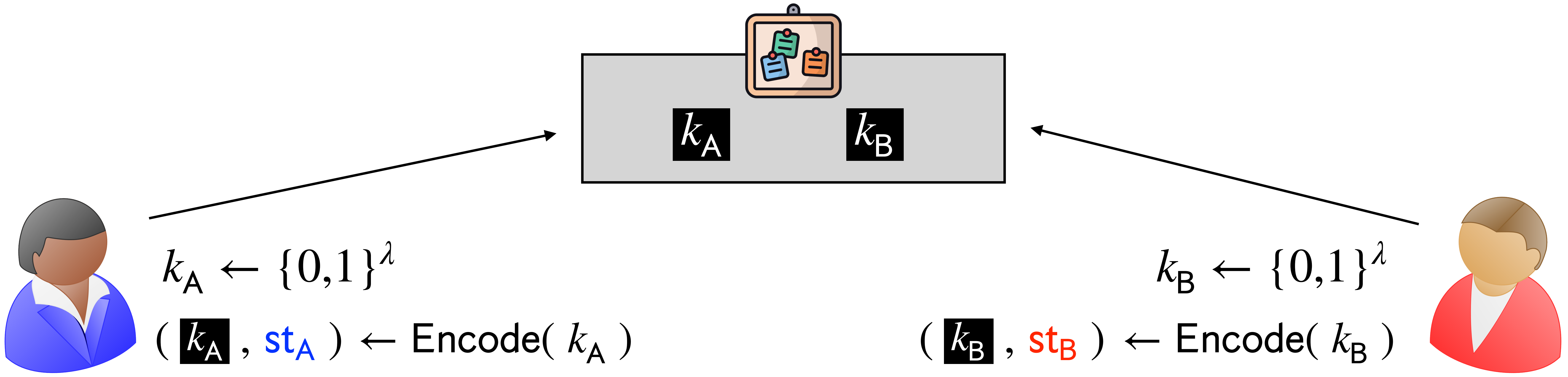
$r_{\mathsf{B}} = \mathsf{PRF}(k_{\mathsf{B}} , i)$

Output $C(r_{\mathsf{A}} , r_{\mathsf{B}})$

$r_{\mathsf{B}} = \mathsf{PRF}(k_{\mathsf{B}} , i)$

$z_{\mathsf{B}} \leftarrow \mathsf{Eval}(C_i^* , \mathsf{st}_{\mathsf{B}} , k_{\mathsf{A}})$

Application 2: Public-Key PCF for Additive Correlations



$\boxed{k_A}$ $\boxed{k_B}$

$k_A \leftarrow \{0,1\}^\lambda$

$(\boxed{k_A} , \mathsf{st}_A) \leftarrow \text{Encode}(k_A)$

$k_B \leftarrow \{0,1\}^\lambda$

$(\boxed{k_B} , \mathsf{st}_B) \leftarrow \text{Encode}(k_B)$

$r_A = \text{PRF}(k_A , i)$

$z_A \leftarrow \text{Eval}(C_i^* , \mathsf{st}_A , \boxed{k_B})$

correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

$C_i^*(k_A , k_B)$

$r_A = \text{PRF}(k_A , i)$

$r_B = \text{PRF}(k_B , i)$

Output $C(r_A , r_B)$

Unbounded number of correlations

$r_B = \text{PRF}(k_B , i)$

$z_B \leftarrow \text{Eval}(C_i^* , \mathsf{st}_B , \boxed{k_A})$

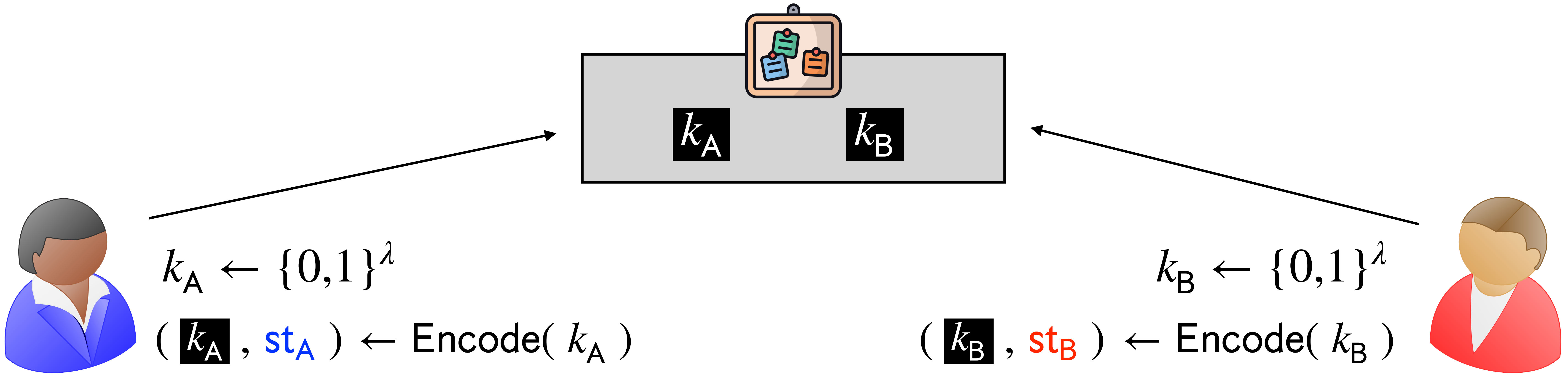
correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

Application 2: Public-Key PCF for Additive Correlations



$k_A \leftarrow \{0,1\}^\lambda$

(k_A , st_A) ← Encode(k_A)

$k_B \leftarrow \{0,1\}^\lambda$

(k_B , st_B) ← Encode(k_B)

r_A = PRF(k_A , i)

z_A ← Eval(C_i^* , st_A , k_B)

correlated-randomness$_1$

correlated-randomness$_2$

correlated-randomness$_3$

⋮

C_i^*(k_A , k_B)

r_A = PRF(k_A , i)

r_B = PRF(k_B , i)

Output C(r_A , r_B)

Unbounded number of correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup

r_B = PRF(k_B , i)

z_B ← Eval(C_i^* , st_B , k_A)

correlated-randomness$_1$

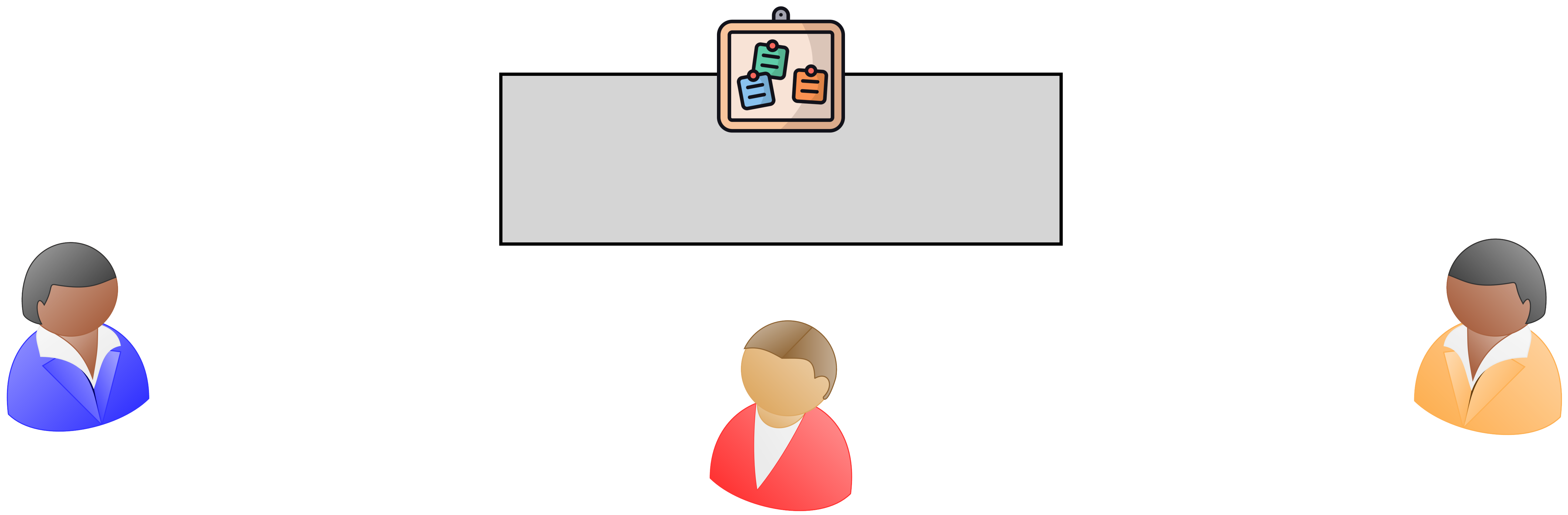
correlated-randomness$_2$

correlated-randomness$_3$

⋮

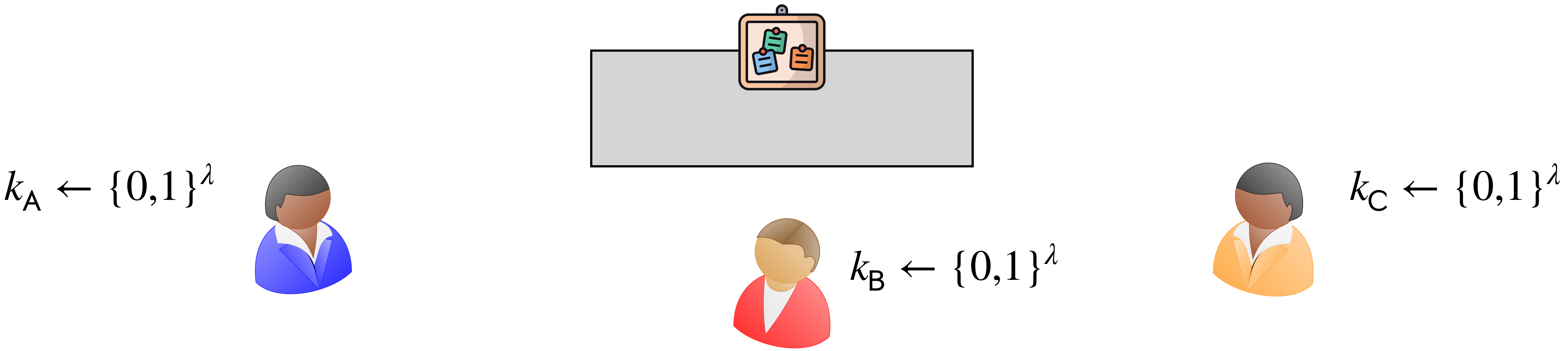
Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



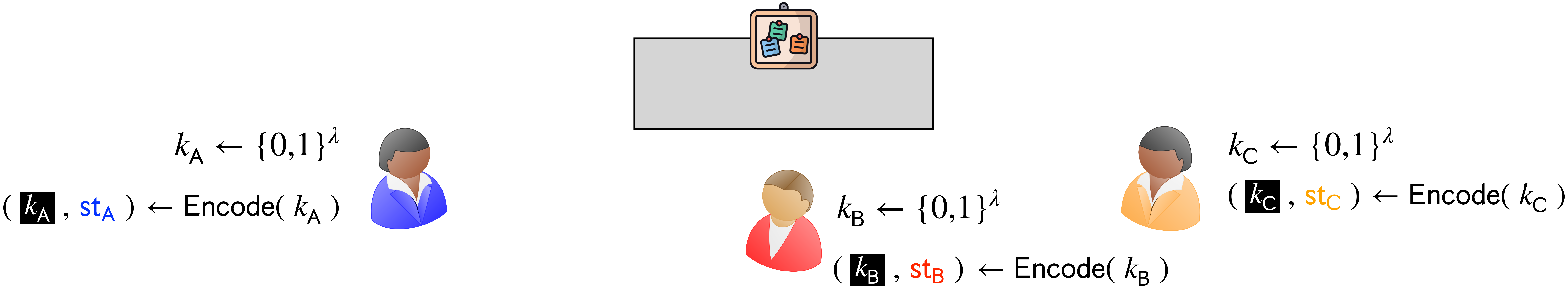
$k_{\mathsf{A}} \leftarrow \{0,1\}^\lambda$

$k_{\mathsf{B}} \leftarrow \{0,1\}^\lambda$

$k_{\mathsf{C}} \leftarrow \{0,1\}^\lambda$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



$k_A \leftarrow \{0,1\}^\lambda$

$(k_A , \mathsf{st}_A) \leftarrow \text{Encode}(k_A)$

$k_B \leftarrow \{0,1\}^\lambda$

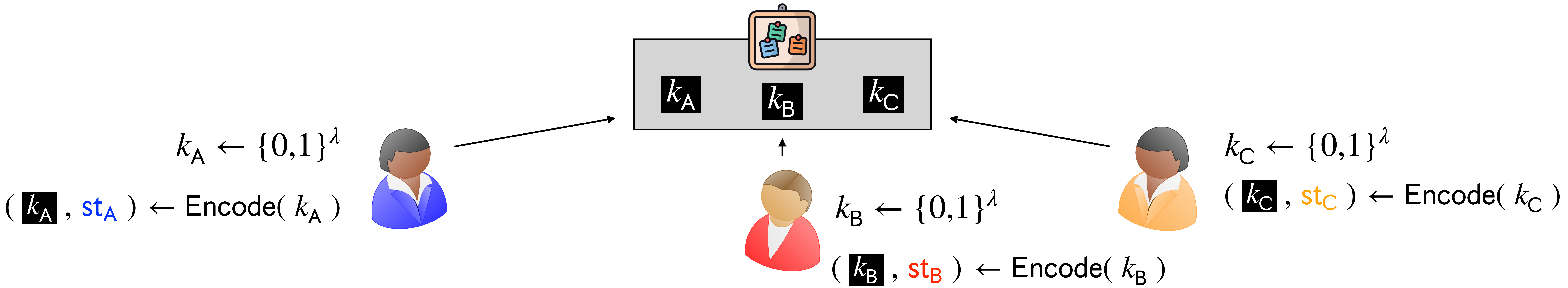
$(k_B , \mathsf{st}_B) \leftarrow \text{Encode}(k_B)$

$k_C \leftarrow \{0,1\}^\lambda$

$(k_C , \mathsf{st}_C) \leftarrow \text{Encode}(k_C)$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



$\boxed{k_A}$ $\boxed{k_B}$ $\boxed{k_C}$

$k_A \leftarrow \{0,1\}^\lambda$

$(\ \boxed{k_A}\ , \mathsf{st}_A\ ) \leftarrow \text{Encode}(\ k_A\ )$

$k_B \leftarrow \{0,1\}^\lambda$

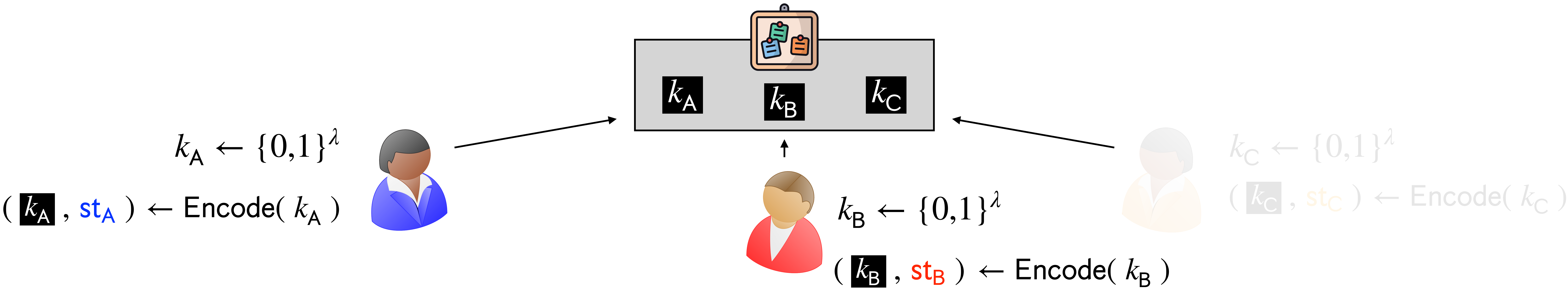
$(\ \boxed{k_B}\ , \mathsf{st}_B\ ) \leftarrow \text{Encode}(\ k_B\ )$

$k_C \leftarrow \{0,1\}^\lambda$

$(\ \boxed{k_C}\ , \mathsf{st}_C\ ) \leftarrow \text{Encode}(\ k_C\ )$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



$\boxed{k_A}$ $\boxed{k_B}$ $\boxed{k_C}$

$k_A \leftarrow \{0,1\}^\lambda$

$(\boxed{k_A} , \mathsf{st}_A) \leftarrow \text{Encode}(k_A)$

$k_B \leftarrow \{0,1\}^\lambda$

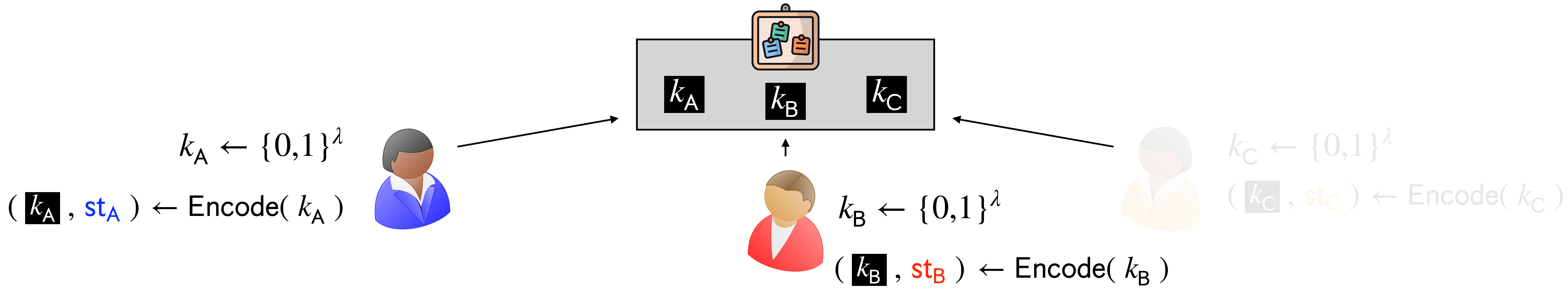
$(\boxed{k_B} , \mathsf{st}_B) \leftarrow \text{Encode}(k_B)$

$k_C \leftarrow \{0,1\}^\lambda$

$(\boxed{k_C} , \mathsf{st}_C) \leftarrow \text{Encode}(k_C)$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

(k_A , st_A) $\leftarrow$ Encode(k_A)

$k_B \leftarrow \{0,1\}^\lambda$

(k_B , st_B) $\leftarrow$ Encode(k_B)

$k_C \leftarrow \{0,1\}^\lambda$

(k_C , st_C) $\leftarrow$ Encode(k_C)

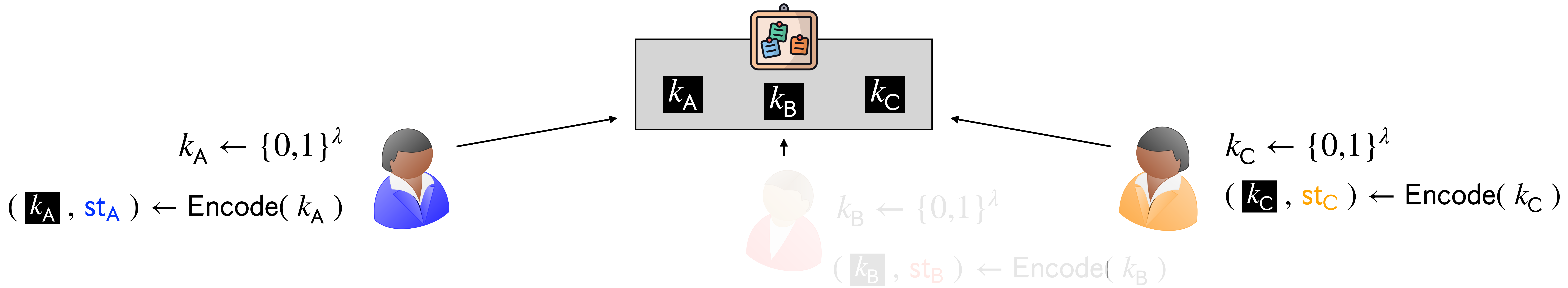
Uses k_A and k_B

correlated-randomness$_{AB}$

correlated-randomness$_{AB}$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings ⟹ non-interactive offline phase i.e., public key setup



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

$(\, \boxed{k_A}\, , \mathsf{st}_A\,) \leftarrow \mathsf{Encode}(\, k_A\,)$

$k_B \leftarrow \{0,1\}^\lambda$

$(\, \boxed{k_B}\, , \mathsf{st}_B\,) \leftarrow \mathsf{Encode}(\, k_B\,)$

$k_C \leftarrow \{0,1\}^\lambda$

$(\, \boxed{k_C}\, , \mathsf{st}_C\,) \leftarrow \mathsf{Encode}(\, k_C\,)$

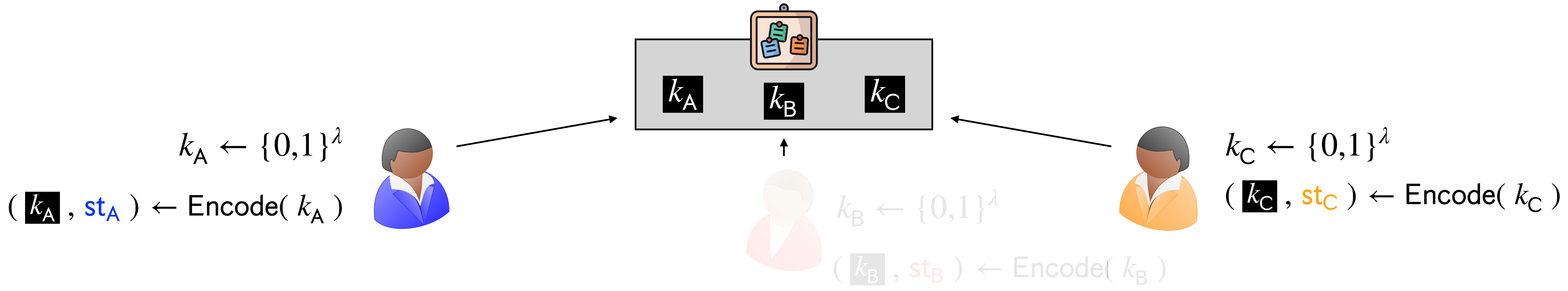
Use $\boxed{k_A}$ and $\boxed{k_B}$

correlated-randomness$_{AB}$

correlated-randomness$_{AB}$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

(k_A , st_A) $\leftarrow$ Encode(k_A)

$k_B \leftarrow \{0,1\}^\lambda$

(k_B , st_B) $\leftarrow$ Encode(k_B)

$k_C \leftarrow \{0,1\}^\lambda$

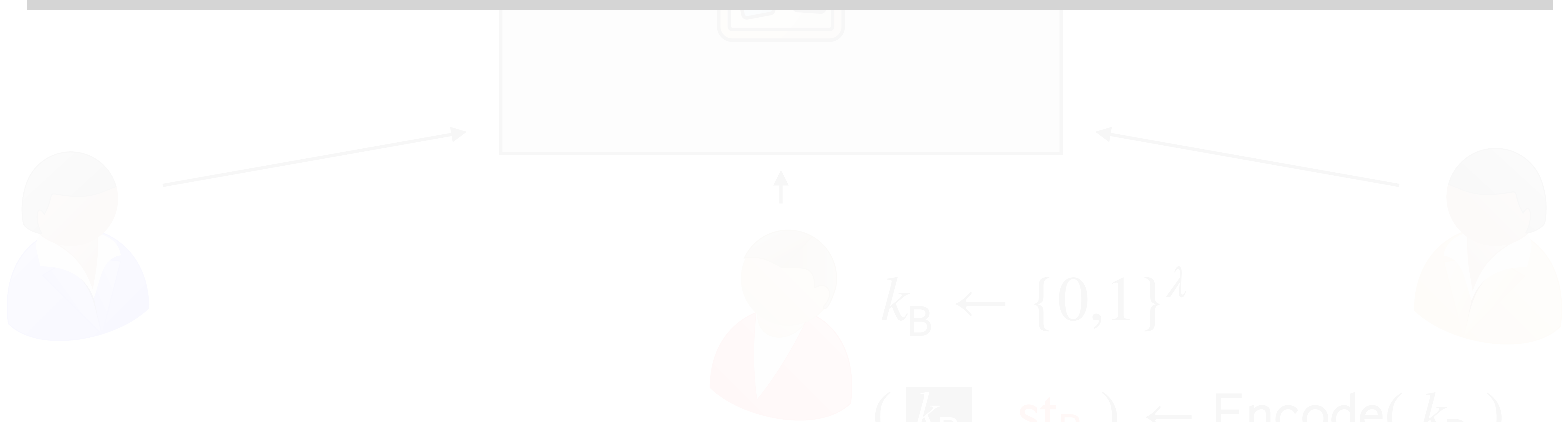
(k_C , st_C) $\leftarrow$ Encode(k_C)

Use k_A and k_B	correlated-randomness$_{AB}$	correlated-randomness$_{AB}$	
Use k_A and k_C	correlated-randomness$_{AC}$		correlated-randomness$_{AC}$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings ⟹ non-interactive offline phase i.e., public key setup

What about multi-party correlations?



$k_A \leftarrow \{0,1\}^\lambda$

$(\boxed{k_A} , \mathsf{st}_A) \leftarrow \mathsf{Encode}(k_A)$

$k_B \leftarrow \{0,1\}^\lambda$

$(\boxed{k_B} , \mathsf{st}_B) \leftarrow \mathsf{Encode}(k_B)$

$k_C \leftarrow \{0,1\}^\lambda$

$(\boxed{k_C} , \mathsf{st}_C) \leftarrow \mathsf{Encode}(k_C)$

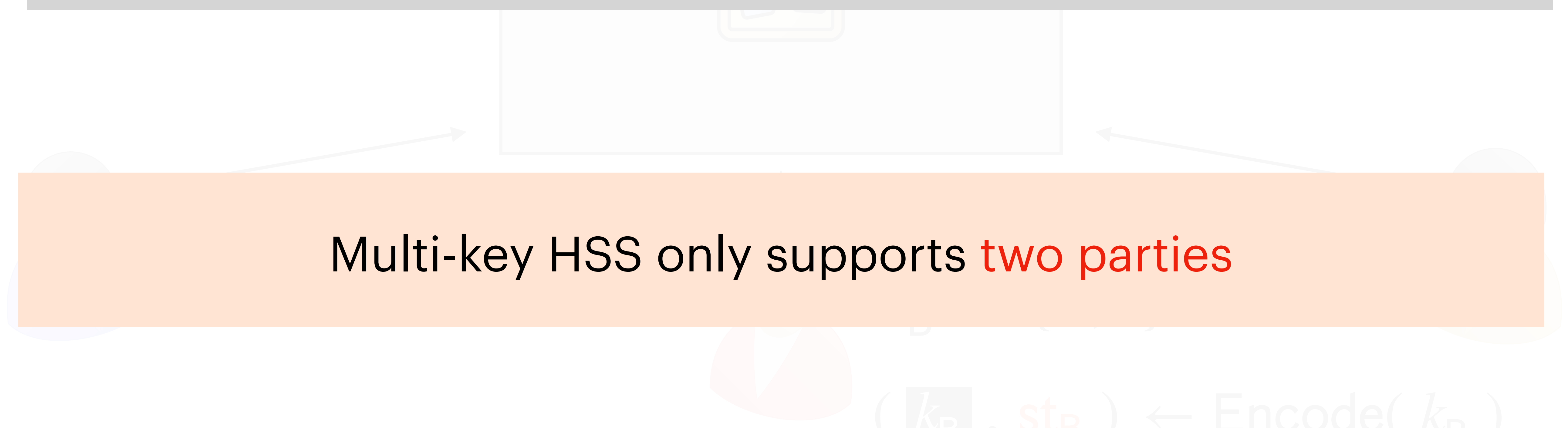
Uses $\boxed{k_A}$ and $\boxed{k_B}$ correlated-randomness$_{AB}$ correlated-randomness$_{AB}$

Uses $\boxed{k_A}$ and $\boxed{k_C}$ correlated-randomness$_{AC}$ correlated-randomness$_{AC}$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings ⟹ non-interactive offline phase i.e., public key setup

What about multi-party correlations?



$k_A \leftarrow \{0,1\}^\lambda$

$(\ k_A\ ,\ st_A\ ) \leftarrow \text{Encode}(\ k_A\ )$

$k_C \leftarrow \{0,1\}^\lambda$

$(\ k_C\ ,\ st_C\ ) \leftarrow \text{Encode}(\ k_C\ )$

$(\ k_B\ ,\ st_B\ ) \leftarrow \text{Encode}(\ k_B\ )$

Multi-key HSS only supports two parties

Uses k_A and k_B correlated-randomness$_{AB}$ correlated-randomness$_{AB}$

Uses k_A and k_C correlated-randomness$_{AC}$ correlated-randomness$_{AC}$

Application 2: Public-Key PCF for Additive Correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase i.e., public key setup

What about multi-party correlations?

$k_A \leftarrow \{0,1\}^\lambda$

(k_A , st_A) ← Encode(k_A)

$k_C \leftarrow \{0,1\}^\lambda$

(k_C , st_C) ← Encode(k_C)

Multi-key HSS only supports two parties

(k_B , st_B) ← Encode(k_B)

Uses k_A and k_B

correlate

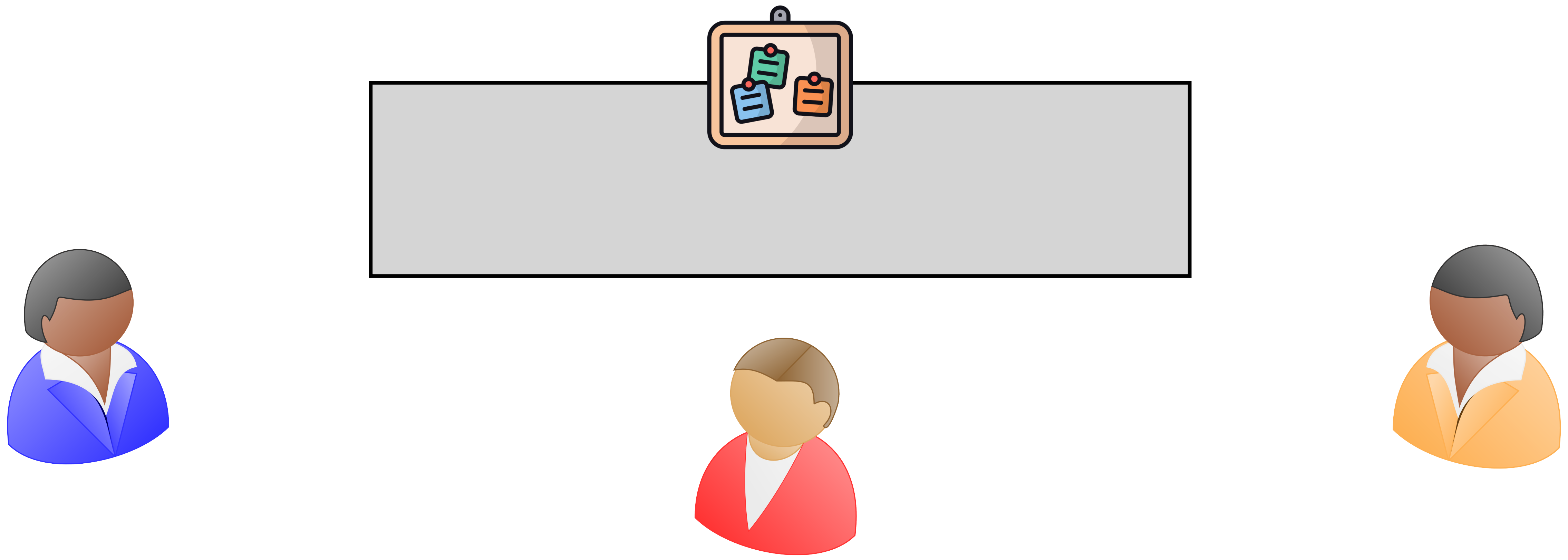
Reusability $\Longrightarrow$ Multi-party public-key PCFs for Beaver triples

Uses k_A and k_C

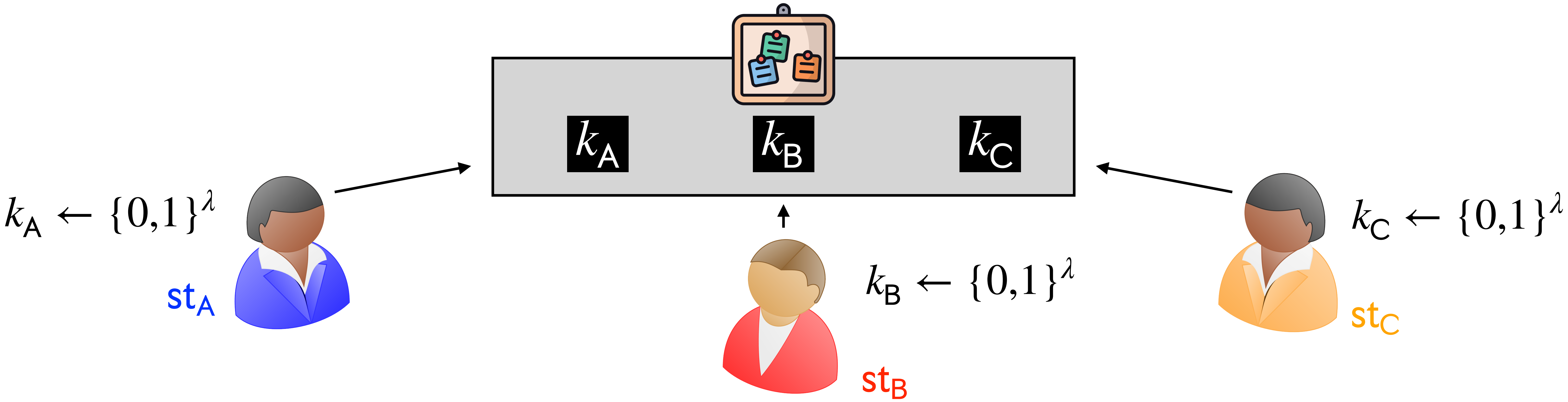
$\text{correlated-randomness}_{AC}$

$\text{correlated-randomness}_{AC}$

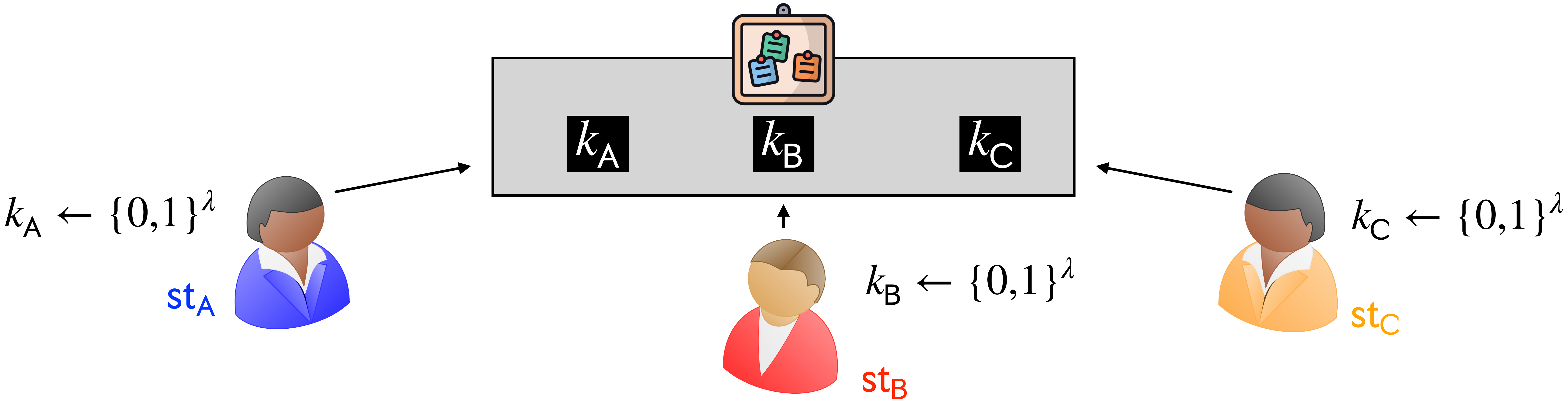
Application 3: Multi-Party Public-Key PCF for Beaver Triples



Application 3: Multi-Party Public-Key PCF for Beaver Triples


k_A
k_B
k_C
$k_A \leftarrow \{0,1\}^\lambda$
st_A
$k_B \leftarrow \{0,1\}^\lambda$
st_B
$k_C \leftarrow \{0,1\}^\lambda$
st_C

Application 3: Multi-Party Public-Key PCF for Beaver Triples

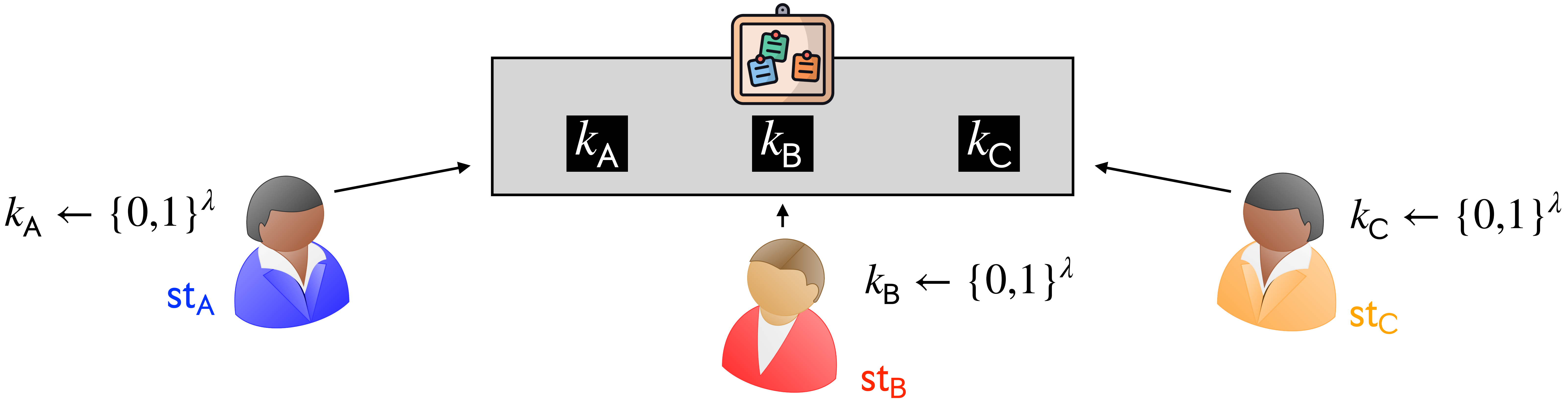

k_A
k_B
k_C
$k_A \leftarrow \{0,1\}^\lambda$
st_A
$k_B \leftarrow \{0,1\}^\lambda$
st_B
$k_C \leftarrow \{0,1\}^\lambda$
st_C

correlated-randomness$_{AB}$

correlated-randomness$_{AB}$

Pairwise OLE correlations using multi-key HSS

Application 3: Multi-Party Public-Key PCF for Beaver Triples



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

st_A

$k_B \leftarrow \{0,1\}^\lambda$

st_B

$k_C \leftarrow \{0,1\}^\lambda$

st_C

correlated-randomness$_{AB}$

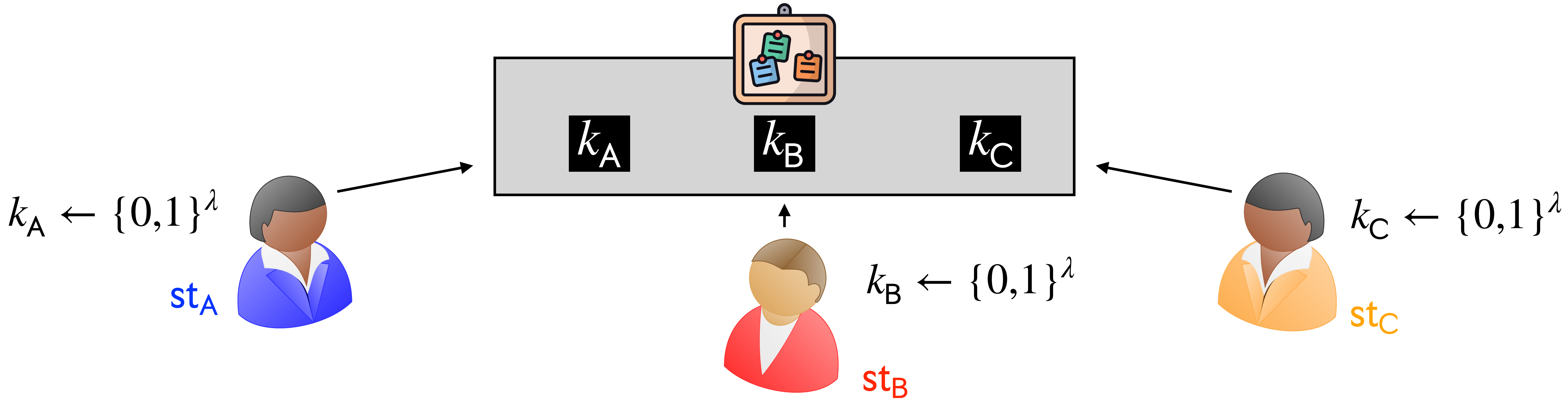
correlated-randomness$_{AB}$

correlated-randomness$_{AC}$

correlated-randomness$_{AC}$

Pairwise OLE correlations using multi-key HSS

Application 3: Multi-Party Public-Key PCF for Beaver Triples



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

st_A

$k_B \leftarrow \{0,1\}^\lambda$

st_B

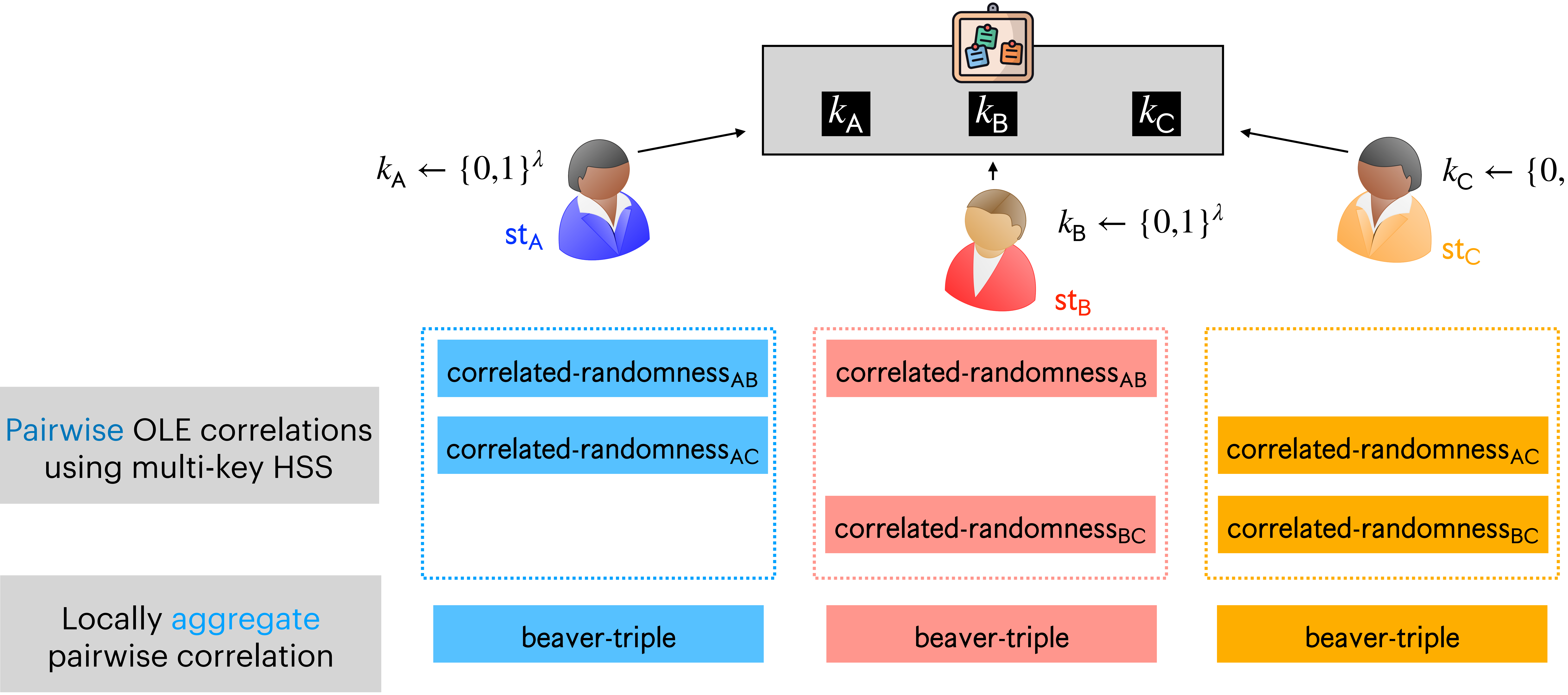
$k_C \leftarrow \{0,1\}^\lambda$

st_C

Pairwise OLE correlations using multi-key HSS

correlated-randomness$_{AB}$	correlated-randomness$_{AB}$	
correlated-randomness$_{AC}$		correlated-randomness$_{AC}$
	correlated-randomness$_{BC}$	correlated-randomness$_{BC}$

Application 3: Multi-Party Public-Key PCF for Beaver Triples



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$

st_A

$k_B \leftarrow \{0,1\}^\lambda$

st_B

$k_C \leftarrow \{0,1\}^\lambda$

st_C

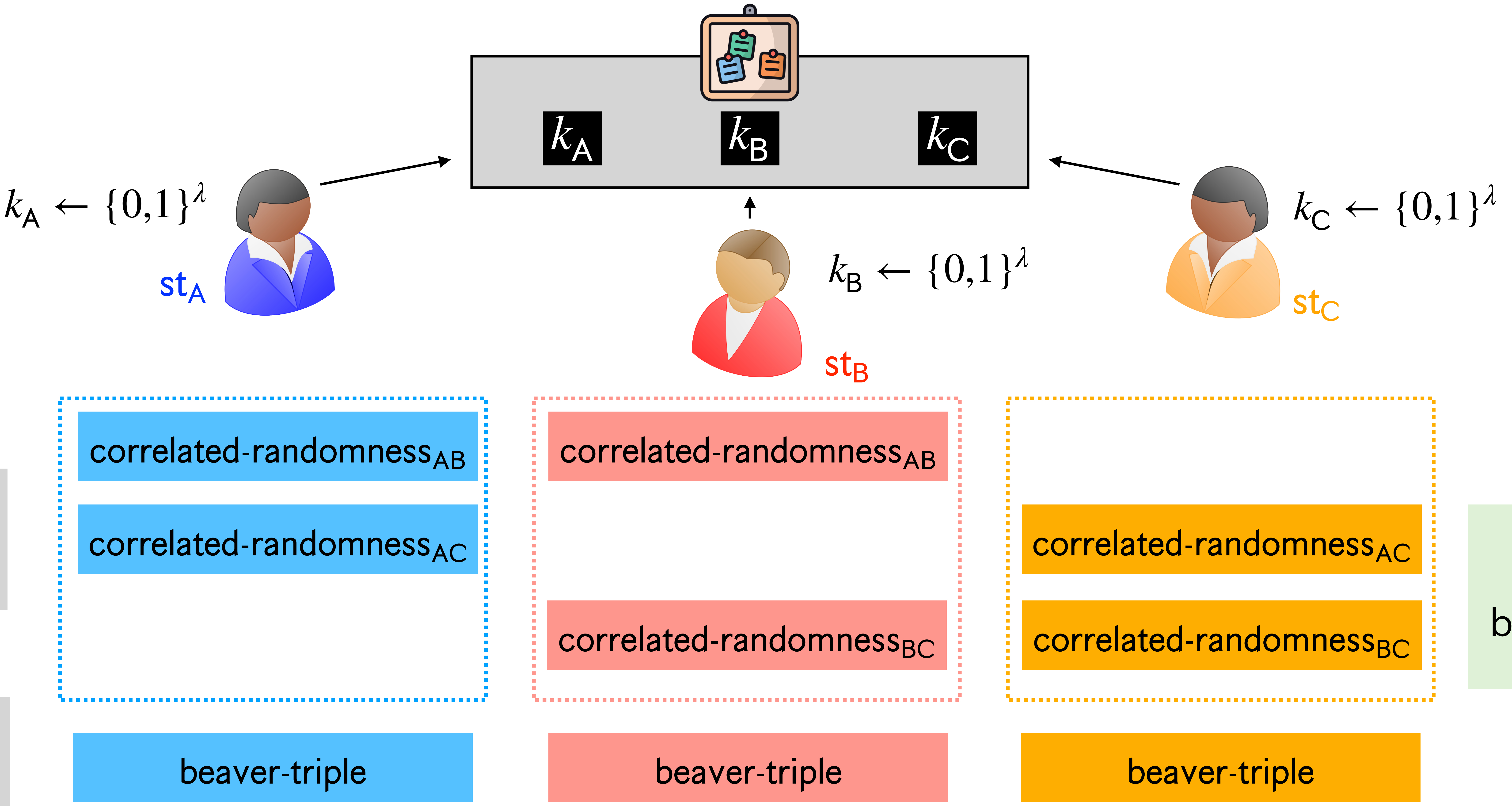
Pairwise OLE correlations using multi-key HSS

A	B	C
correlated-randomness$_{AB}$	correlated-randomness$_{AB}$	
correlated-randomness$_{AC}$		correlated-randomness$_{AC}$
	correlated-randomness$_{BC}$	correlated-randomness$_{BC}$

Locally aggregate pairwise correlation

A	B	C
beaver-triple	beaver-triple	beaver-triple

Application 3: Multi-Party Public-Key PCF for Beaver Triples



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$ st_A

$k_B \leftarrow \{0,1\}^\lambda$ st_B

$k_C \leftarrow \{0,1\}^\lambda$ st_C

correlated-randomness$_{AB}$

correlated-randomness$_{AC}$

beaver-triple

correlated-randomness$_{AB}$

correlated-randomness$_{BC}$

beaver-triple

correlated-randomness$_{AC}$

correlated-randomness$_{BC}$

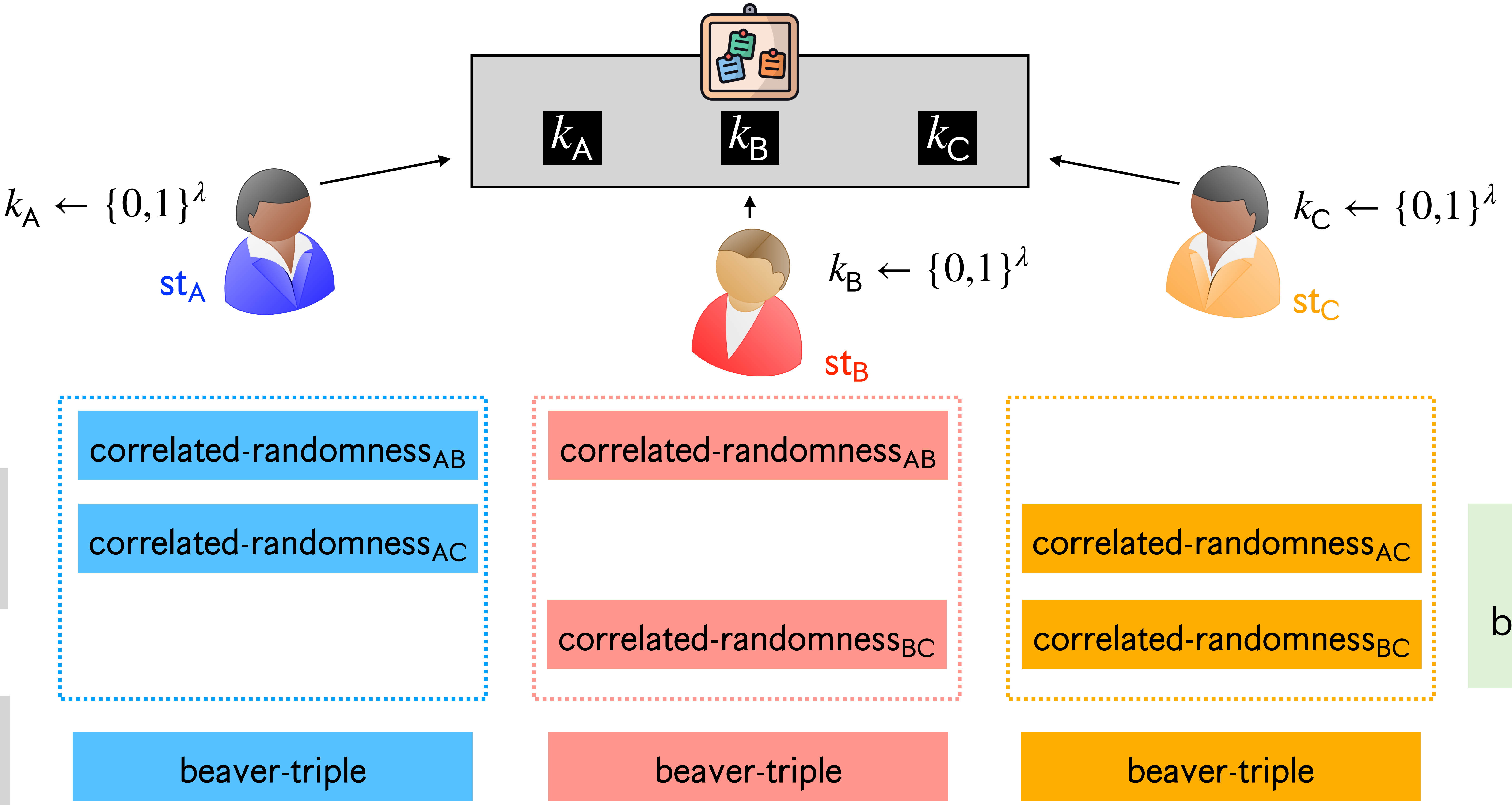
beaver-triple

Pairwise OLE correlations using multi-key HSS

Locally aggregate pairwise correlation

Unbounded number of beaver triple correlations

Application 3: Multi-Party Public-Key PCF for Beaver Triples



k_A k_B k_C

$k_A \leftarrow \{0,1\}^\lambda$ st_A

$k_B \leftarrow \{0,1\}^\lambda$ st_B

$k_C \leftarrow \{0,1\}^\lambda$ st_C

correlated-randomness$_{AB}$

correlated-randomness$_{AC}$

correlated-randomness$_{AB}$

correlated-randomness$_{BC}$

correlated-randomness$_{AC}$

correlated-randomness$_{BC}$

beaver-triple

beaver-triple

beaver-triple

Pairwise OLE correlations using multi-key HSS

Locally aggregate pairwise correlation

Unbounded number of beaver triple correlations

Reusability of input encodings $\Longrightarrow$ non-interactive offline phase with communication linear in the number of parties.

Outline

Applications

Our Results

Constructing Multi-Key HSS

Our Results: Multi-Key HSS

Two party multi-key HSS schemes for evaluating NC^1 functions

DDH

DCR

DDH over class groups

Previously known only from LWE and $i\mathcal{O}$ + DDH [Dodis-Halevi-Rothblum-Wichs'16]

Our Results: Multi-Key HSS

Two party multi-key HSS schemes for evaluating NC^1 functions

	HSS Schemes from Prior Works (Require Correlated Setup)
DDH	[Boyle-Gilboa-Ishai’16]
DCR	[Orlandi-Scholl-Yakoubov’21] [Roy-Singh’21]
DDH over class groups	[Abram-Damgård-Orlandi-Scholl'22]

Previously known only from LWE and $i\mathcal{O}$ + DDH [Dodis-Halevi-Rothblum-Wichs’16]

Our Results: Multi-Key HSS

Two party multi-key HSS schemes for evaluating NC^1 functions

Inverse polynomial correctness error

	HSS Schemes from Prior Works (Require Correlated Setup)
DDH	[Boyle-Gilboa-Ishai'16]
DCR	[Orlandi-Scholl-Yakoubov'21] [Roy-Singh'21]
DDH over class groups	[Abram-Damgård-Orlandi-Scholl'22]

Previously known only from LWE and $i\mathcal{O}$ + DDH [Dodis-Halevi-Rothblum-Wichs'16]

Our Results: Multi-Key HSS

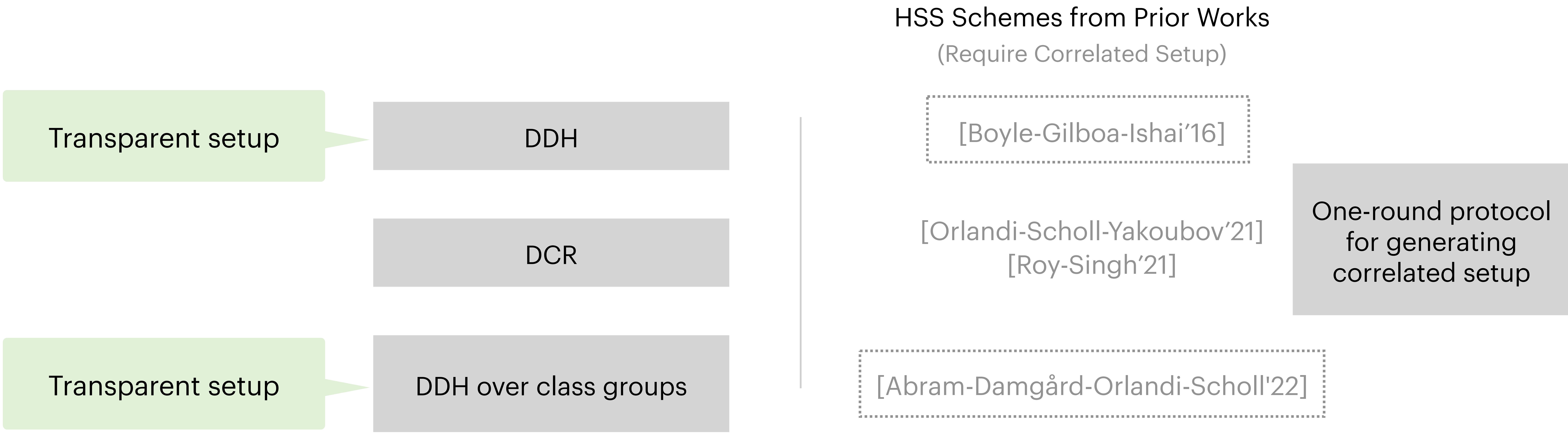
Two party multi-key HSS schemes for evaluating NC^1 functions

		HSS Schemes from Prior Works (Require Correlated Setup)
Transparent setup	DDH	[Boyle-Gilboa-Ishai’16]
	DCR	[Orlandi-Scholl-Yakoubov’21] [Roy-Singh’21]
Transparent setup	DDH over class groups	[Abram-Damgård-Orlandi-Scholl'22]

Previously known only from LWE and $i\mathcal{O}$ + DDH [Dodis-Halevi-Rothblum-Wichs’16]

Our Results: Multi-Key HSS

Two party multi-key HSS schemes for evaluating NC^1 functions



Transparent setup

DDH

DCR

Transparent setup

DDH over class groups

HSS Schemes from Prior Works

(Require Correlated Setup)

[Boyle-Gilboa-Ishai'16]

[Orlandi-Scholl-Yakoubov'21]
[Roy-Singh'21]

One-round protocol for generating correlated setup

[Abram-Damgård-Orlandi-Scholl'22]

Previously known only from LWE and $i\mathcal{O}$ + DDH [Dodis-Halevi-Rothblum-Wichs'16]

Our Results: Applications of Multi-key HSS

Two-round sublinear 2PC for NC^1 circuits in the CRS model

DDH

DCR

DDH over class groups

Previously known only from multi-key FHE [Mukherjee-Wichs'16]

Our Results: Applications of Multi-key HSS

Two-round sublinear 2PC for NC^1 circuits in the CRS model

DDH

DCR

DDH over class groups

Previously from group-based assumptions

3 round protocol in the CRS model

Previously known only from multi-key FHE [Mukherjee-Wichs'16]

Our Results: Applications of Multi-key HSS

Two-round sublinear 2PC for NC^1 circuits in the CRS model

DDH

DCR

DDH over class groups

Previously from group-based assumptions

3 round protocol in the CRS model

Previously known only from multi-key FHE [Mukherjee-Wichs'16]

Attribute-based NIKE supporting NC^1 predicates

DCR

DDH over class groups

Our Results: Applications of Multi-key HSS

Public-key PCFs for NC^1 additive correlations

DCR

DDH over class groups

Our Results: Applications of Multi-key HSS

Public-key PCFs for NC^1 additive correlations

Includes Beaver triples, correlated OT, OLE etc.,

DCR

DDH over class groups

Our Results: Applications of Multi-key HSS

Public-key PCFs for NC^1 additive correlations

DCR

DDH over class groups

Previously from group-based assumptions

Public-key PCFs for OT and Vector-OLE correlations

Our Results: Applications of Multi-key HSS

Public-key PCFs for NC^1 additive correlations

DCR

DDH over class groups

Previously from group-based assumptions

Public-key PCFs for OT and Vector-OLE correlations

n-party secure computation protocol in the preprocessing model with communication complexity

- Offline phase: poly$(\lambda) \cdot n$
- Online phase: $O(|C| \cdot n)$

DCR

DDH over class groups

Previously from group-based assumptions

Offline communication complexity poly$(\lambda) \cdot n^2$

Outline

Applications

Our Results

Constructing Multi-Key HSS

Group-Based HSS Schemes

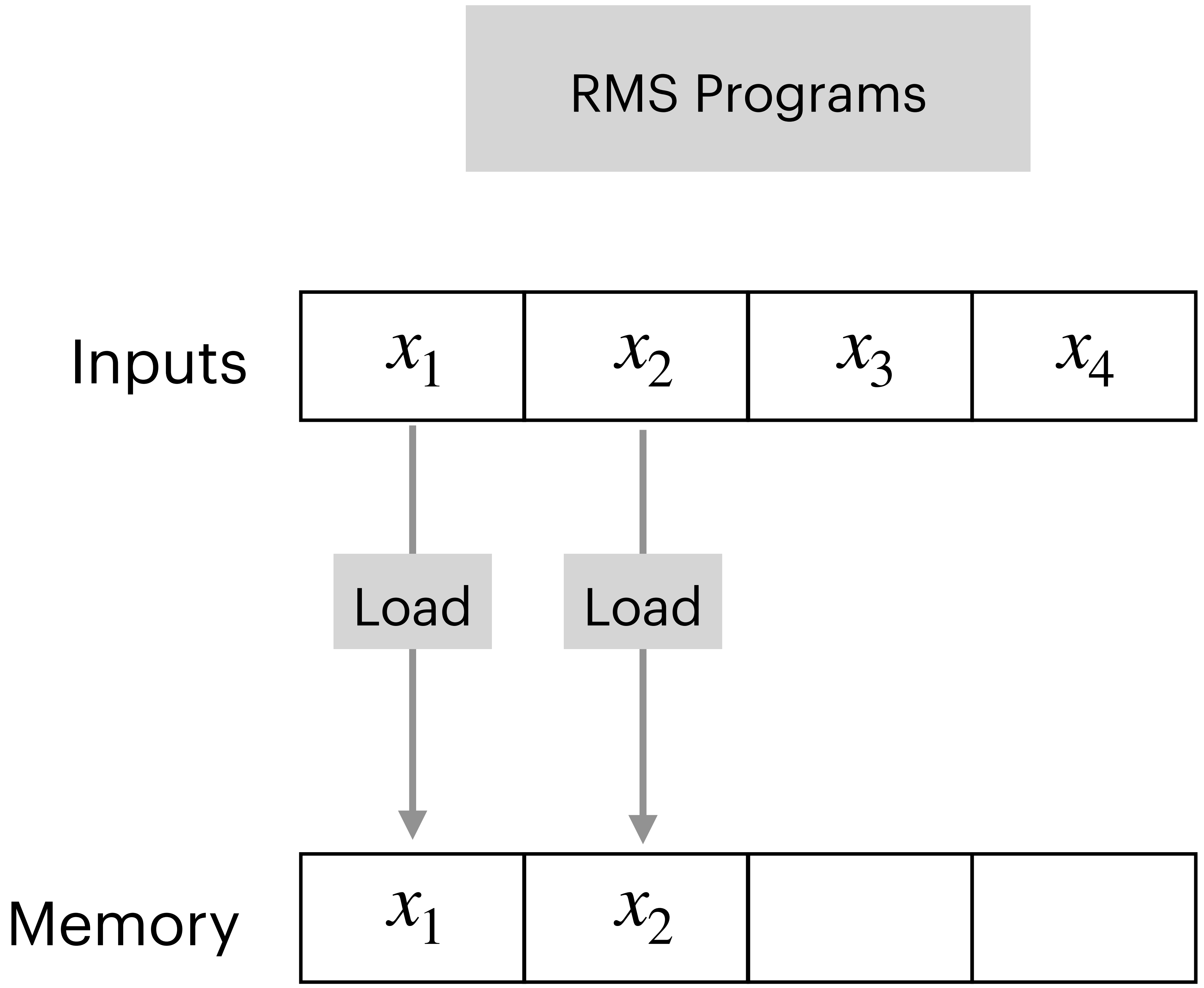
[Boyle-Gilboa-Ishai'16]

RMS Programs

Inputs	x_1	x_2	x_3	x_4
Memory				

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai'16]



RMS Programs

Inputs x_1 x_2 x_3 x_4

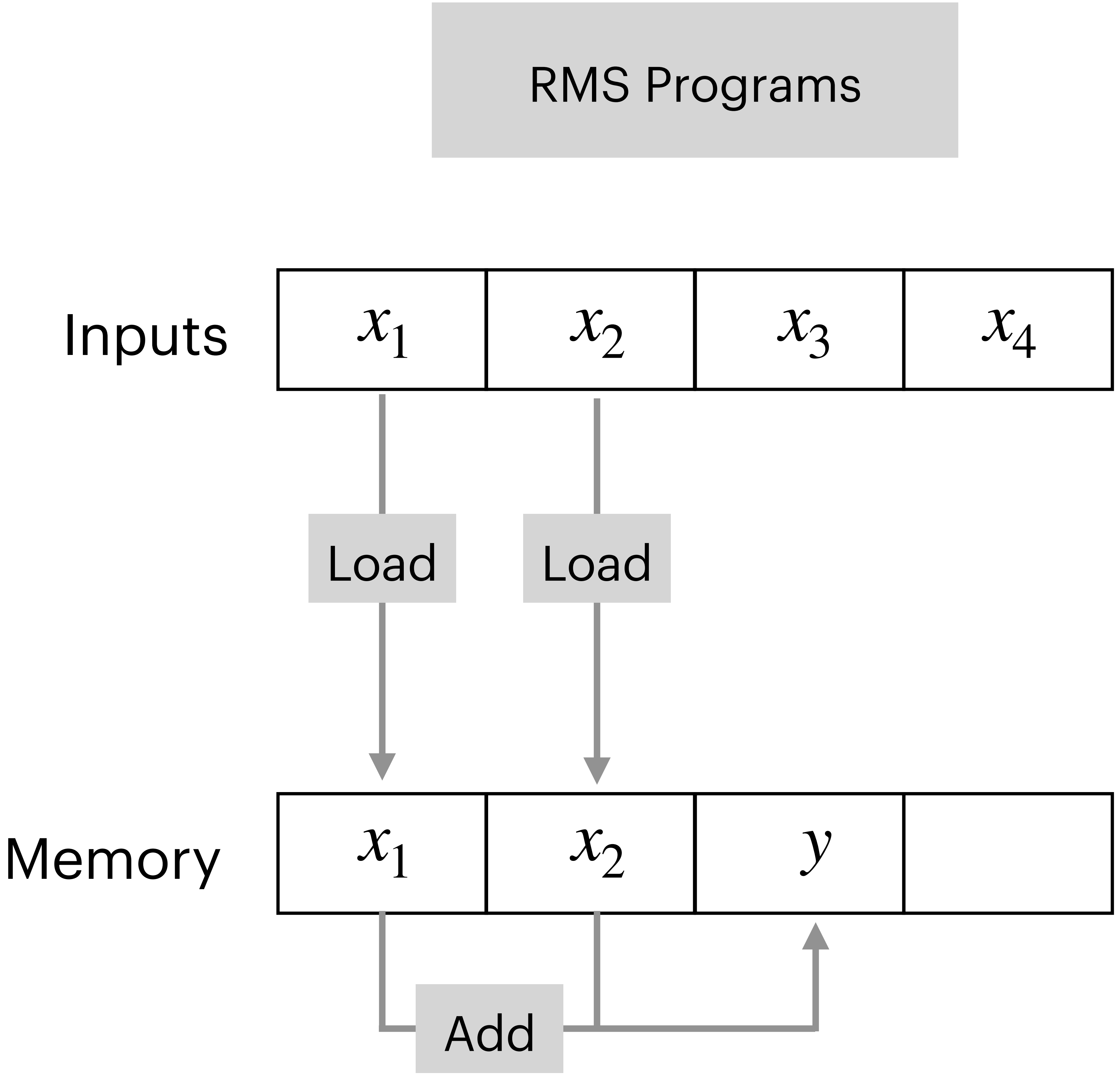
Load

Load

Memory x_1 x_2

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

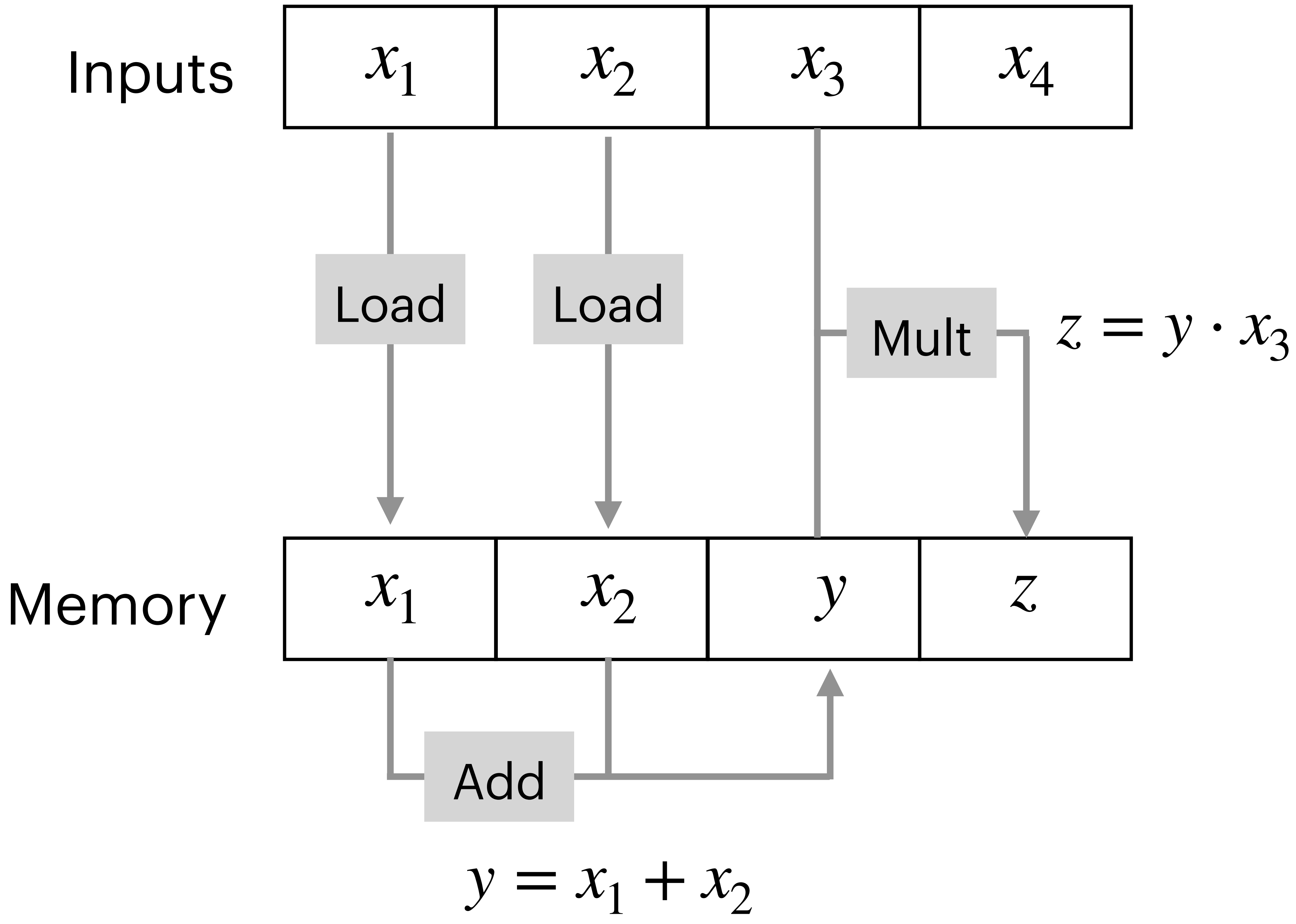


$$y = x_1 + x_2$$

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

RMS Programs



Inputs

x_1 x_2 x_3 x_4

Load

Load

Mult

$z = y \cdot x_3$

Memory

x_1 x_2 y z

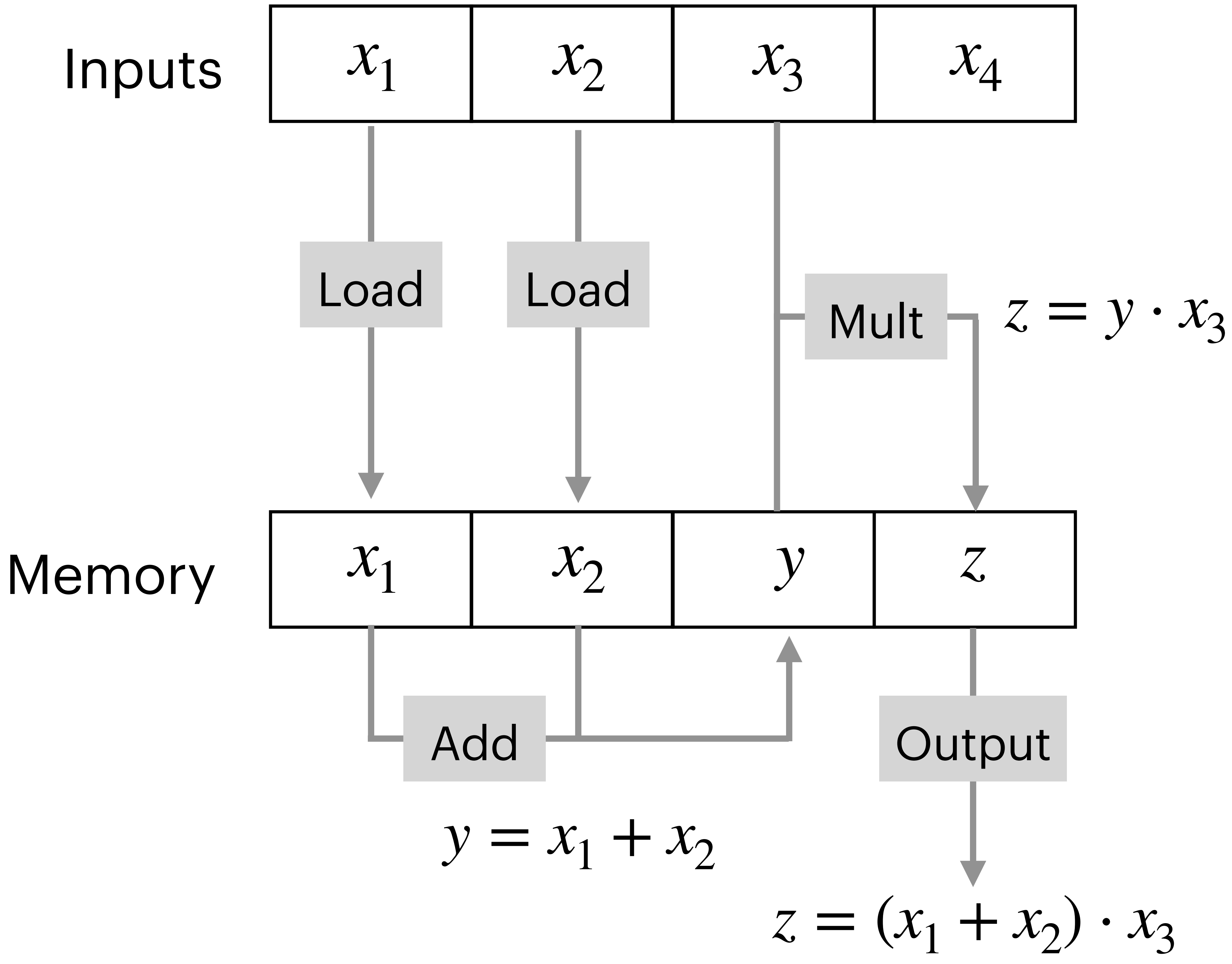
Add

$y = x_1 + x_2$

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

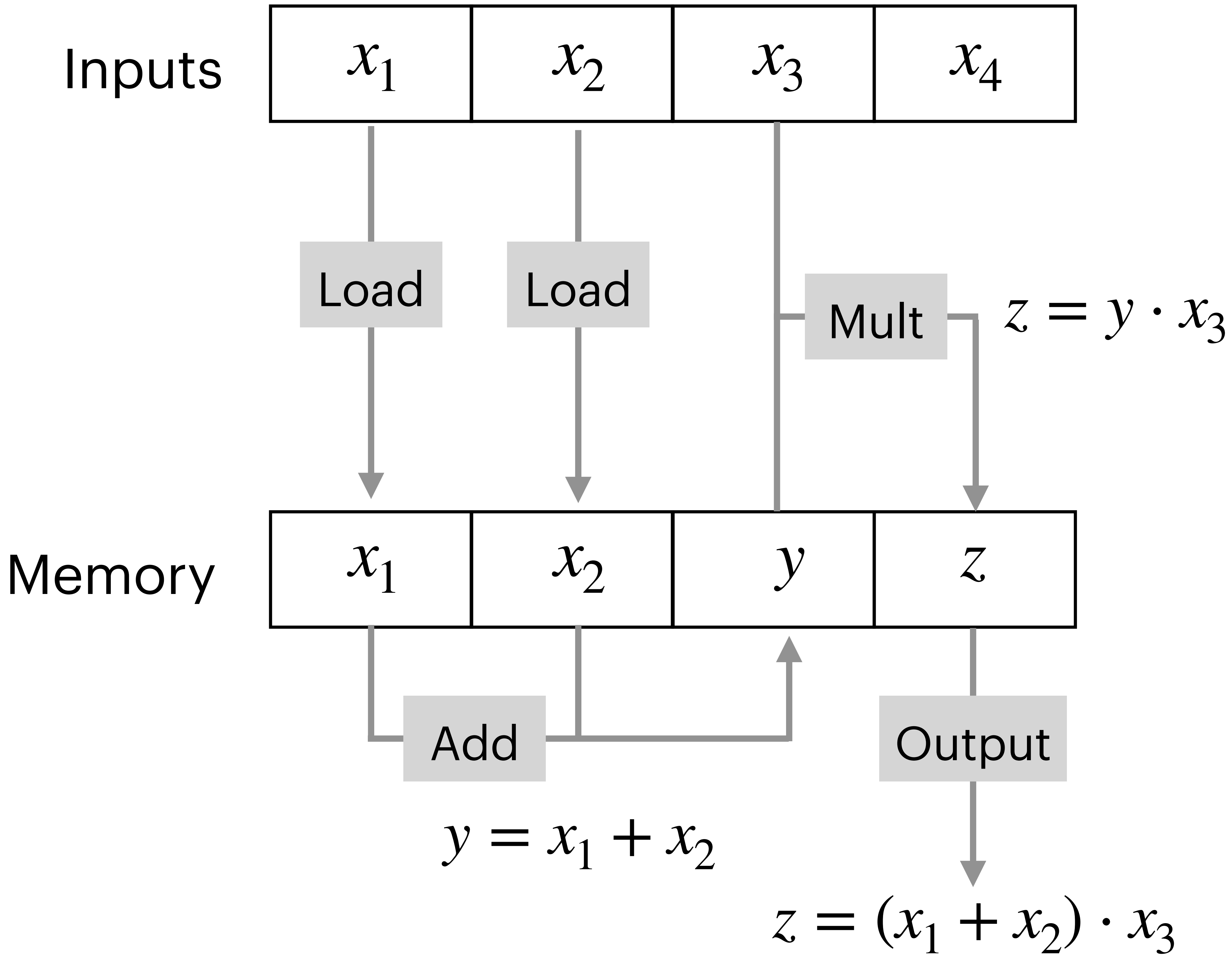
RMS Programs



Group-Based HSS Schemes

[Boyle-Gilboa-Ishai'16]

RMS Programs



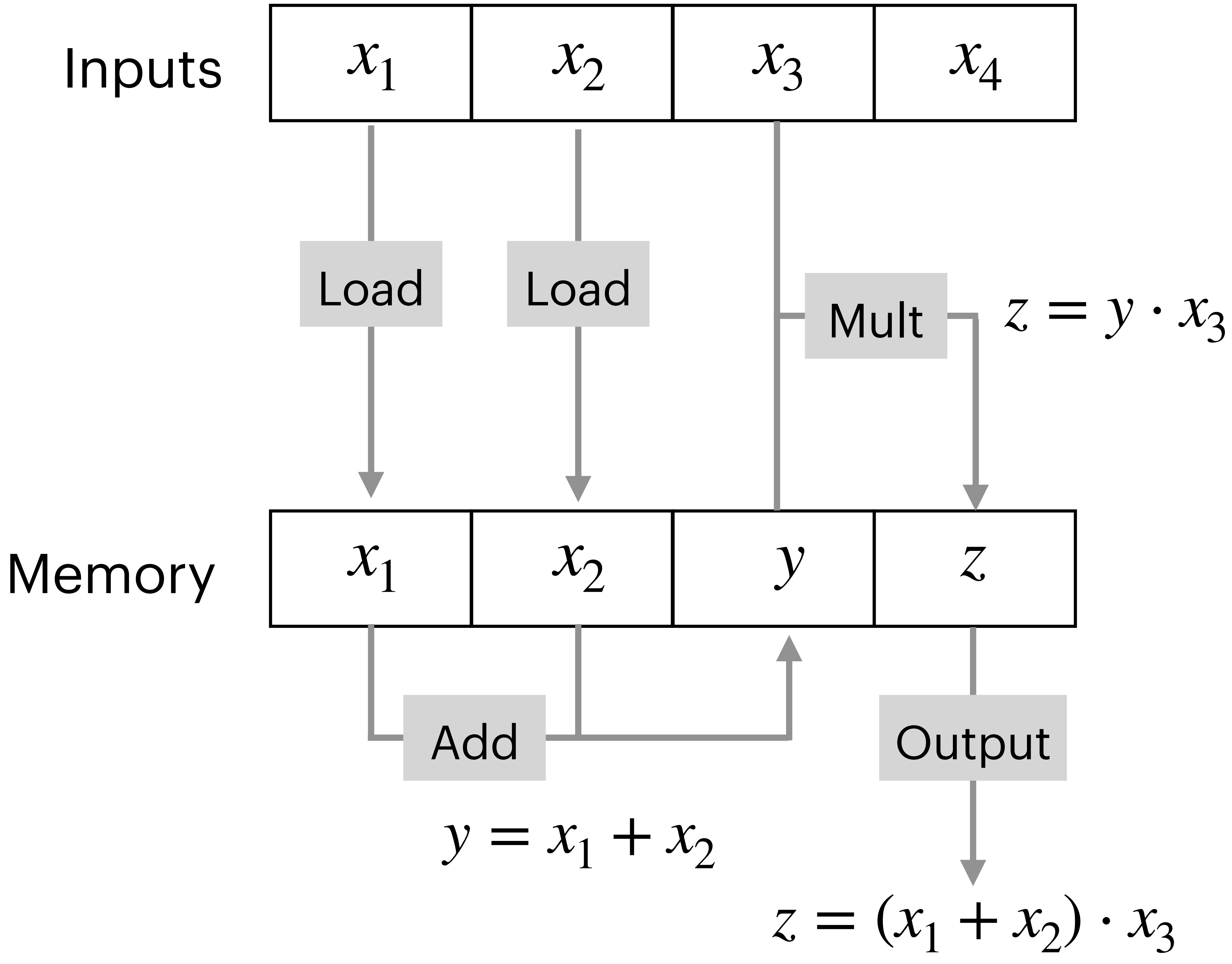
$\mathrm{NC}^1 \subseteq$ RMS Programs

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

RMS Programs

HSS Evaluation



Inputs x_1 x_2 x_3 x_4

Load

Load

Mult $z = y \cdot x_3$

Memory x_1 x_2 y z

Add

$y = x_1 + x_2$

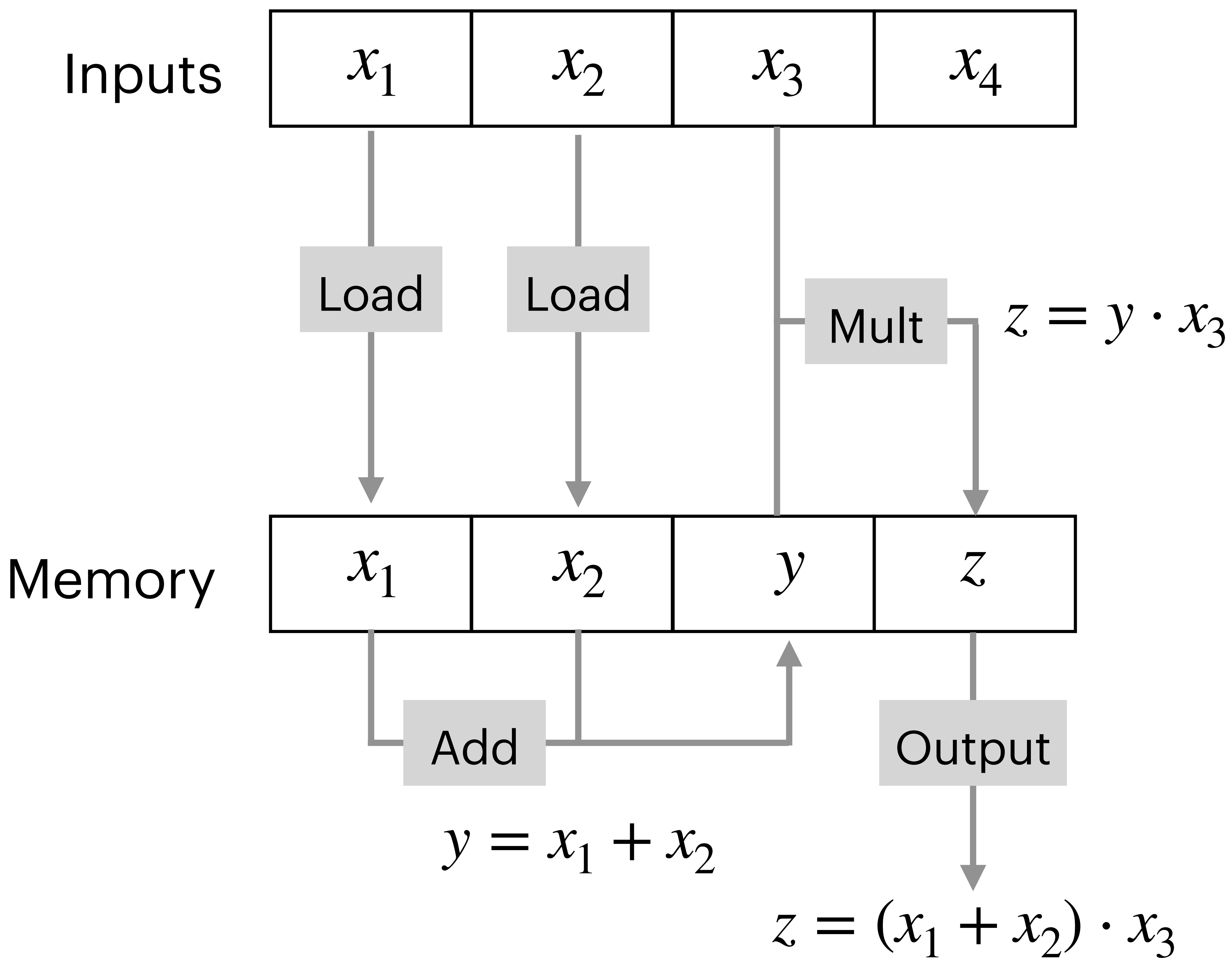
Output

$z = (x_1 + x_2) \cdot x_3$

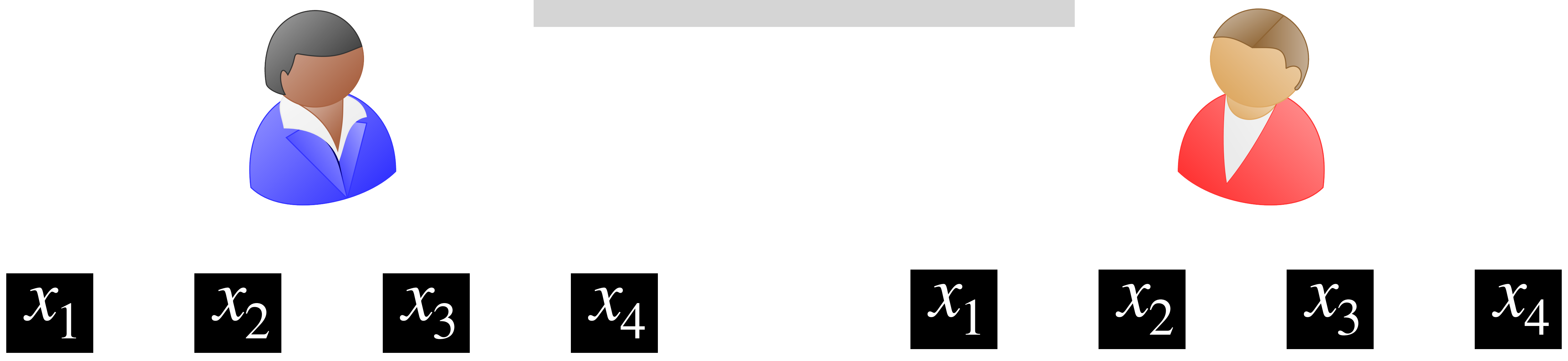
Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

RMS Programs



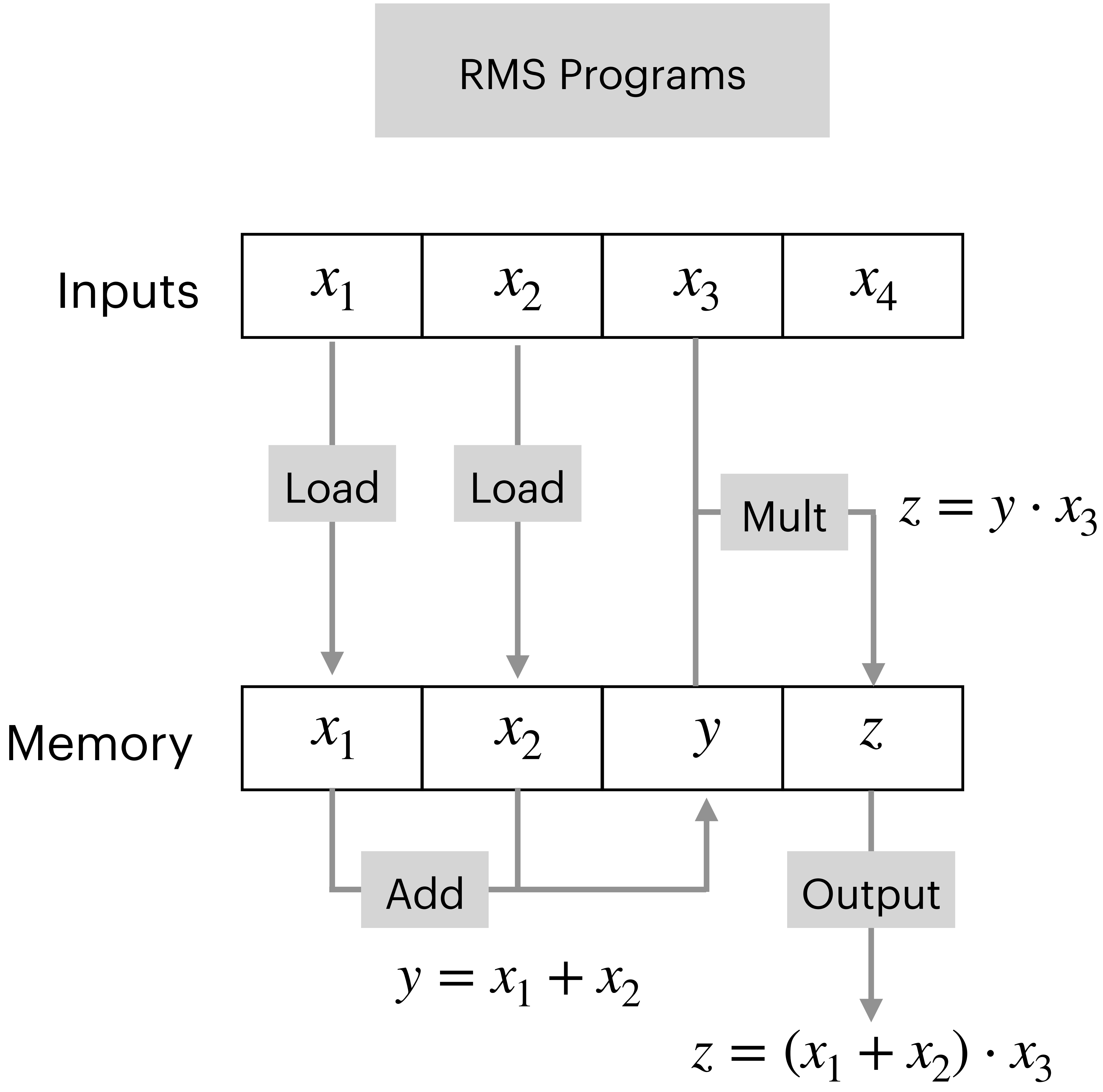
HSS Evaluation

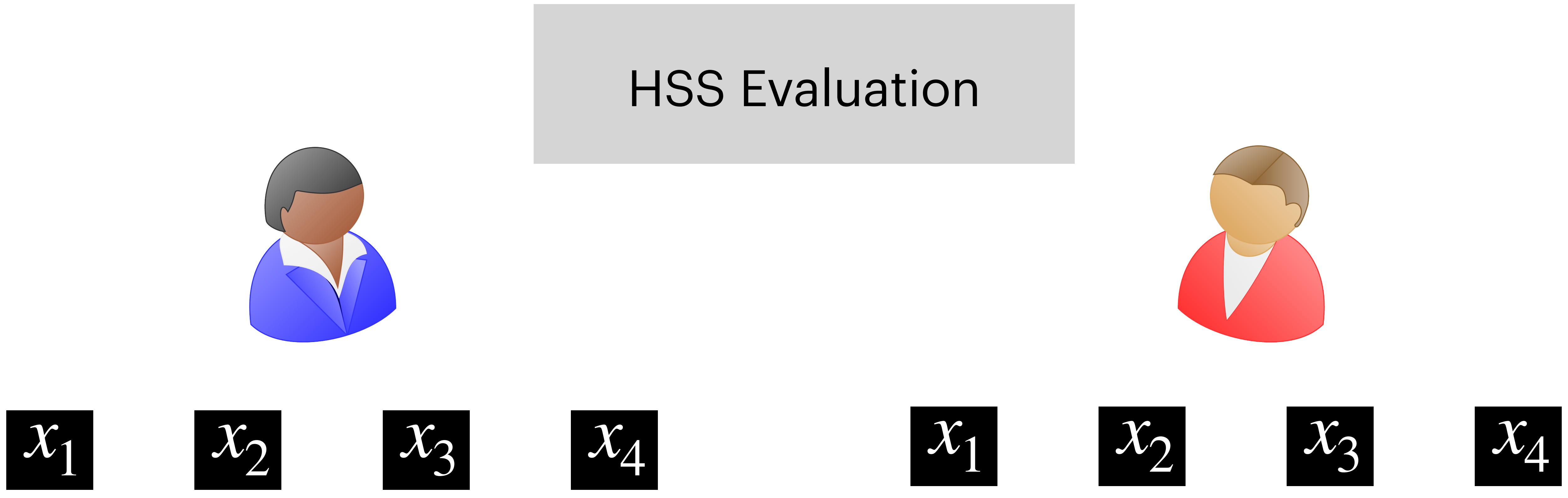


Inputs are encrypted

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai'16]



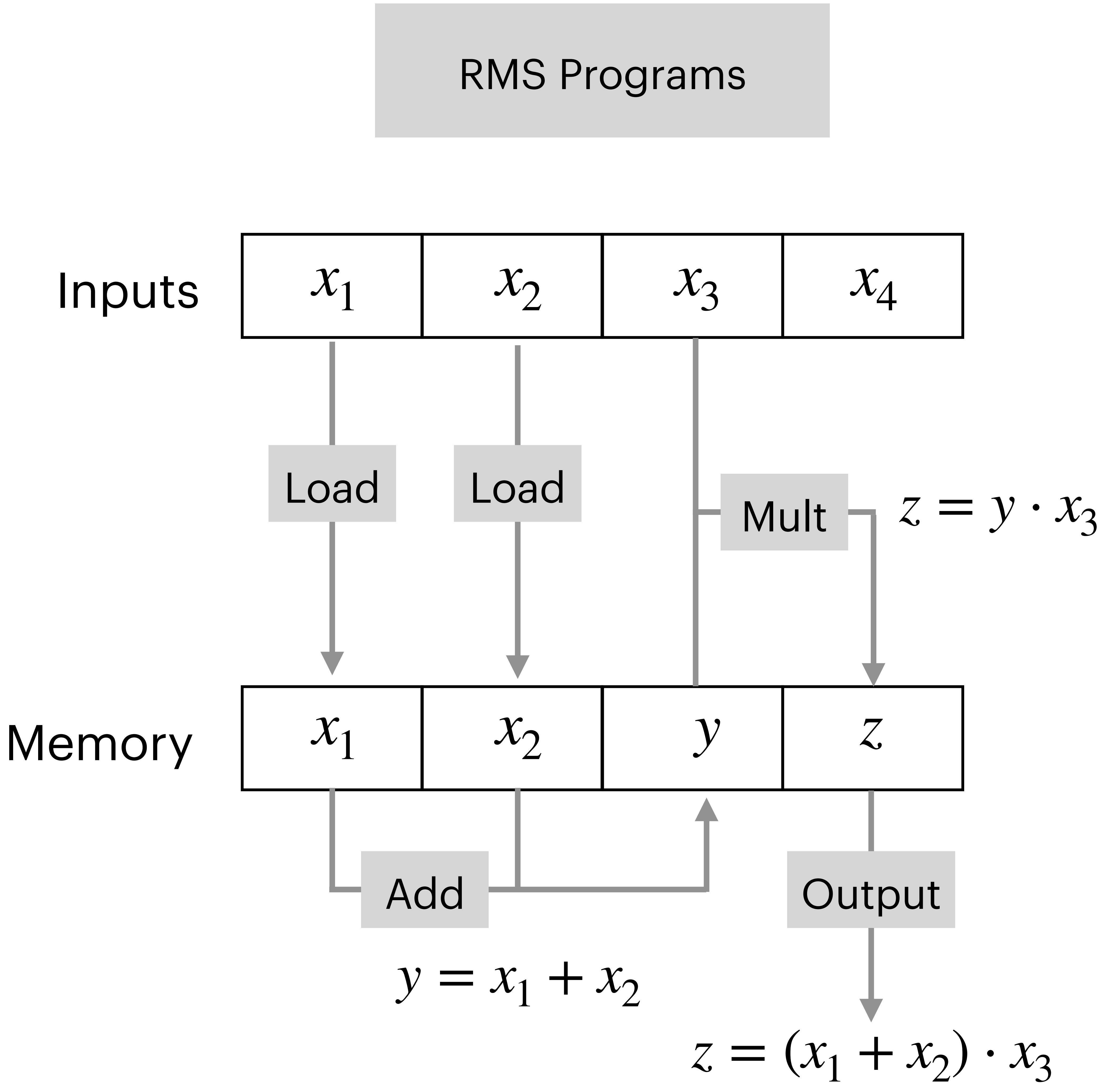


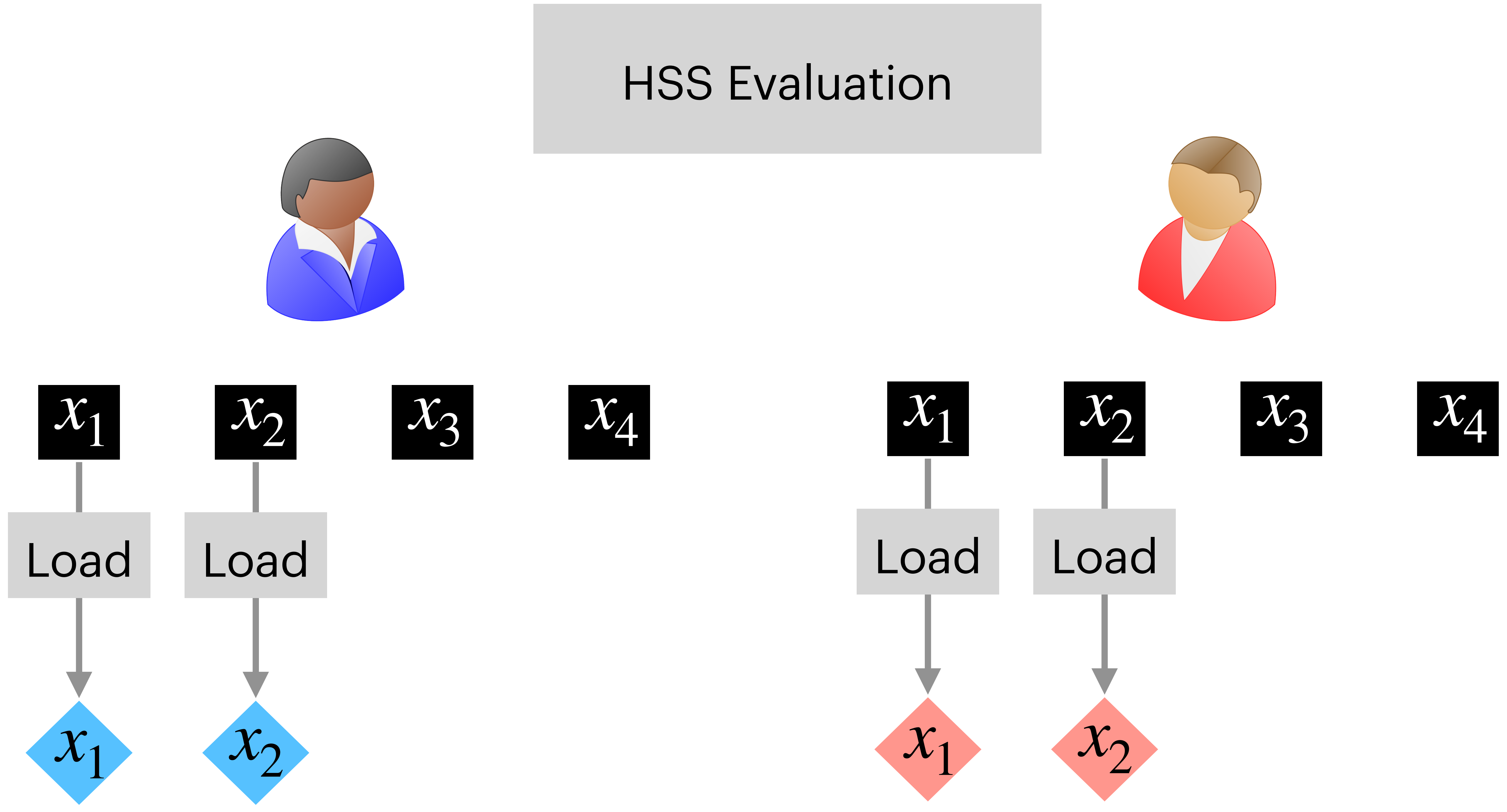
Inputs are encrypted

Evaluation Invariant: Sharing of each memory value

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai'16]



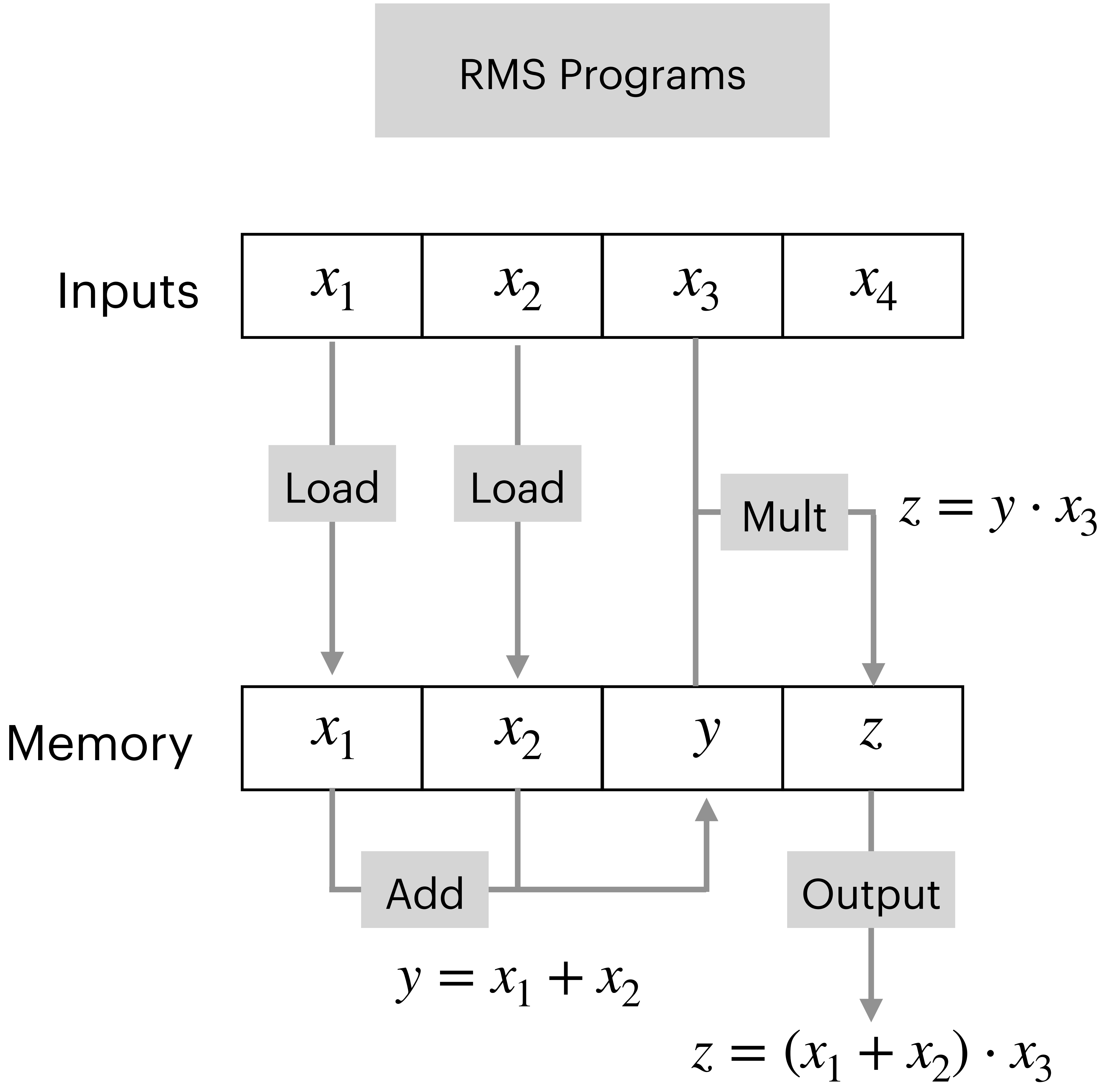


Inputs are encrypted

Evaluation Invariant: Sharing of each memory value

Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]



RMS Programs

Inputs

x_1 x_2 x_3 x_4

Load

Load

Mult

$z = y \cdot x_3$

Memory

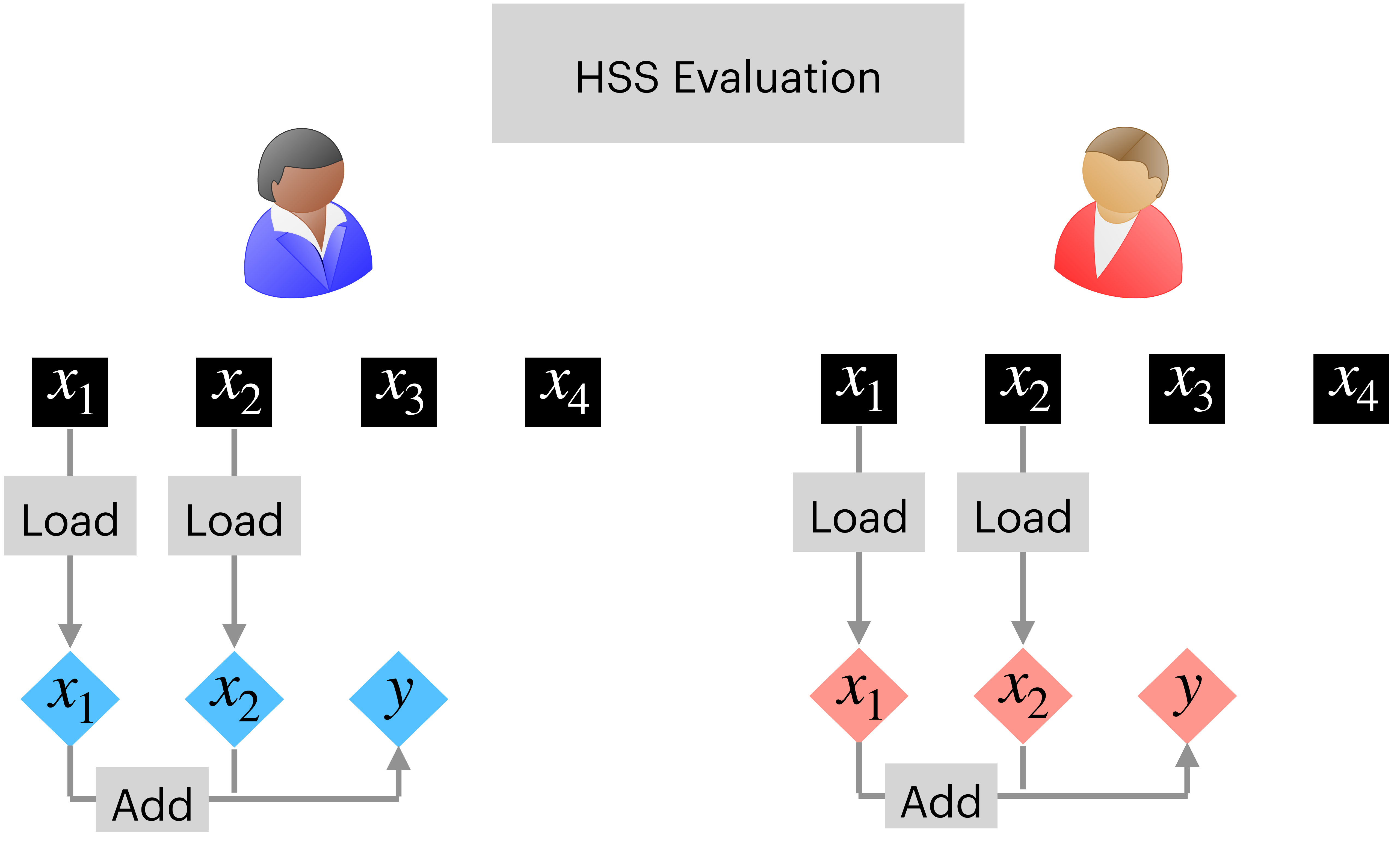
x_1 x_2 y z

Add

Output

$y = x_1 + x_2$

$z = (x_1 + x_2) \cdot x_3$



HSS Evaluation

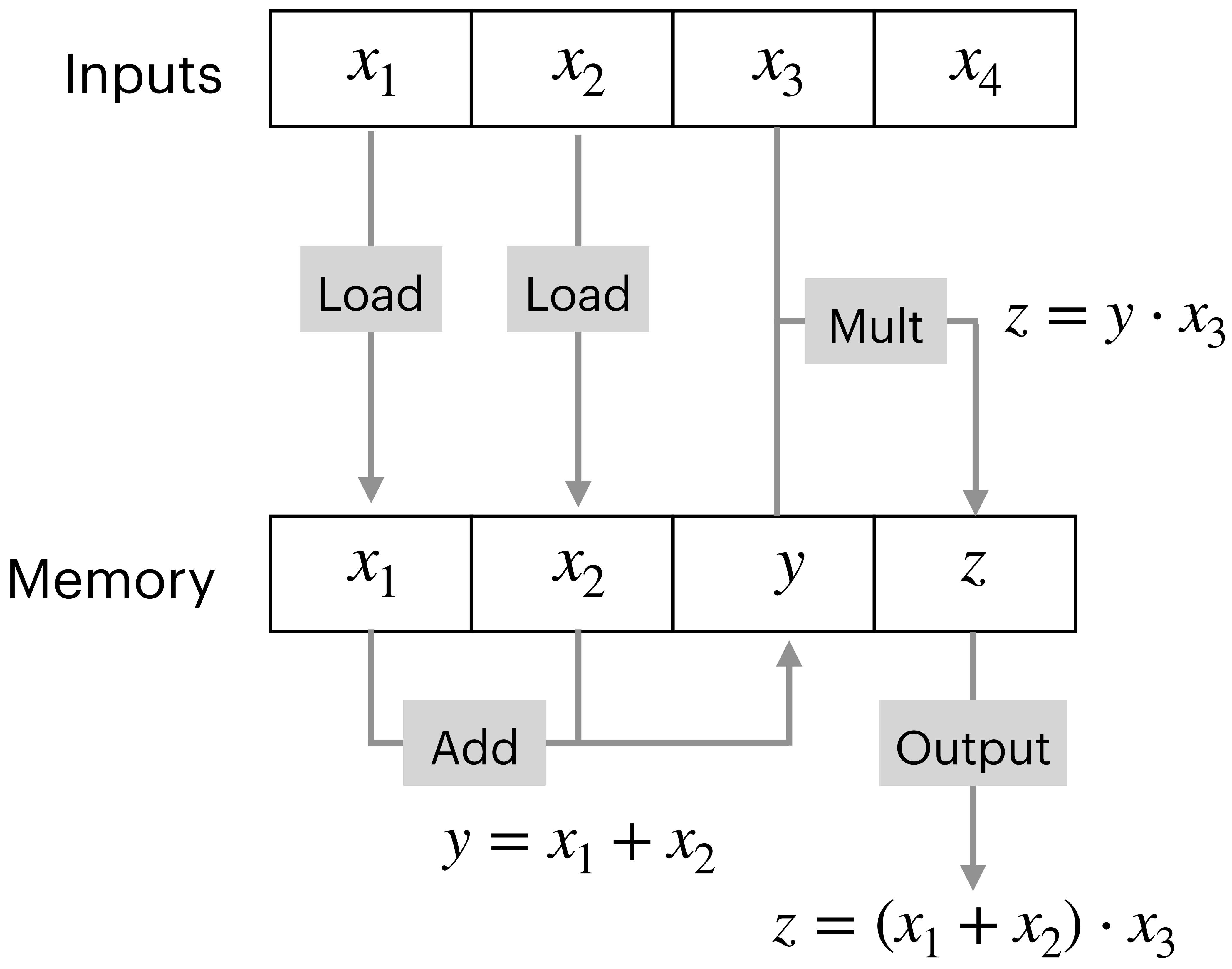
Inputs are encrypted

Evaluation Invariant: Sharing of each memory value

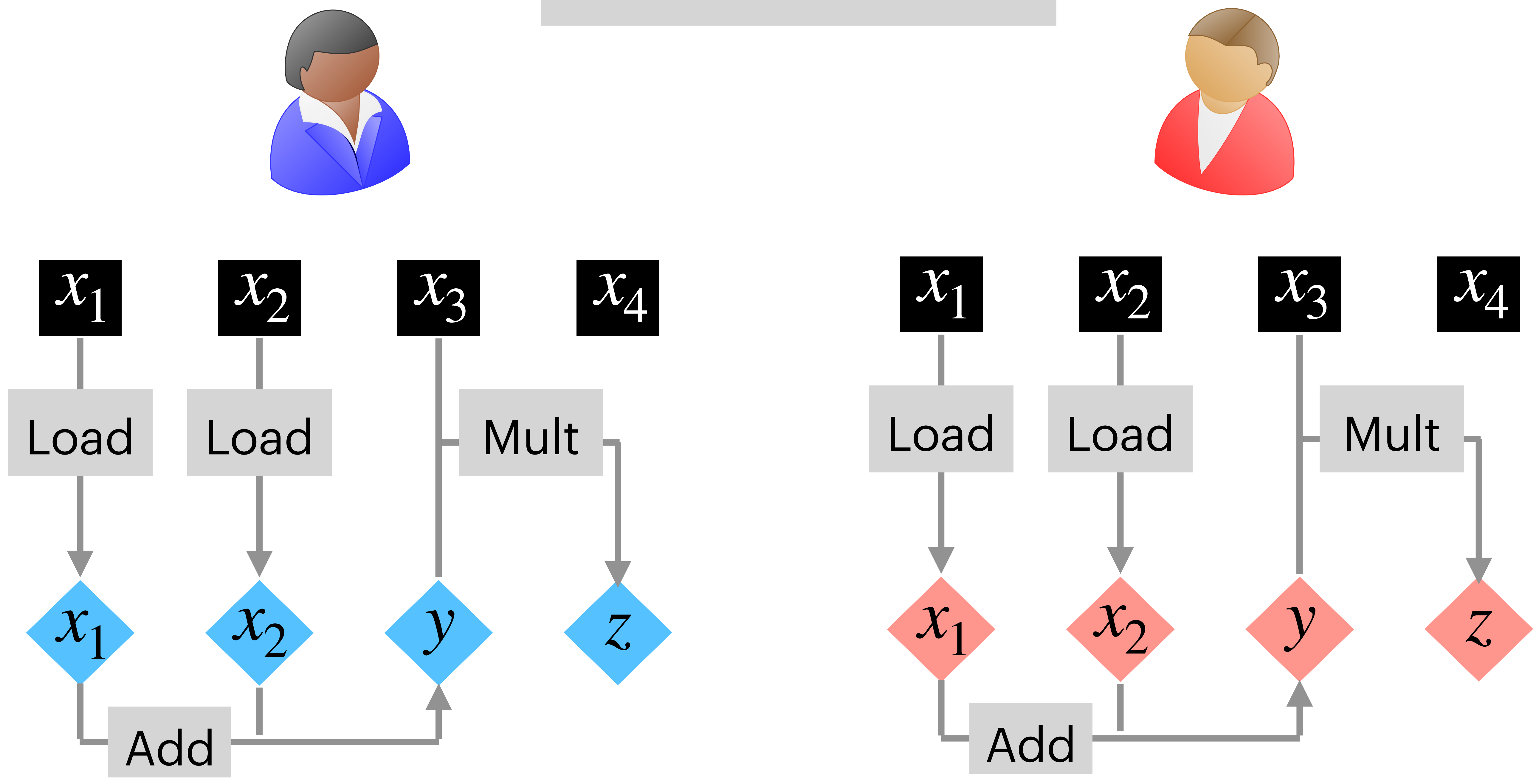
Group-Based HSS Schemes

[Boyle-Gilboa-Ishai’16]

RMS Programs



HSS Evaluation



Inputs are encrypted

Evaluation Invariant: Sharing of each memory value

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai’16]

Input Encryption

ElGamal public key in
correlated setup

$$\boxed{x} = \text{Enc}(\ \text{pk}\ , x\ )\ ,\ \text{Enc}(\ \text{pk}\ , \text{sk} \cdot x\ )$$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai’16]

Input Encryption

ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

$$\boxed{x} = \text{Enc}(\,\text{pk}\,, x\,)\,,\, \text{Enc}(\,\text{pk}\,,\, \text{sk} \cdot x\,)$$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai’16]

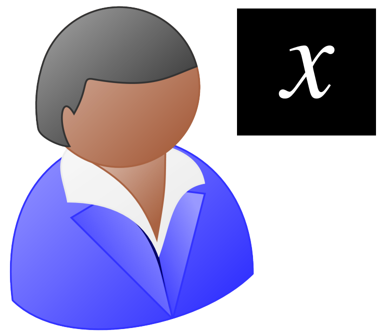
Input Encryption

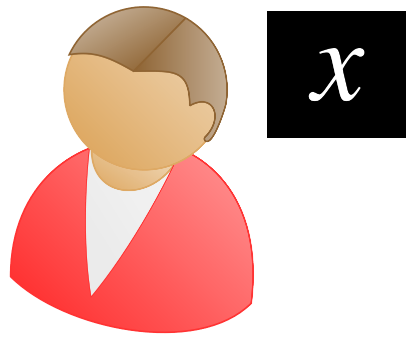
Memory Share

ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

$$\boxed{x} = \text{Enc}(\ \text{pk}\ , x\ )\ ,\ \text{Enc}(\ \text{pk}\ , \text{sk}\cdot x\ )$$





$$y = \boxed{y}\ ,\ \boxed{\text{sk}\cdot y}$$

$$y = \boxed{y}\ ,\ \boxed{\text{sk}\cdot y}$$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai'16]

Input Encryption

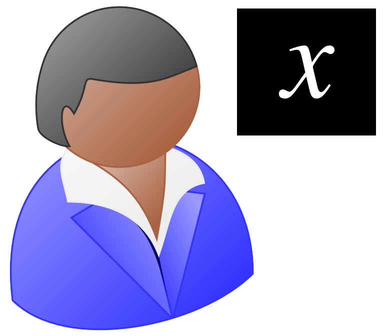
Memory Share

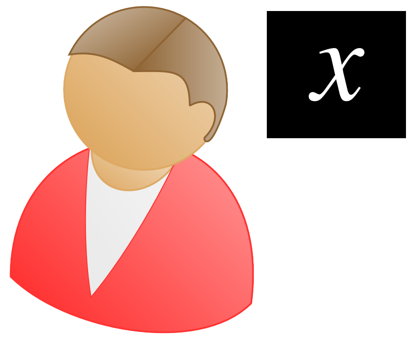
Multiplication

ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

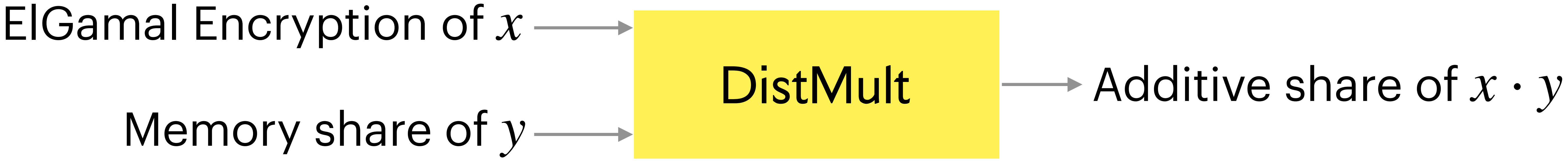
$$\boxed{x} = \text{Enc}(\text{ pk }, x\ ), \text{Enc}(\text{ pk }, \text{sk}\cdot x\ )$$





$$\diamond\, y = \boxed{y}\,, \boxed{\text{sk}\cdot y}$$

$$\diamond\, y = \boxed{y}\,, \boxed{\text{sk}\cdot y}$$



ElGamal Encryption of x

Memory share of y

DistMult

Additive share of $x \cdot y$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai'16]

Input Encryption

Memory Share

Multiplication

ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

$$\boxed{x} = \mathrm{Enc}(\,\mathrm{pk}\,,\, x\,)\,,\ \mathrm{Enc}(\,\mathrm{pk}\,,\, \mathrm{sk}\cdot x\,)$$

$\boxed{x}$

$\boxed{x}$

$$y = \boxed{y}\,,\ \boxed{\mathrm{sk}\cdot y}$$

$$y = \boxed{y}\,,\ \boxed{\mathrm{sk}\cdot y}$$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai’16]

Input Encryption

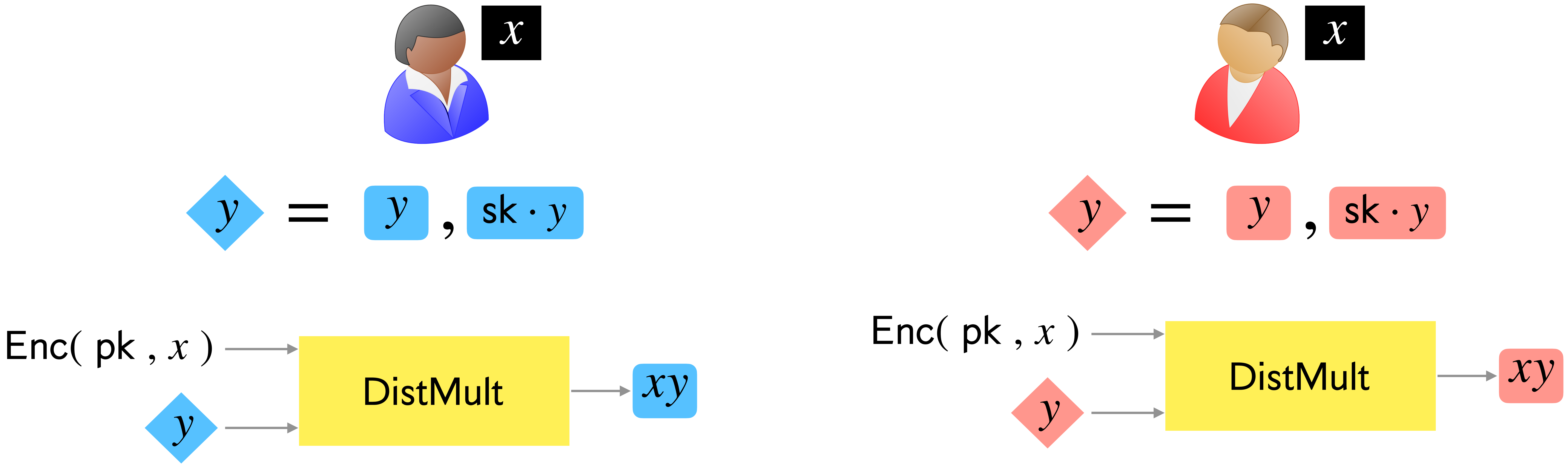
ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

$$\boxed{x} = \mathrm{Enc}(\,\mathrm{pk}\,,\,x\,)\,,\,\mathrm{Enc}(\,\mathrm{pk}\,,\,\mathrm{sk}\cdot x\,)$$

Memory Share

Multiplication



$$\boxed{x}$$

$$y = y\,,\,\mathrm{sk}\cdot y$$

$$\boxed{x}$$

$$y = y\,,\,\mathrm{sk}\cdot y$$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai'16]

Input Encryption

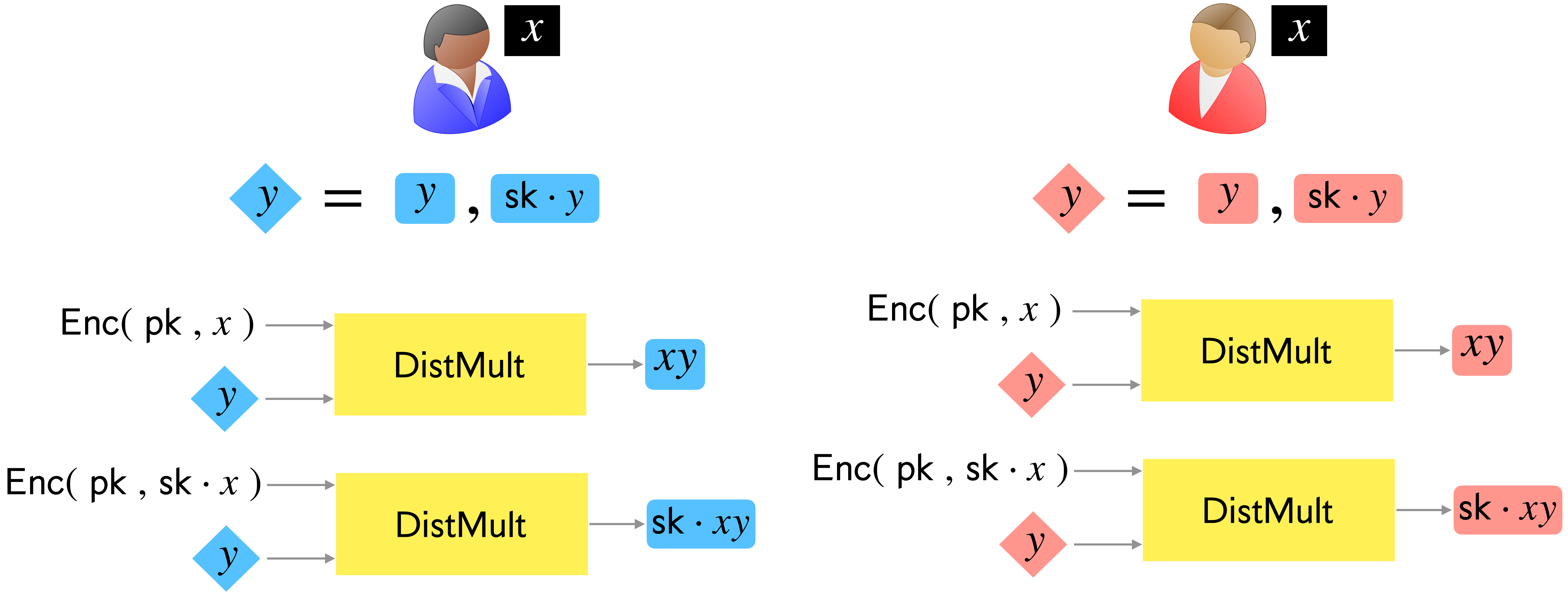
ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

$$\boxed{x} = \text{Enc}(\text{ pk }, x\ ), \text{Enc}(\text{ pk }, \text{sk} \cdot x\ )$$

Memory Share

Multiplication



$\boxed{x}$

$\boxed{x}$

$y = y\ , \text{sk} \cdot y$

$y = y\ , \text{sk} \cdot y$

$\text{Enc}(\text{ pk }, x\ )$, y → DistMult → xy

$\text{Enc}(\text{ pk }, \text{sk} \cdot x\ )$, y → DistMult → $\text{sk} \cdot xy$

$\text{Enc}(\text{ pk }, x\ )$, y → DistMult → xy

$\text{Enc}(\text{ pk }, \text{sk} \cdot x\ )$, y → DistMult → $\text{sk} \cdot xy$

Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai'16]

Input Encryption

ElGamal public key in correlated setup

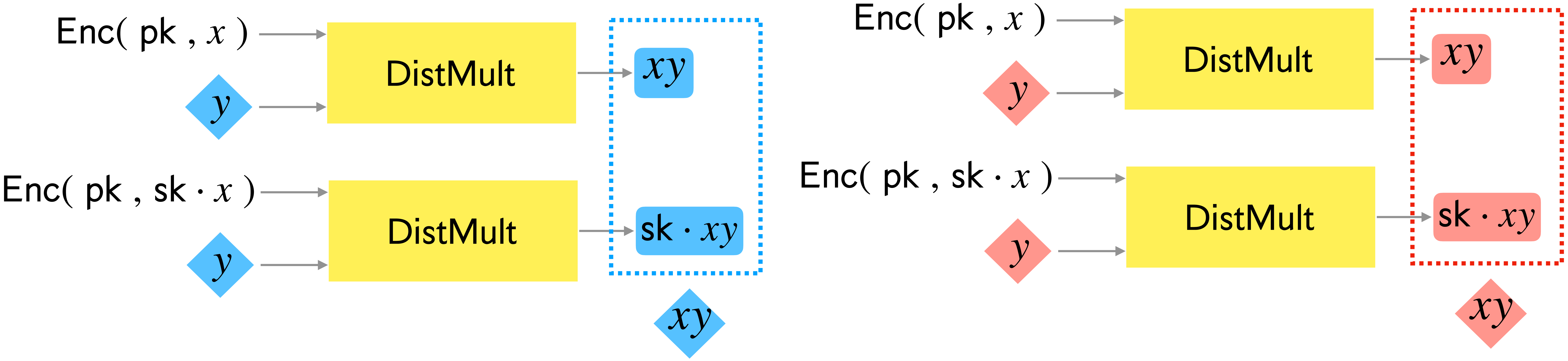
Can be computed using correlated setup without knowing sk

$$\boxed{x} = \text{Enc}(\,\text{pk}\,,\,x\,)\,,\,\text{Enc}(\,\text{pk}\,,\,\text{sk}\cdot x\,)$$

Memory Share

$$\diamond y = \boxed{y}\,,\,\boxed{\text{sk}\cdot y} \qquad \diamond y = \boxed{y}\,,\,\boxed{\text{sk}\cdot y}$$

Multiplication



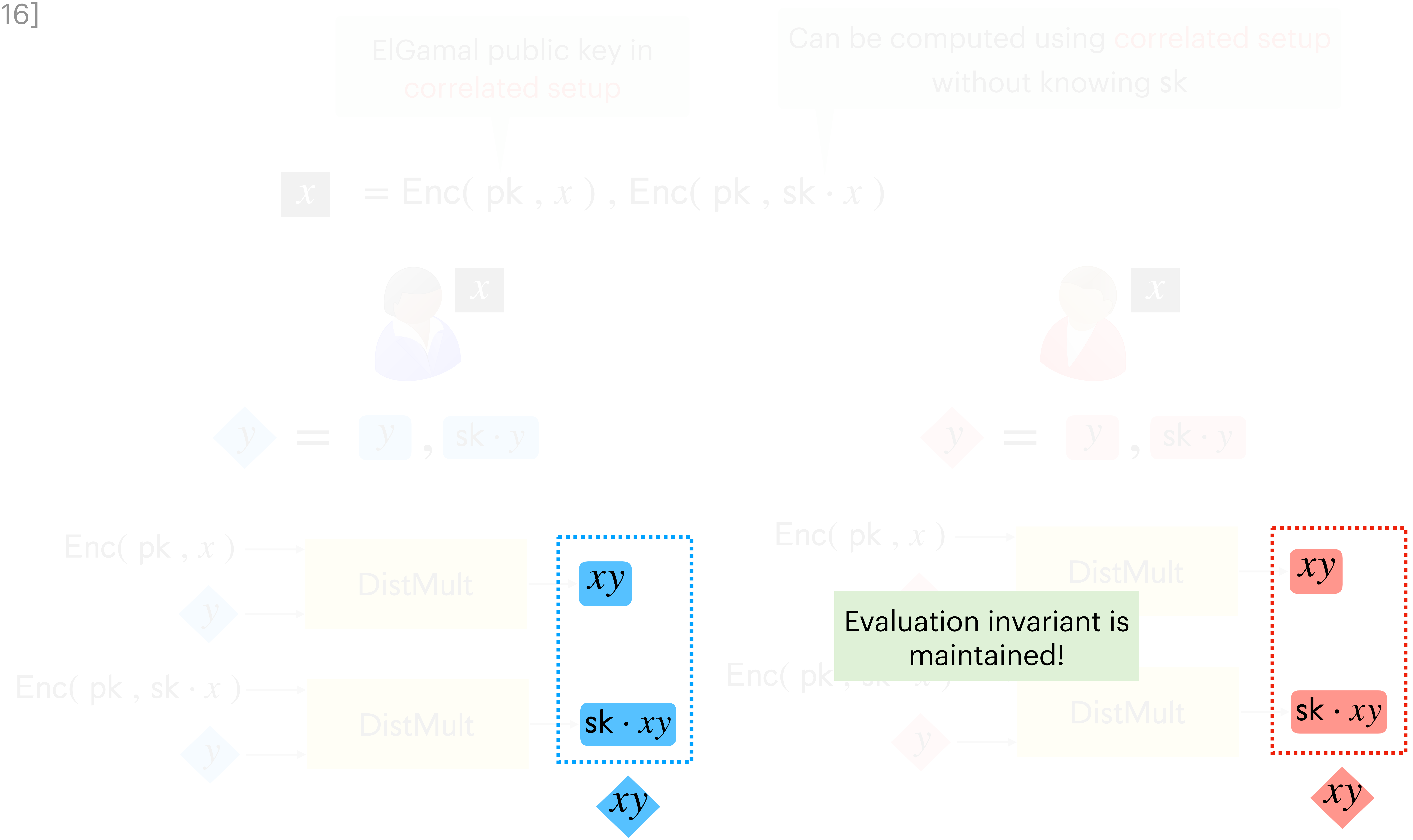
Group-Based HSS: Multiplication

[Boyle-Gilboa-Ishai’16]

Input Encryption

Memory Share

Multiplication



ElGamal public key in correlated setup

Can be computed using correlated setup without knowing sk

x = Enc(pk , x) , Enc(pk , sk · x)

Evaluation invariant is maintained!

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding

x

y

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

x

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

y

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding

$(\text{pk}_A, \text{sk}_A) \leftarrow \text{KeyGen}$

x

$(\text{pk}_B, \text{sk}_B) \leftarrow \text{KeyGen}$

y

$\boxed{x} = \text{Enc}(\text{pk}_A, x), \text{Enc}(\text{pk}_A, \text{sk}_A \cdot x)$

$\boxed{y} = \text{Enc}(\text{pk}_B, y), \text{Enc}(\text{pk}_B, \text{sk}_B \cdot y)$

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

x

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

y

$$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x), \mathsf{Enc}(\mathsf{pk}_A, \mathsf{sk}_A \cdot x)$$

$$\boxed{y} = \mathsf{Enc}(\mathsf{pk}_B, y), \mathsf{Enc}(\mathsf{pk}_B, \mathsf{sk}_B \cdot y)$$

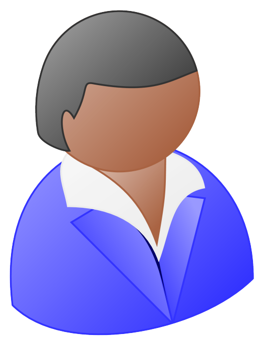
Memory Share

$$\Diamond z = \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$$

$$\Diamond z = \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$$

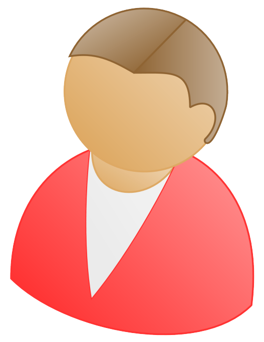
Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding



x

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$



y

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x), \mathsf{Enc}(\mathsf{pk}_A, \mathsf{sk}_A \cdot x)$

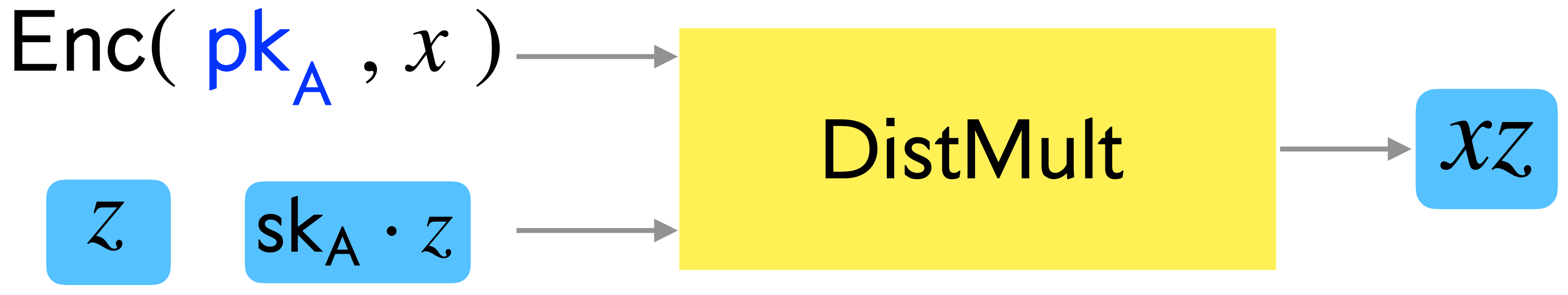
$\boxed{y} = \mathsf{Enc}(\mathsf{pk}_B, y), \mathsf{Enc}(\mathsf{pk}_B, \mathsf{sk}_B \cdot y)$

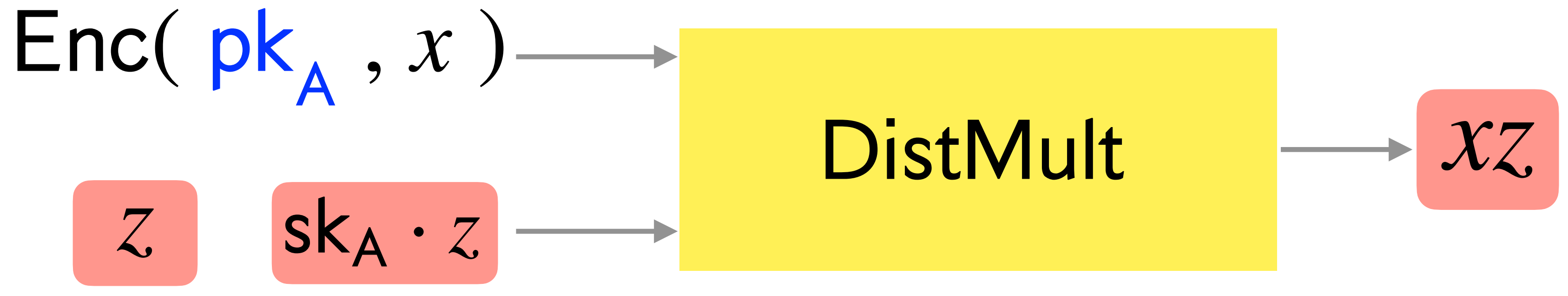
Memory Share

◆z $= \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$

◆z $= \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$

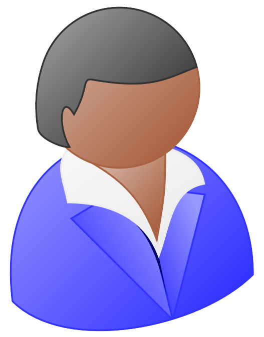
Multiplication





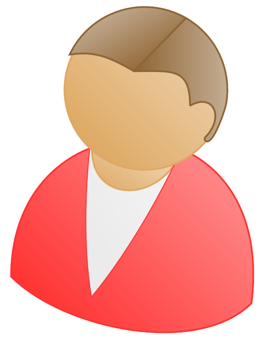
Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding



x

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$



y

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x), \mathsf{Enc}(\mathsf{pk}_A, \mathsf{sk}_A \cdot x)$

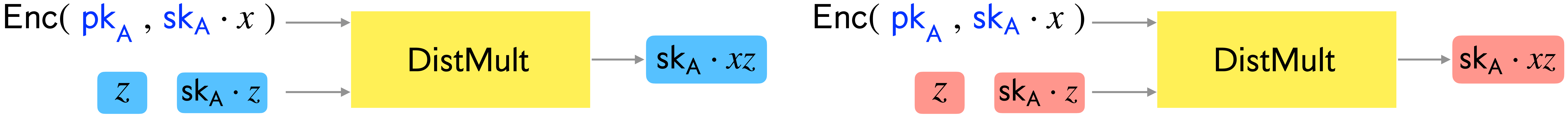
$\boxed{y} = \mathsf{Enc}(\mathsf{pk}_B, y), \mathsf{Enc}(\mathsf{pk}_B, \mathsf{sk}_B \cdot y)$

Memory Share

◆z $= \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$

◆z $= \boxed{z}, \boxed{\mathsf{sk}_A \cdot z}, \boxed{\mathsf{sk}_B \cdot z}$

Multiplication



$\mathsf{Enc}(\mathsf{pk}_A, \mathsf{sk}_A \cdot x)$

$\boxed{z}$ $\boxed{\mathsf{sk}_A \cdot z}$

DistMult

$\boxed{\mathsf{sk}_A \cdot xz}$

$\mathsf{Enc}(\mathsf{pk}_A, \mathsf{sk}_A \cdot x)$

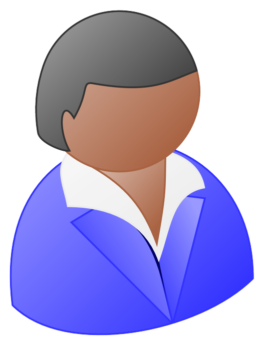
$\boxed{z}$ $\boxed{\mathsf{sk}_A \cdot z}$

DistMult

$\boxed{\mathsf{sk}_A \cdot xz}$

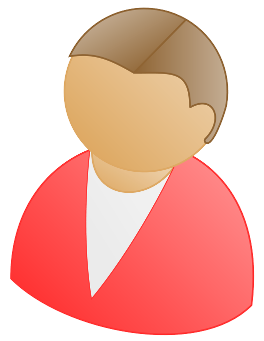
Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding



x

(pk_A , sk_A) ← KeyGen



y

(pk_B , sk_B) ← KeyGen

$\boxed{x}$ = Enc(pk_A , x) , Enc(pk_A , $\mathsf{sk}_A \cdot x$)

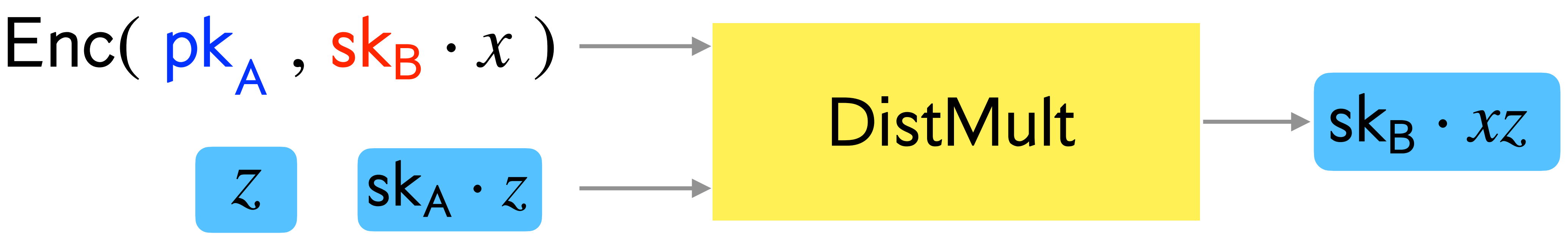
$\boxed{y}$ = Enc(pk_B , y) , Enc(pk_B , $\mathsf{sk}_B \cdot y$)

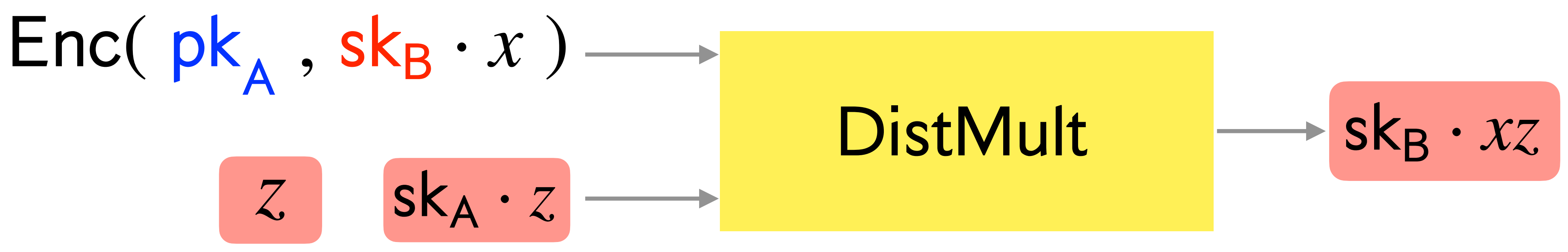
Memory Share

$\boxed{z} = \boxed{z}\,,\ \boxed{\mathsf{sk}_A \cdot z},\ \boxed{\mathsf{sk}_B \cdot z}$

$\boxed{z} = \boxed{z}\,,\ \boxed{\mathsf{sk}_A \cdot z},\ \boxed{\mathsf{sk}_B \cdot z}$

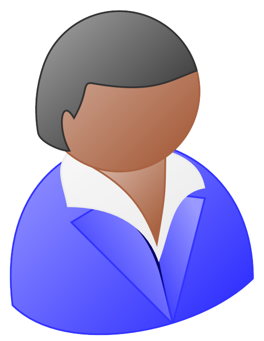
Multiplication





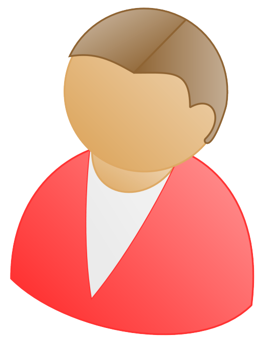
Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding



x

$(\text{pk}_A, \text{sk}_A) \leftarrow \text{KeyGen}$



y

$(\text{pk}_B, \text{sk}_B) \leftarrow \text{KeyGen}$

$\boxed{x} = \text{Enc}(\text{pk}_A, x), \text{Enc}(\text{pk}_A, \text{sk}_A \cdot x)$

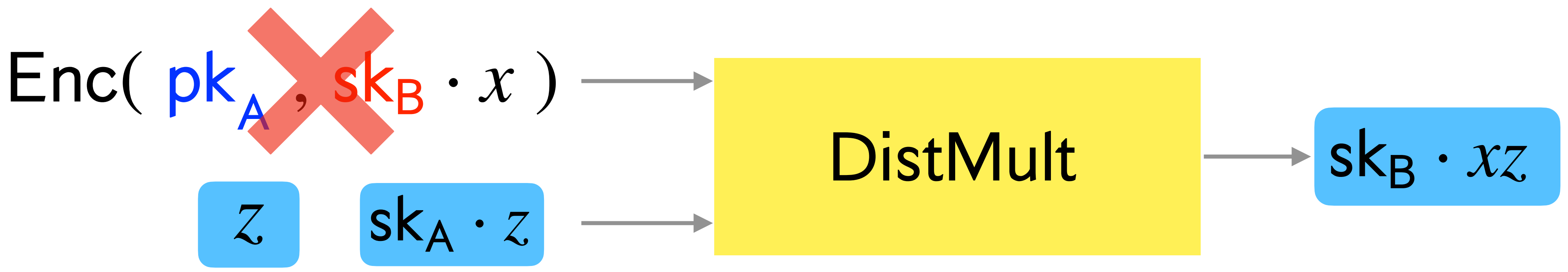
$\boxed{y} = \text{Enc}(\text{pk}_B, y), \text{Enc}(\text{pk}_B, \text{sk}_B \cdot y)$

Memory Share

◆ z $=$ z , $\text{sk}_A \cdot z$, $\text{sk}_B \cdot z$

◆ z $=$ z , $\text{sk}_A \cdot z$, $\text{sk}_B \cdot z$

Multiplication

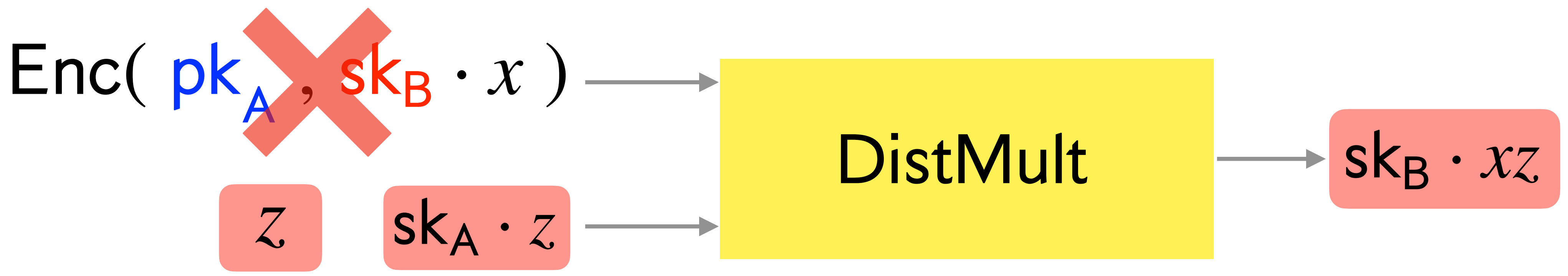


$\text{Enc}(\text{pk}_A \cancel{, \text{sk}_B} \cdot x)$

z $\text{sk}_A \cdot z$

DistMult

$\text{sk}_B \cdot xz$



$\text{Enc}(\text{pk}_A \cancel{, \text{sk}_B} \cdot x)$

z $\text{sk}_A \cdot z$

DistMult

$\text{sk}_B \cdot xz$

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding

$(\, pk_A \,,\, sk_A \,) \leftarrow$ KeyGen

x

$(\, pk_B \,,\, sk_B \,) \leftarrow$ KeyGen

y

$\boxed{x} = \mathrm{Enc}(\, pk_A \,,\, x \,)\,,\, \mathrm{Enc}(\, pk_A \,,\, sk_A \cdot x \,)$

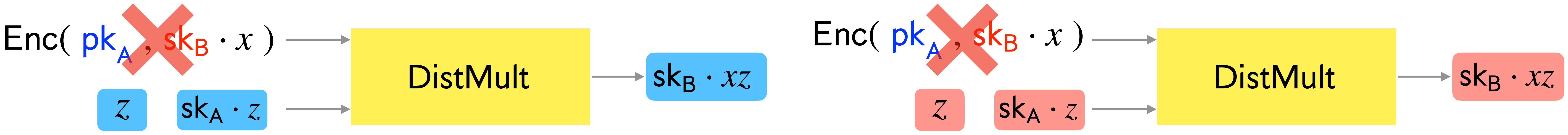
$\boxed{y} = \mathrm{Enc}(\, pk_B \,,\, y \,)\,,\, \mathrm{Enc}(\, pk_B \,,\, sk_B \cdot y \,)$

Memory Share

◆ $= \boxed{z}\,,\, \boxed{sk_A \cdot z}\,,\, \boxed{sk_B \cdot z}$

◆ $= \boxed{z}\,,\, \boxed{sk_A \cdot z}\,,\, \boxed{sk_B \cdot z}$

Multiplication



$\mathrm{Enc}(\, pk_A \,,\, sk_B \cdot x \,) \rightarrow$ DistMult

$z \quad sk_A \cdot z \rightarrow$ DistMult

DistMult $\rightarrow sk_B \cdot xz$

$\mathrm{Enc}(\, pk_A \,,\, sk_B \cdot x \,) \rightarrow$ DistMult

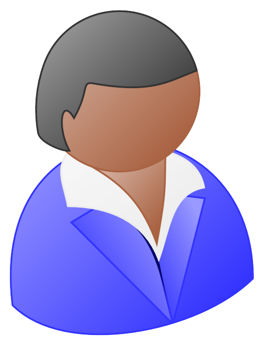
$z \quad sk_A \cdot z \rightarrow$ DistMult

DistMult $\rightarrow sk_B \cdot xz$

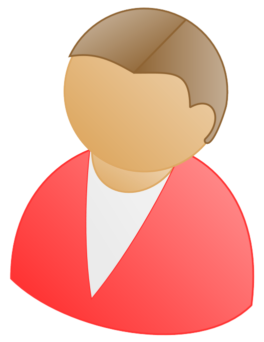
DistMult requires an encryption of $sk_B \cdot x$ to compute shares of $sk_B \cdot xz$

Constructing Multi-Key HSS: Removing Correlated Setup

Input Encoding


x

(pk_A , sk_A) ← KeyGen


y

(pk_B , sk_B) ← KeyGen

$\boxed{x} = \text{Enc}(\ \text{pk}_A\ , x\ )\ ,\ \text{Enc}(\ \text{pk}_A\ , \text{sk}_A \cdot x\ )$

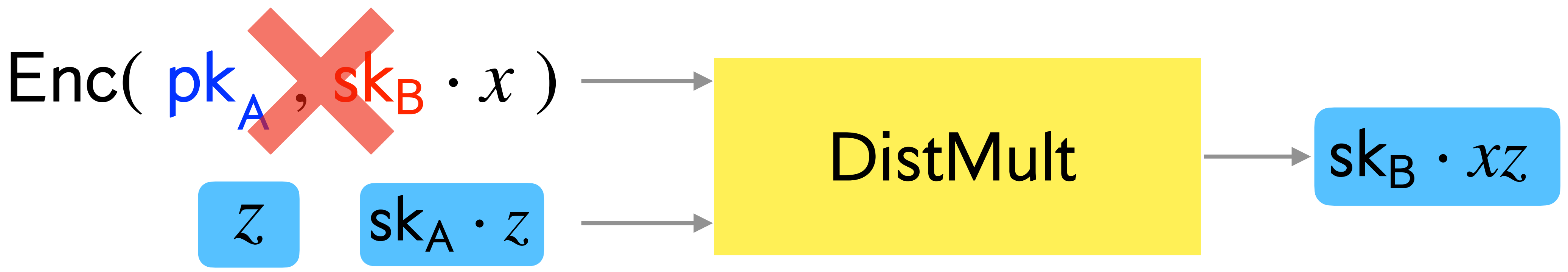
$\boxed{y} = \text{Enc}(\ \text{pk}_B\ , y\ )\ ,\ \text{Enc}(\ \text{pk}_B\ , \text{sk}_B \cdot y\ )$

Memory Share

◆z = z , $\text{sk}_A \cdot z$, $\text{sk}_B \cdot z$

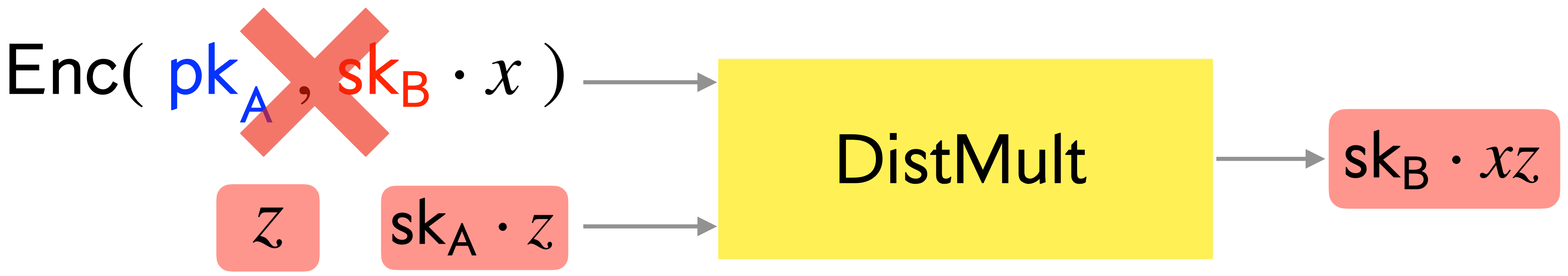
◆z = z , $\text{sk}_A \cdot z$, $\text{sk}_B \cdot z$

Multiplication


$\text{Enc}(\ \text{pk}_A \cancel{,\ \text{sk}_B} \cdot x\ )$ →

z $\quad$ $\text{sk}_A \cdot z$ →

DistMult → $\text{sk}_B \cdot xz$


$\text{Enc}(\ \text{pk}_A \cancel{,\ \text{sk}_B} \cdot x\ )$ →

z $\quad$ $\text{sk}_A \cdot z$ →

DistMult → $\text{sk}_B \cdot xz$

DistMult requires an encryption of $\text{sk}_B \cdot x$ to compute shares of $\text{sk}_B \cdot xz$

Shares of $\text{sk}_B \cdot xz$ are needed to multiply with Bob's input y

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x)$

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x)$

$\mathsf{Synchronize}(\mathsf{sk}_A, \mathsf{pk}_B, \boxed{x}) \rightarrow$

$\leftarrow \mathsf{Synchronize}(\mathsf{sk}_B, \mathsf{pk}_A, \boxed{x})$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\,\mathsf{pk}_A\,,\,\mathsf{sk}_A\,) \leftarrow \mathsf{KeyGen}$

$(\,\mathsf{pk}_B\,,\,\mathsf{sk}_B\,) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\,\mathsf{pk}_A\,,\,x\,)$

$\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,x\,)$

$\mathsf{Synchronize}(\,\mathsf{sk}_A\,,\,\mathsf{pk}_B\,,\,\boxed{x}\,) \rightarrow \quad \mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,\mathsf{sk}_A \cdot x\,) \quad \leftarrow \mathsf{Synchronize}(\,\mathsf{sk}_B\,,\,\mathsf{pk}_A\,,\,\boxed{x}\,)$

$\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,\mathsf{sk}_B \cdot x\,)$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\,\mathsf{pk}_A\,,\,\mathsf{sk}_A\,) \leftarrow \mathsf{KeyGen}$

$(\,\mathsf{pk}_B\,,\,\mathsf{sk}_B\,) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\,\mathsf{pk}_A\,,\,x\,)$

$\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,x\,)$

$\mathsf{Synchronize}(\,\mathsf{sk}_A\,,\,\mathsf{pk}_B\,,\,\boxed{x}\,) \rightarrow$ $\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,\mathsf{sk}_A \cdot x\,)$ $\leftarrow \mathsf{Synchronize}(\,\mathsf{sk}_B\,,\,\mathsf{pk}_A\,,\,\boxed{x}\,)$

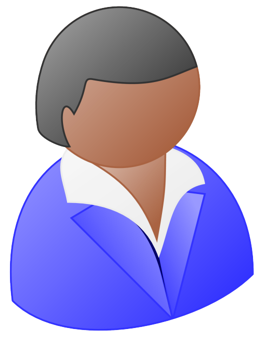
$\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,\mathsf{sk}_B \cdot x\,)$

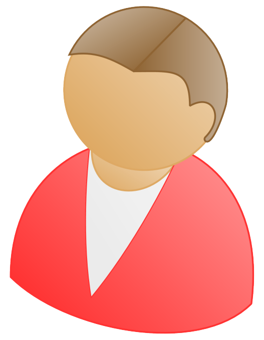
$$\mathsf{Enc}(\,\mathsf{pk}_A \| \mathsf{pk}_B\,,\,x\,) = \mathsf{Enc}(\,\mathsf{pk}_A\,,\,r\,)\,,\,\mathsf{Enc}(\,\mathsf{pk}_B\,,\,x - r\,)$$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\text{pk}_A , \text{sk}_A) \leftarrow \text{KeyGen}$

$\boxed{x} = \text{Enc}(\text{pk}_A , x)$





$(\text{pk}_B , \text{sk}_B) \leftarrow \text{KeyGen}$

$\text{Enc}(\text{pk}_A \| \text{pk}_B , x)$

$\text{Synchronize}(\text{sk}_A , \text{pk}_B , \boxed{x}) \rightarrow$ $\text{Enc}(\text{pk}_A \| \text{pk}_B , \text{sk}_A \cdot x)$ $\leftarrow \text{Synchronize}(\text{sk}_B , \text{pk}_A , \boxed{x})$

$\text{Enc}(\text{pk}_A \| \text{pk}_B , \text{sk}_B \cdot x)$

$\text{Enc}(\text{pk}_A \| \text{pk}_B , x) = \text{Enc}(\text{pk}_A , r) , \text{Enc}(\text{pk}_B , x - r)$

Multiplication

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x)$

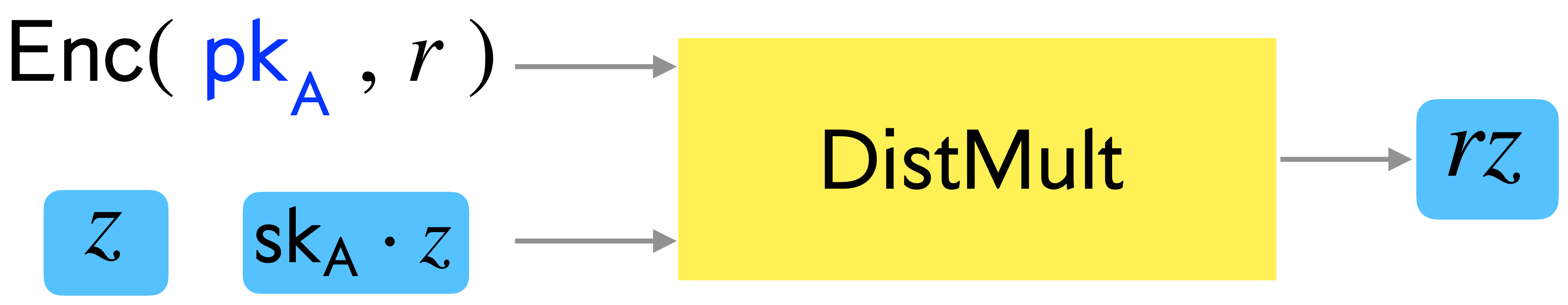
$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x)$

$\mathsf{Synchronize}(\mathsf{sk}_A, \mathsf{pk}_B, \boxed{x}) \rightarrow$ $\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_A \cdot x)$ $\leftarrow \mathsf{Synchronize}(\mathsf{sk}_B, \mathsf{pk}_A, \boxed{x})$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_B \cdot x)$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x) = \mathsf{Enc}(\mathsf{pk}_A, r), \mathsf{Enc}(\mathsf{pk}_B, x - r)$

Multiplication



$\mathsf{Enc}(\mathsf{pk}_A, r)$

z

$\mathsf{sk}_A \cdot z$

DistMult

rz

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathrm{pk}_A, \mathrm{sk}_A) \leftarrow \mathrm{KeyGen}$

$\boxed{x} = \mathrm{Enc}(\mathrm{pk}_A, x)$

$(\mathrm{pk}_B, \mathrm{sk}_B) \leftarrow \mathrm{KeyGen}$

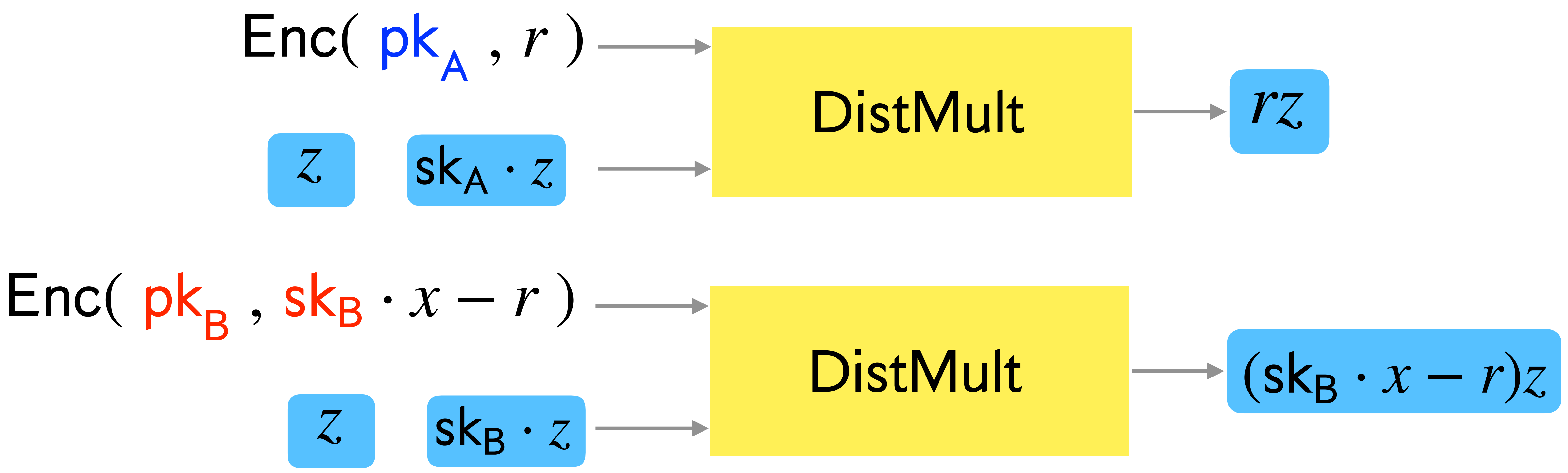
$$\mathrm{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, x)$$

$$\mathrm{Synchronize}(\mathrm{sk}_A, \mathrm{pk}_B, \boxed{x}) \rightarrow \quad \mathrm{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, \mathrm{sk}_A \cdot x) \quad \leftarrow \mathrm{Synchronize}(\mathrm{sk}_B, \mathrm{pk}_A, \boxed{x})$$

$$\mathrm{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, \mathrm{sk}_B \cdot x)$$

$$\mathrm{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, x) = \mathrm{Enc}(\mathrm{pk}_A, r), \mathrm{Enc}(\mathrm{pk}_B, x - r)$$

Multiplication



$\mathrm{Enc}(\mathrm{pk}_A, r)$
z
$\mathrm{sk}_A \cdot z$
DistMult
rz
$\mathrm{Enc}(\mathrm{pk}_B, \mathrm{sk}_B \cdot x - r)$
z
$\mathrm{sk}_B \cdot z$
DistMult
$(\mathrm{sk}_B \cdot x - r)z$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow \mathsf{KeyGen}$

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow \mathsf{KeyGen}$

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x)$

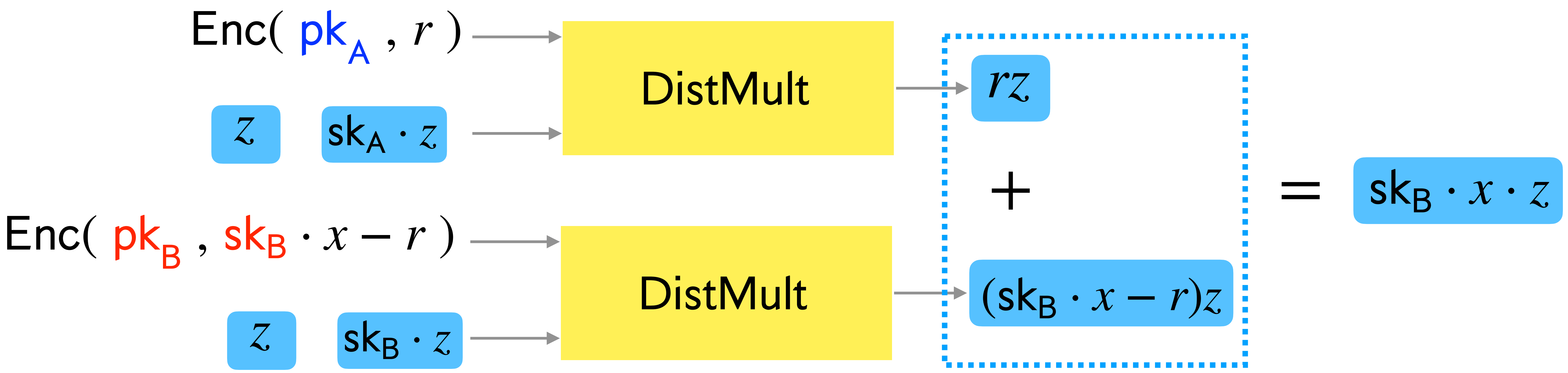
$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x)$

$\mathsf{Synchronize}(\mathsf{sk}_A, \mathsf{pk}_B, \boxed{x}) \rightarrow$ $\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_A \cdot x)$ $\leftarrow \mathsf{Synchronize}(\mathsf{sk}_B, \mathsf{pk}_A, \boxed{x})$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_B \cdot x)$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x) = \mathsf{Enc}(\mathsf{pk}_A, r), \mathsf{Enc}(\mathsf{pk}_B, x - r)$

Multiplication



$\mathsf{Enc}(\mathsf{pk}_A, r)$

z $\quad$ $\mathsf{sk}_A \cdot z$

$\mathsf{Enc}(\mathsf{pk}_B, \mathsf{sk}_B \cdot x - r)$

z $\quad$ $\mathsf{sk}_B \cdot z$

DistMult

DistMult

rz

$+$

$(\mathsf{sk}_B \cdot x - r)z$

$= \mathsf{sk}_B \cdot x \cdot z$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathsf{pk}_A, \mathsf{sk}_A) \leftarrow$ KeyGen

$(\mathsf{pk}_B, \mathsf{sk}_B) \leftarrow$ KeyGen

$\boxed{x} = \mathsf{Enc}(\mathsf{pk}_A, x)$

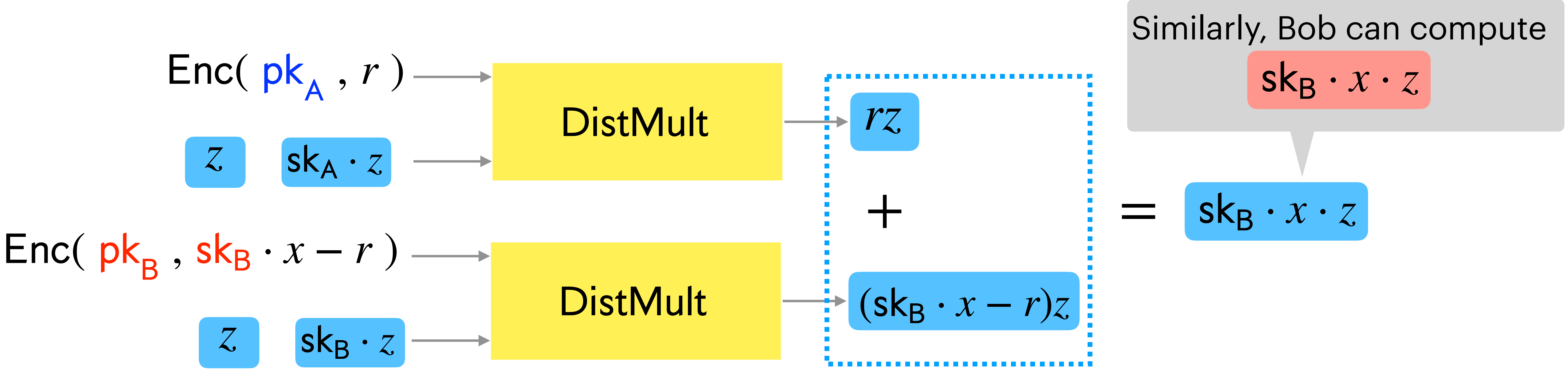
$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x)$

Synchronize$(\mathsf{sk}_A, \mathsf{pk}_B, \boxed{x}) \rightarrow$ $\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_A \cdot x)$ $\leftarrow$ Synchronize$(\mathsf{sk}_B, \mathsf{pk}_A, \boxed{x})$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, \mathsf{sk}_B \cdot x)$

$\mathsf{Enc}(\mathsf{pk}_A \| \mathsf{pk}_B, x) = \mathsf{Enc}(\mathsf{pk}_A, r), \mathsf{Enc}(\mathsf{pk}_B, x - r)$

Multiplication



$\mathsf{Enc}(\mathsf{pk}_A, r)$

z $\quad \mathsf{sk}_A \cdot z$

DistMult

$\mathsf{Enc}(\mathsf{pk}_B, \mathsf{sk}_B \cdot x - r)$

z $\quad \mathsf{sk}_B \cdot z$

DistMult

rz

$+$

$(\mathsf{sk}_B \cdot x - r)z$

$= \mathsf{sk}_B \cdot x \cdot z$

Similarly, Bob can compute $\mathsf{sk}_B \cdot x \cdot z$

Constructing Multi-Key HSS: Synchronizable Encryption Scheme

$(\mathrm{pk}_A, \mathrm{sk}_A) \leftarrow \text{KeyGen}$

$\boxed{x} = \text{Enc}(\mathrm{pk}_A, x)$

$(\mathrm{pk}_B, \mathrm{sk}_B) \leftarrow \text{KeyGen}$

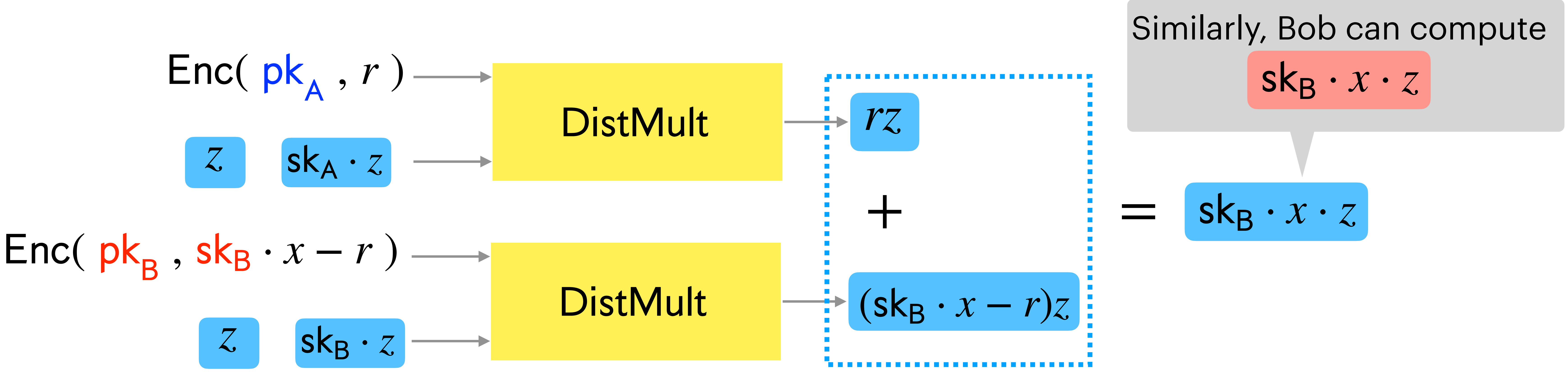
$$\text{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, x)$$

$$\text{Synchronize}(\mathrm{sk}_A, \mathrm{pk}_B, \boxed{x}) \rightarrow \text{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, \mathrm{sk}_A \cdot x) \leftarrow \text{Synchronize}(\mathrm{sk}_B, \mathrm{pk}_A, \boxed{x})$$

$$\text{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, \mathrm{sk}_B \cdot x)$$

$$\text{Enc}(\mathrm{pk}_A \| \mathrm{pk}_B, x) = \text{Enc}(\mathrm{pk}_A, r), \text{Enc}(\mathrm{pk}_B, x - r)$$

Multiplication



$\text{Enc}(\mathrm{pk}_A, r)$
z
$\mathrm{sk}_A \cdot z$
DistMult
$\text{Enc}(\mathrm{pk}_B, \mathrm{sk}_B \cdot x - r)$
z
$\mathrm{sk}_B \cdot z$
DistMult
rz
$+$
$(\mathrm{sk}_B \cdot x - r)z$
$=$
$\mathrm{sk}_B \cdot x \cdot z$

Similarly, Bob can compute $\mathrm{sk}_B \cdot x \cdot z$

Evaluation invariant recovered

Thank You

eprint.iacr.org/2025/094