

Enhanced Trapdoor Hashing from DDH and DCR

EUROCRYPT 2025



Geoffroy Couteau

CNRS, IRIF

Université Paris Cité

Aditya Hegde

JHU



Sihang Pu

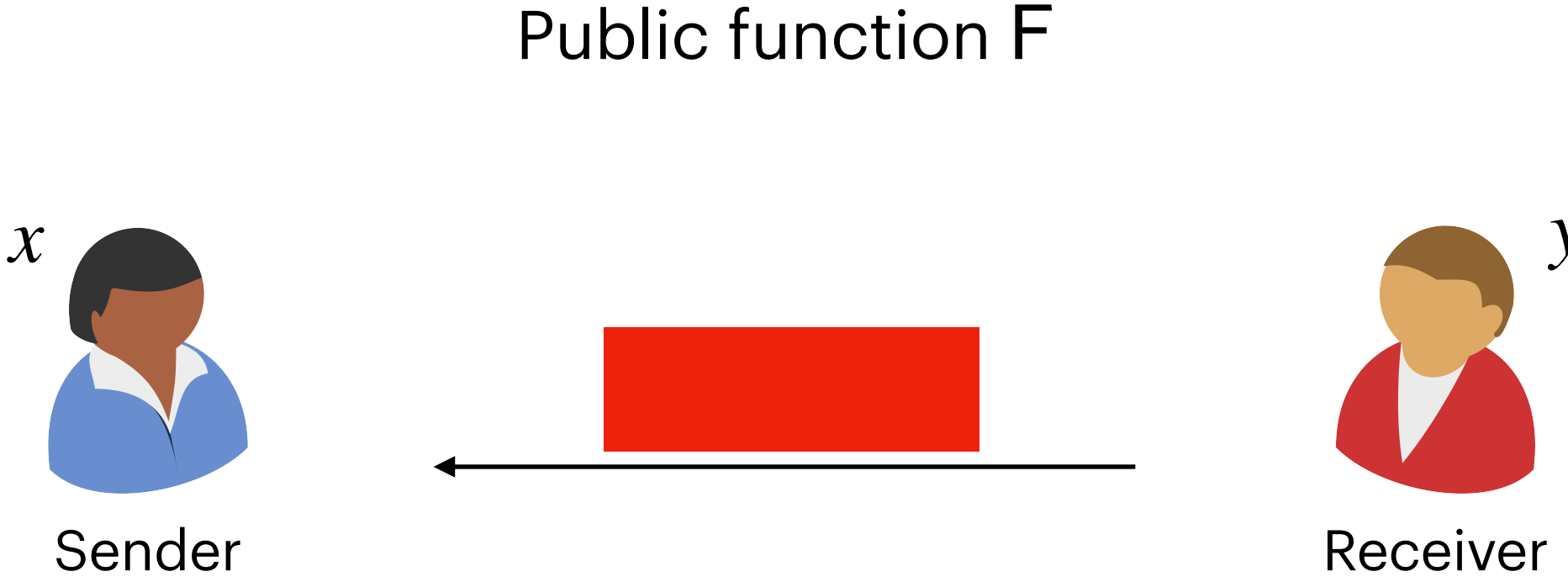
CNRS, IRIF

Université Paris Cité

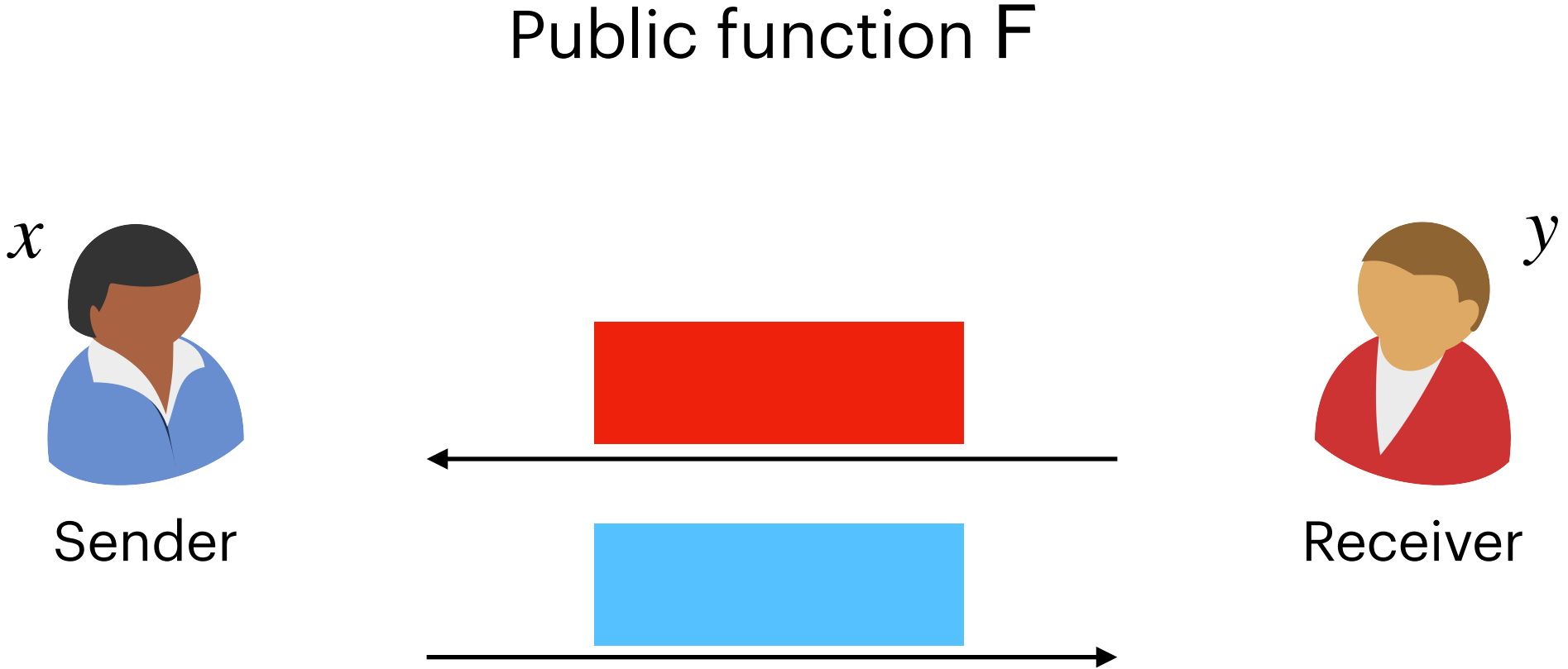
Setting: Two-Round Sender-Receiver Computation



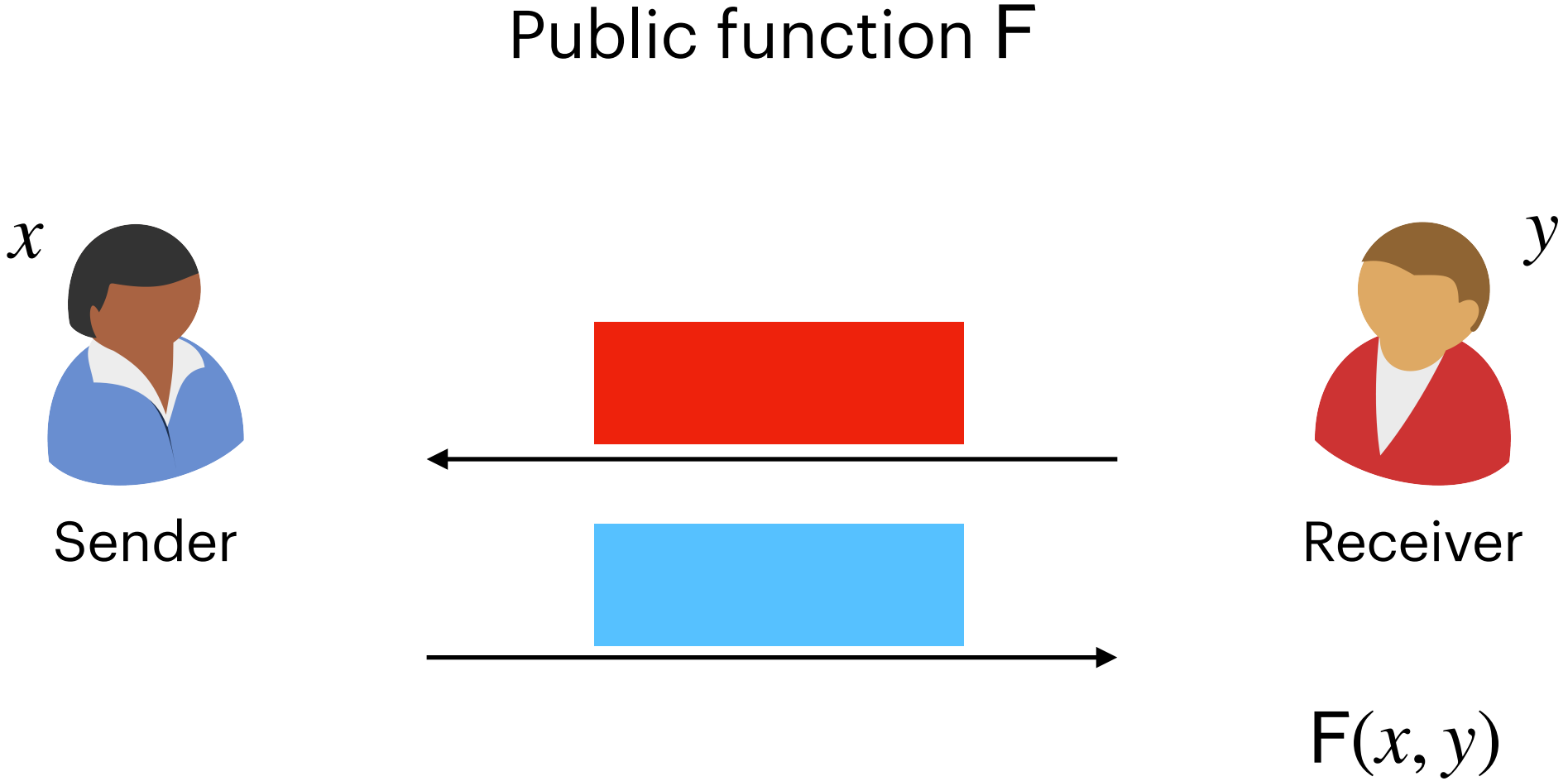
Setting: Two-Round Sender-Receiver Computation



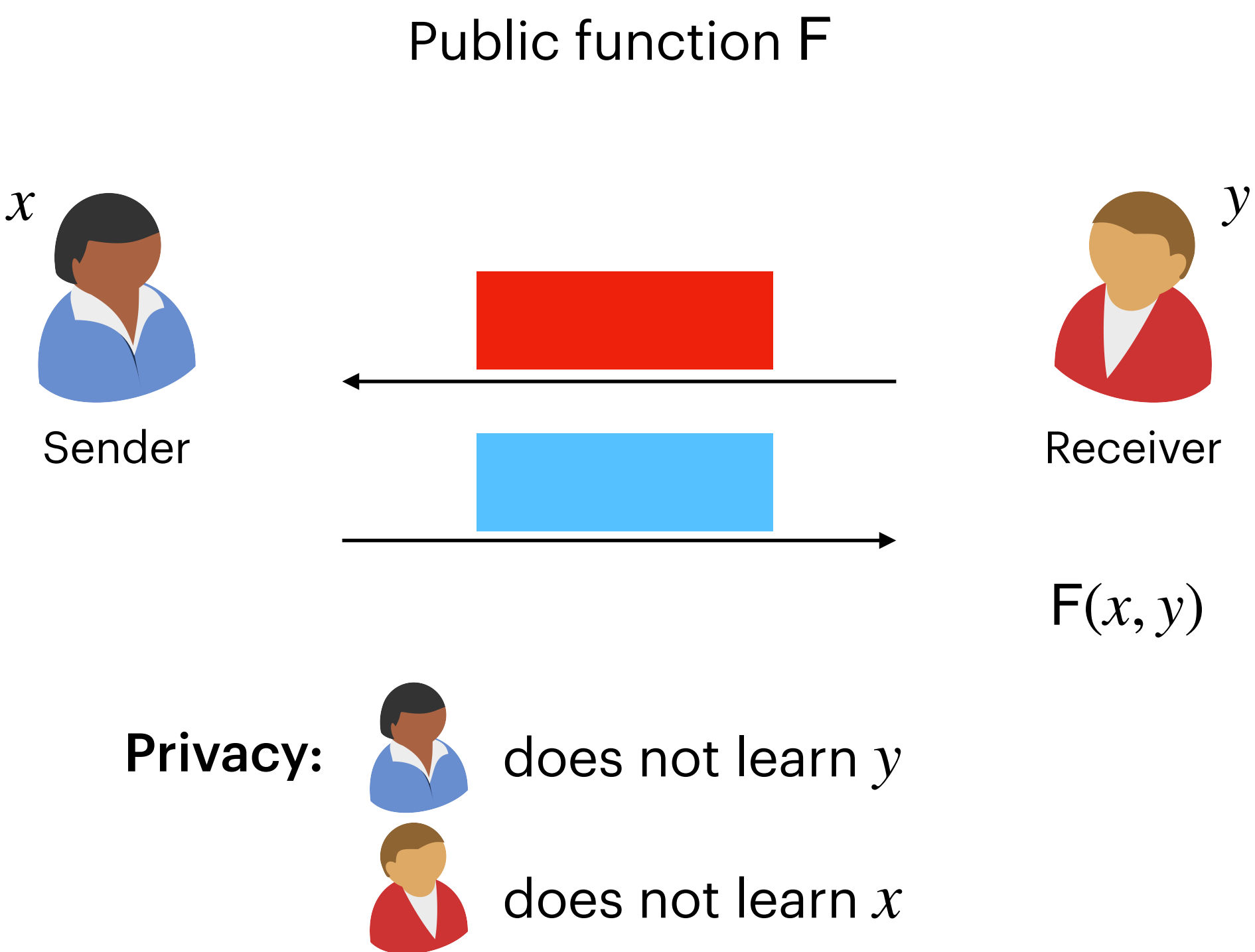
Setting: Two-Round Sender-Receiver Computation



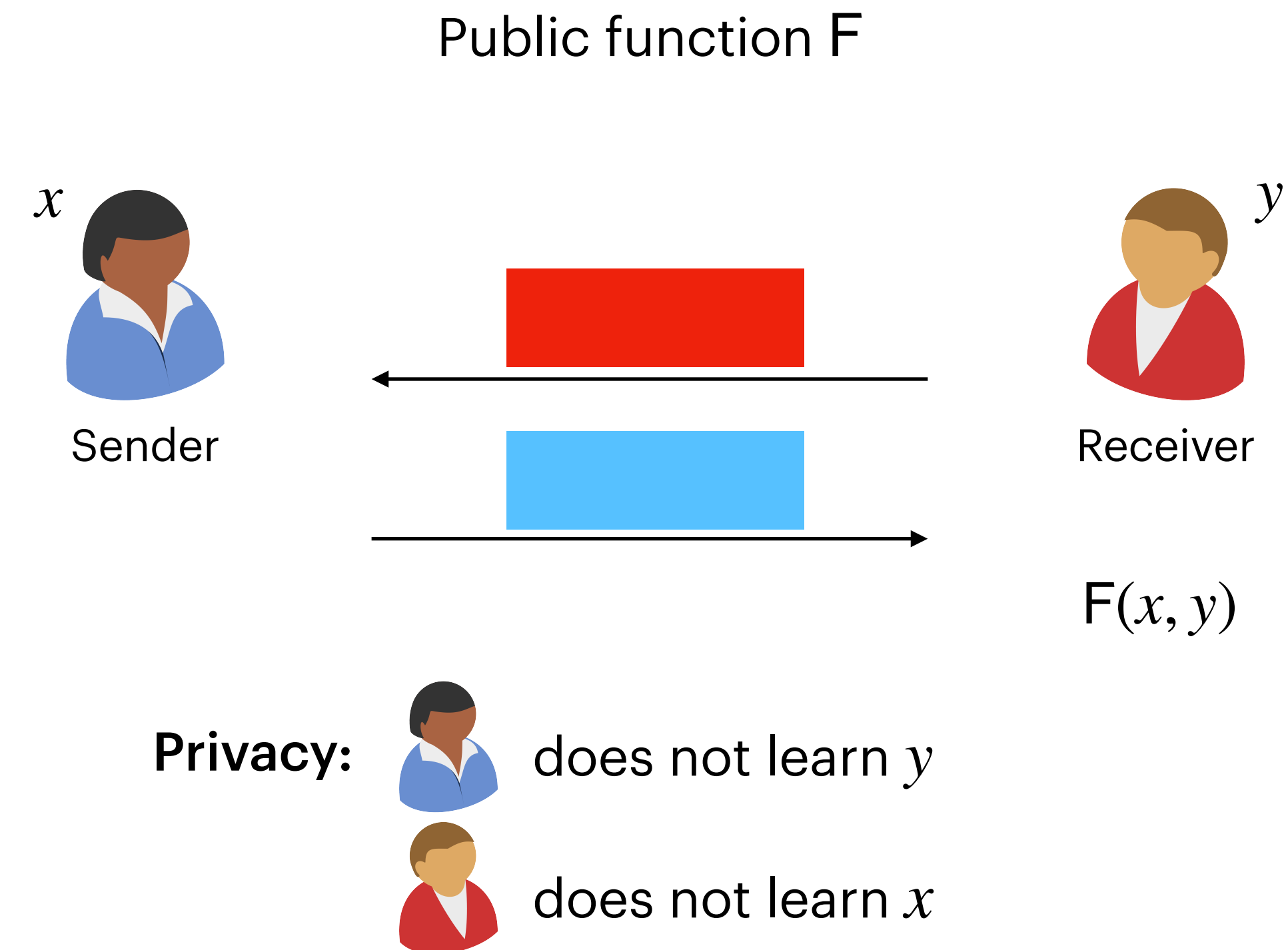
Setting: Two-Round Sender-Receiver Computation



Setting: Two-Round Sender-Receiver Computation

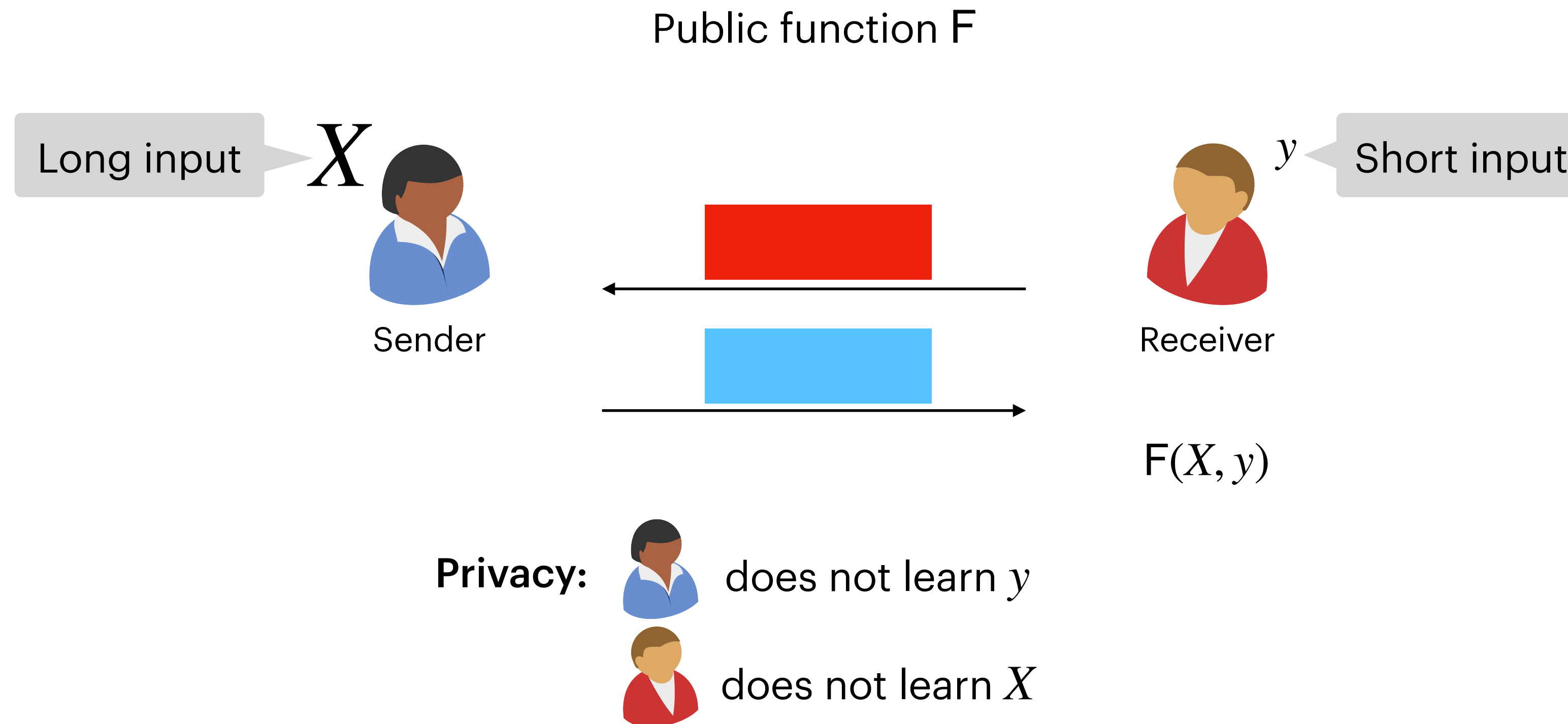


Setting: Two-Round Sender-Receiver Computation



What is the minimum communication cost of semi-honest secure protocols?

Setting: Two-Round Sender-Receiver Computation



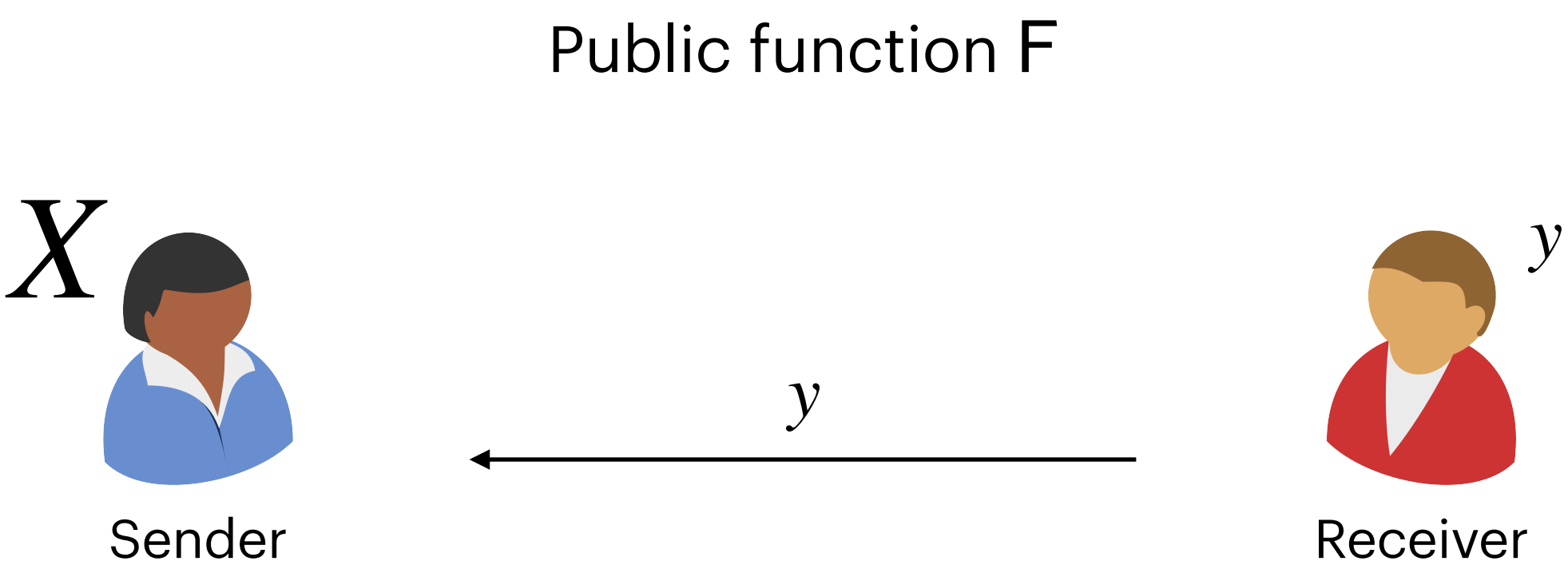
What is the minimum communication cost of semi-honest secure protocols?

Ideal World Two-Round Sender-Receiver Computation

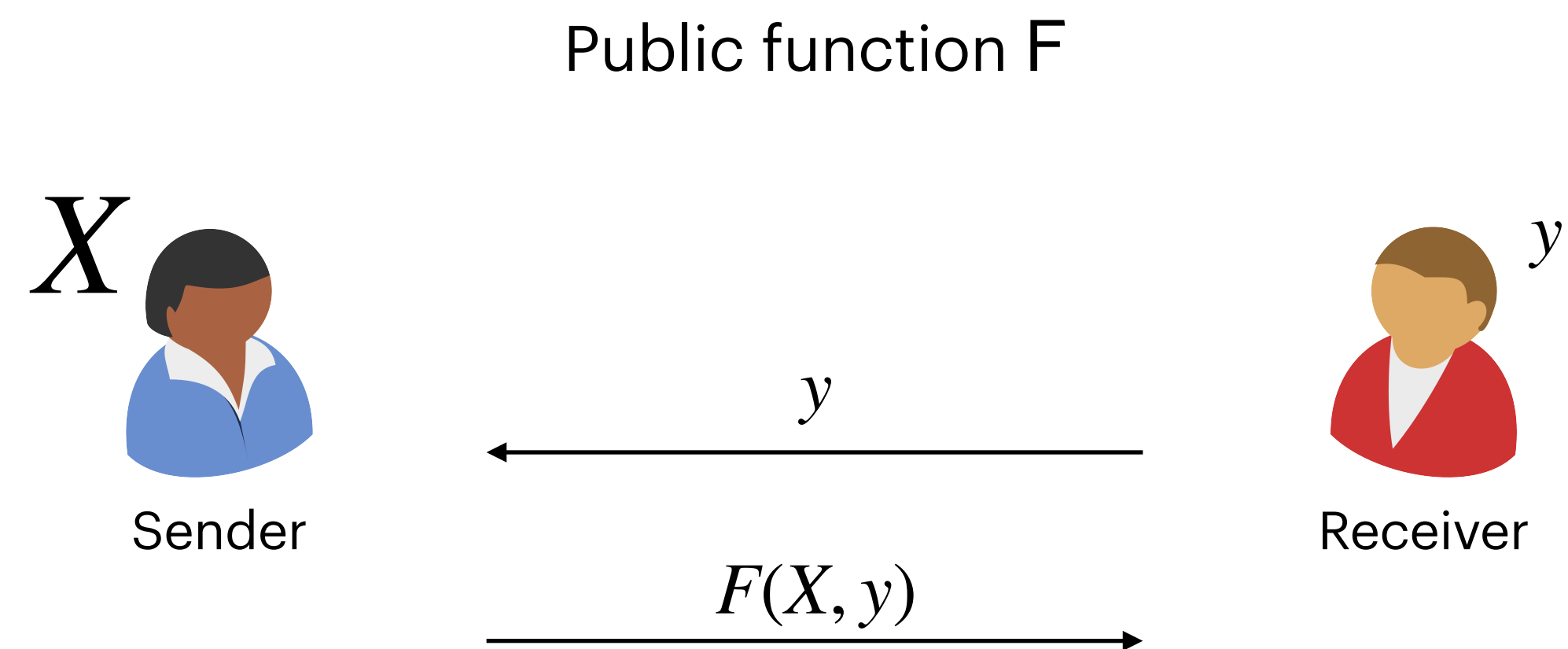
Public function F



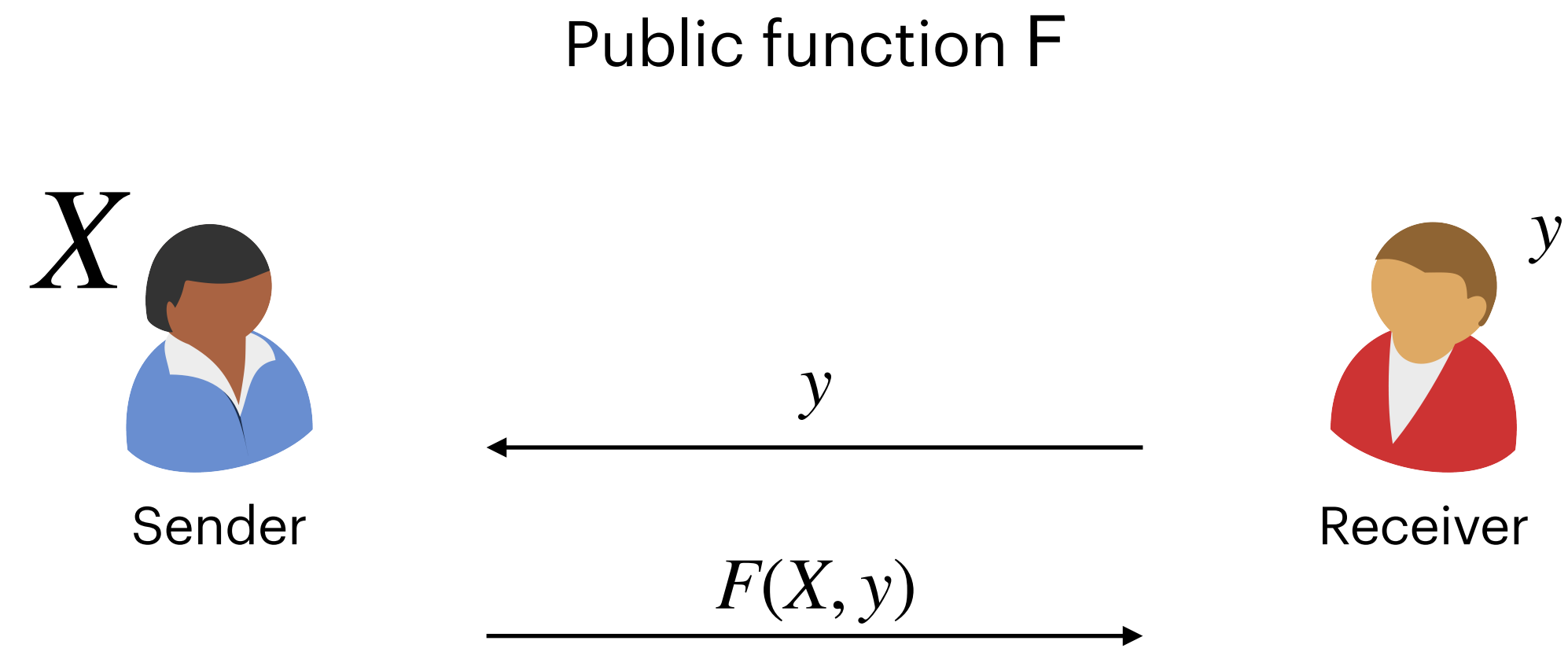
Ideal World Two-Round Sender-Receiver Computation



Ideal World Two-Round Sender-Receiver Computation

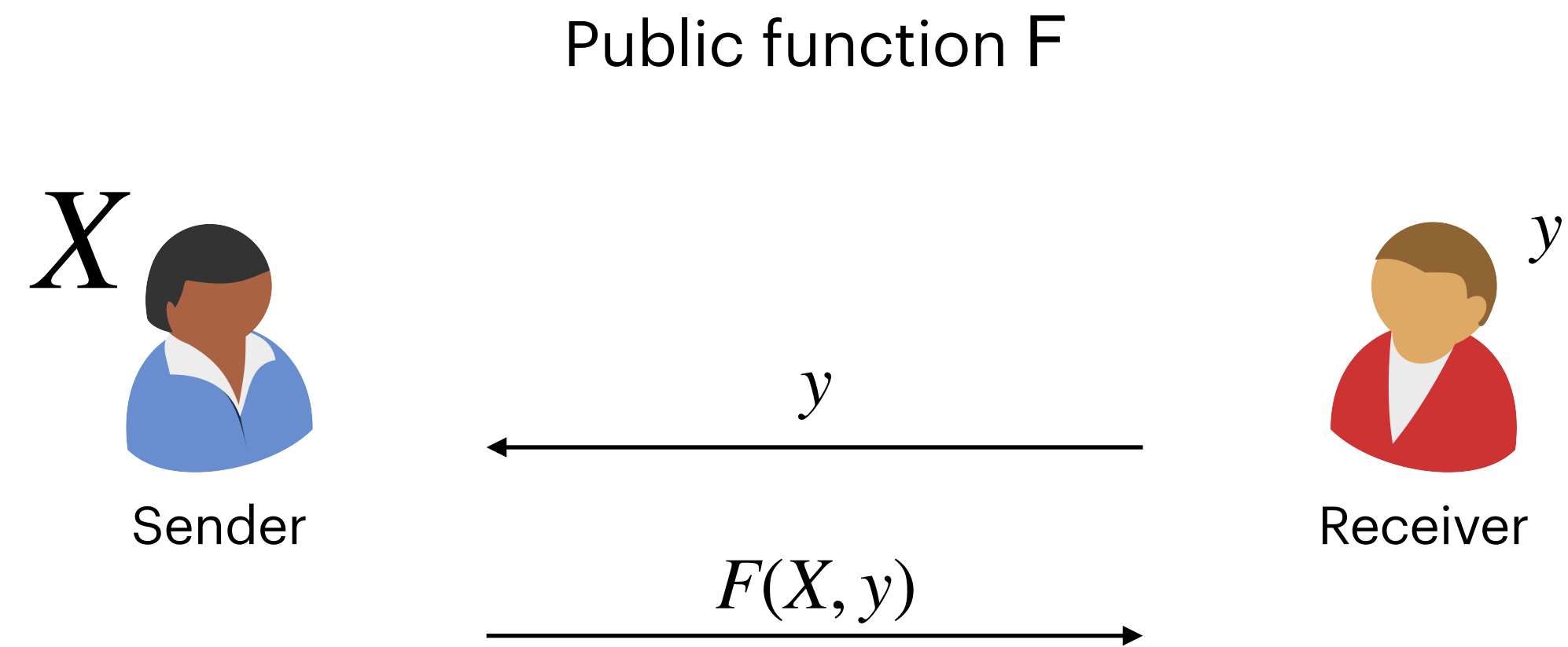


Ideal World Two-Round Sender-Receiver Computation



Total communication: $|y| + |F(X, y)|$

Ideal World Two-Round Sender-Receiver Computation



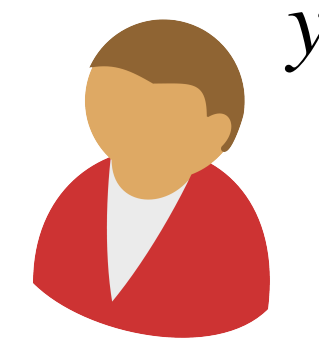
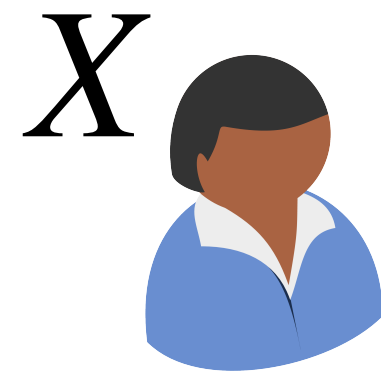
Total communication: $|y| + |F(X, y)|$

Can secure protocols achieve similar efficiency?

Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

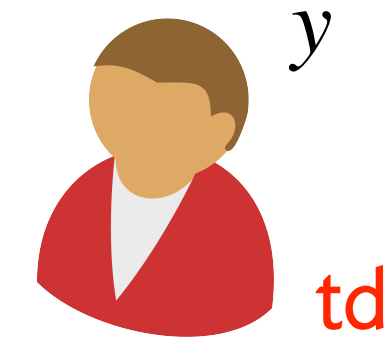
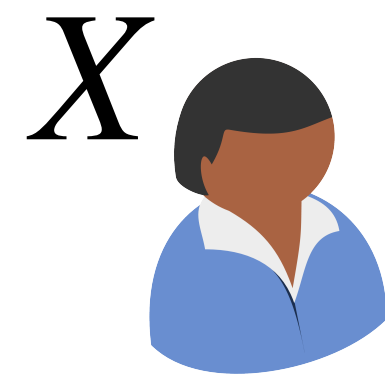
Public function F



Trapdoor Hash Functions (TDH)

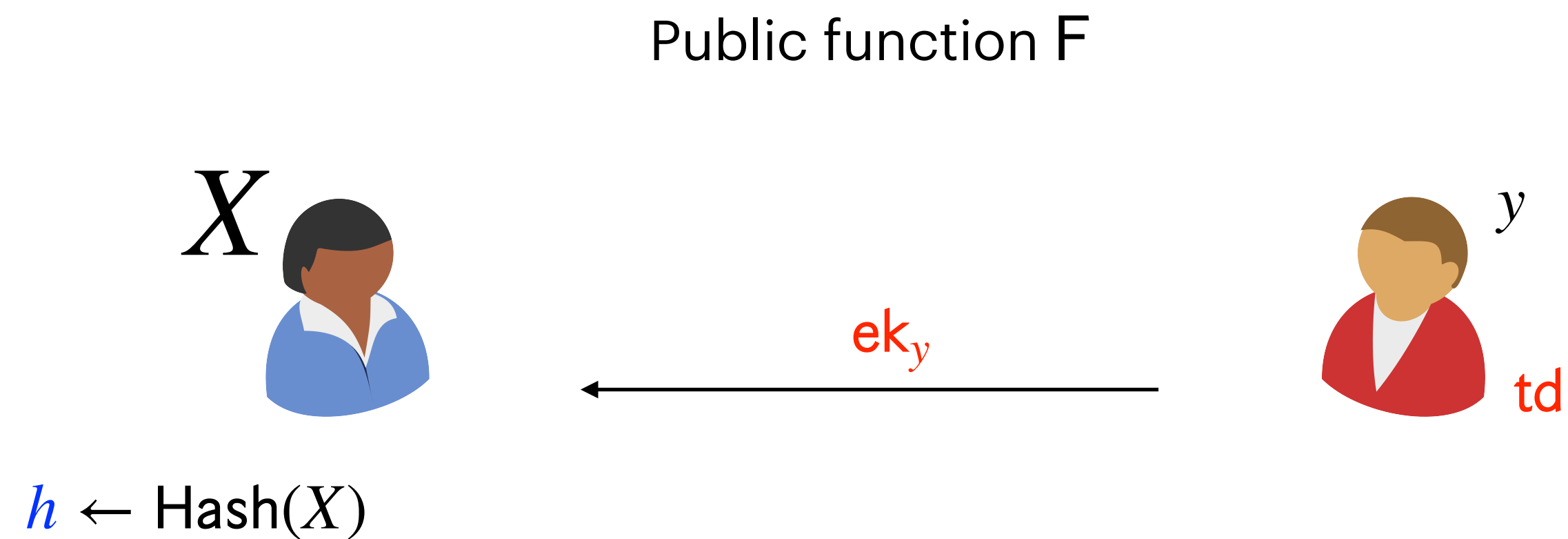
[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

Public function F



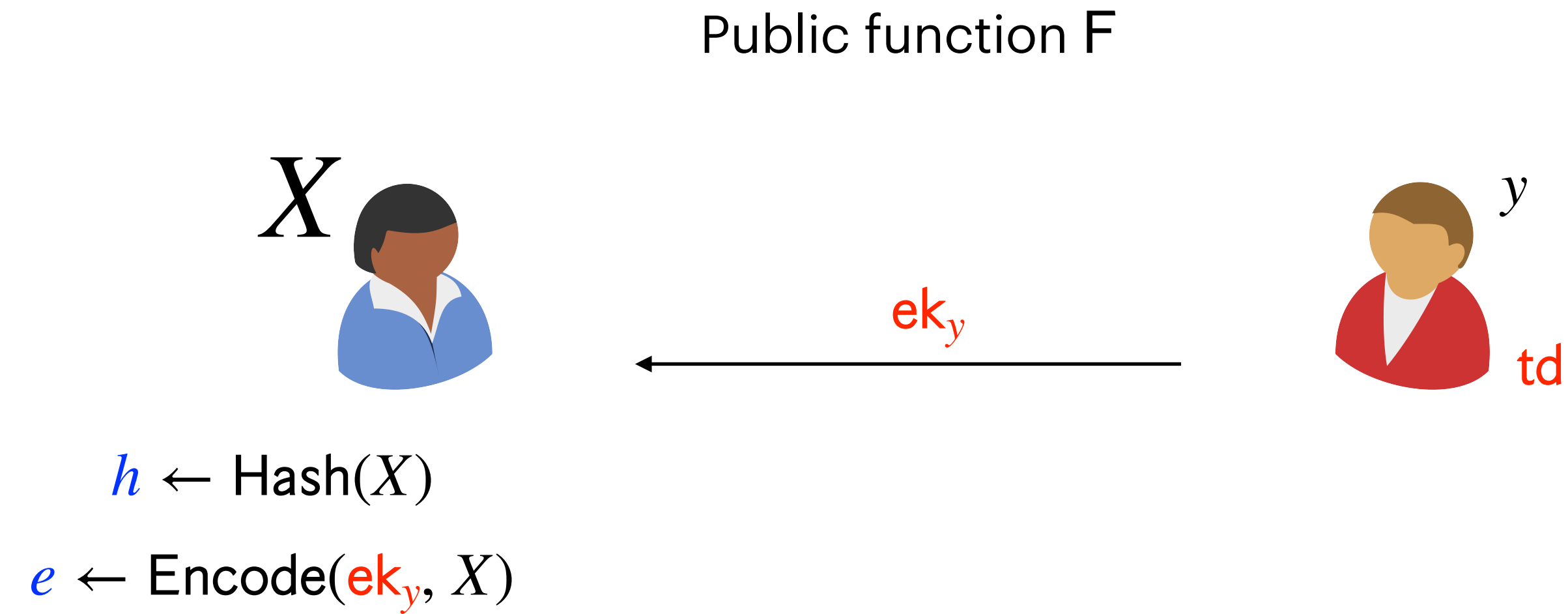
Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



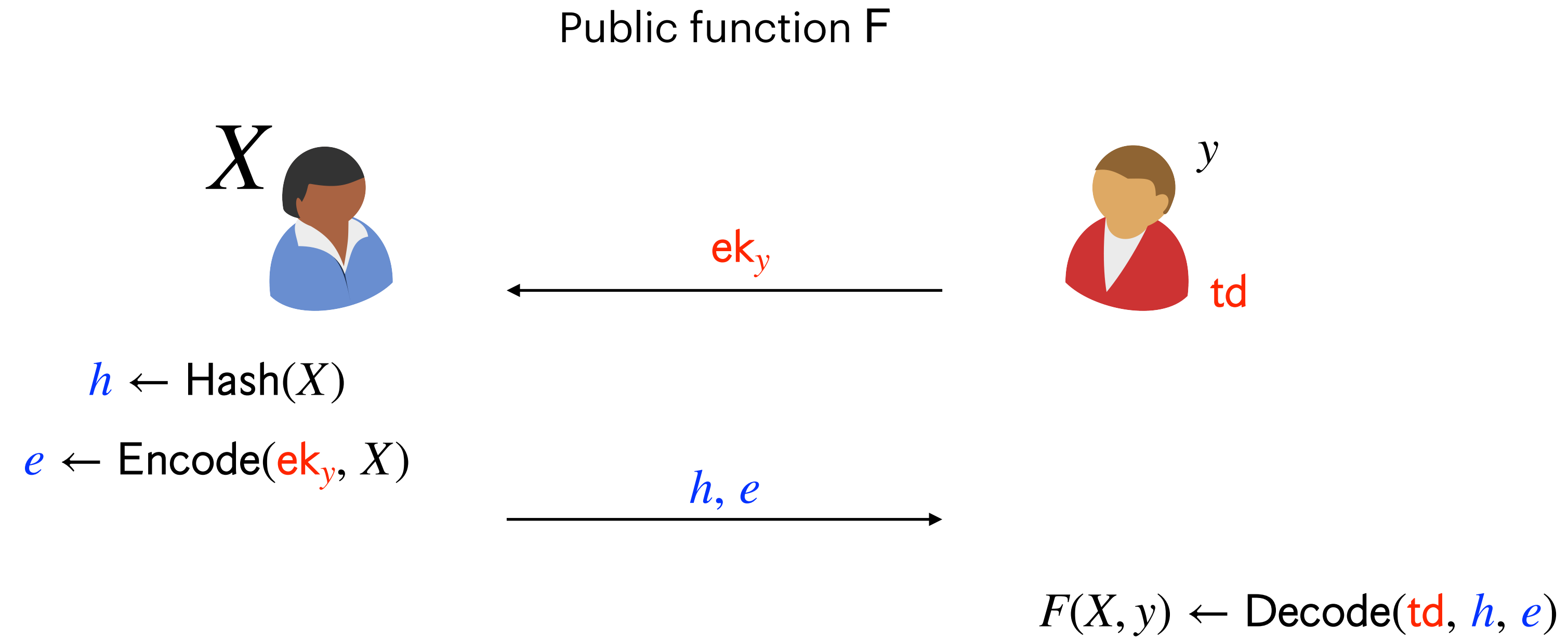
Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



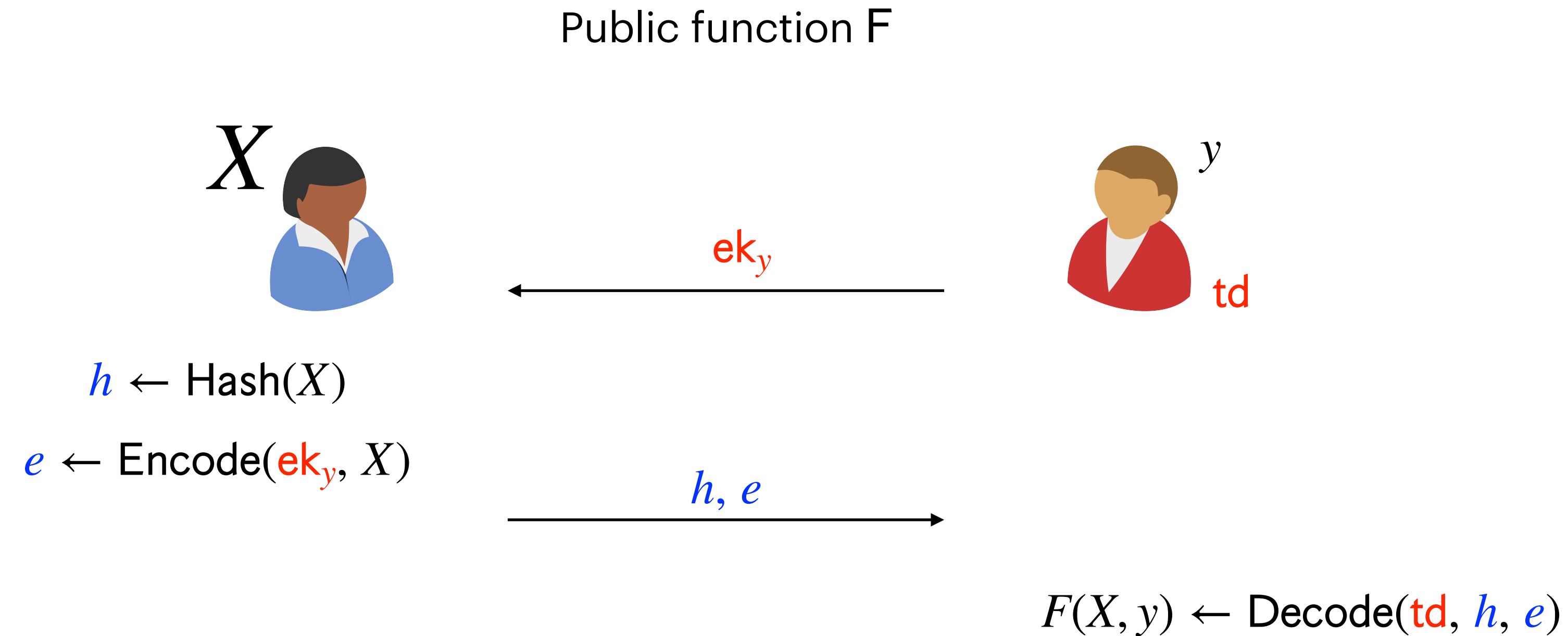
Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

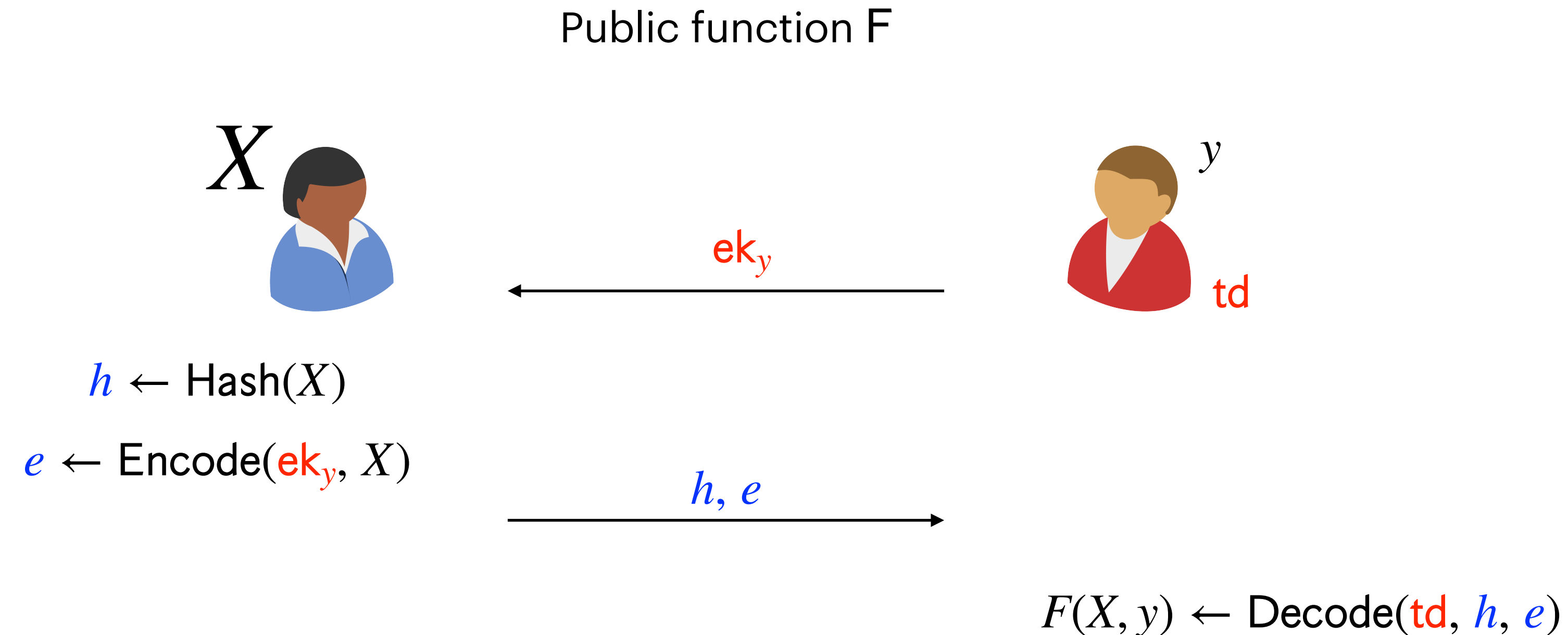


Privacy: ek_y hides y

(h, e) hides X

Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



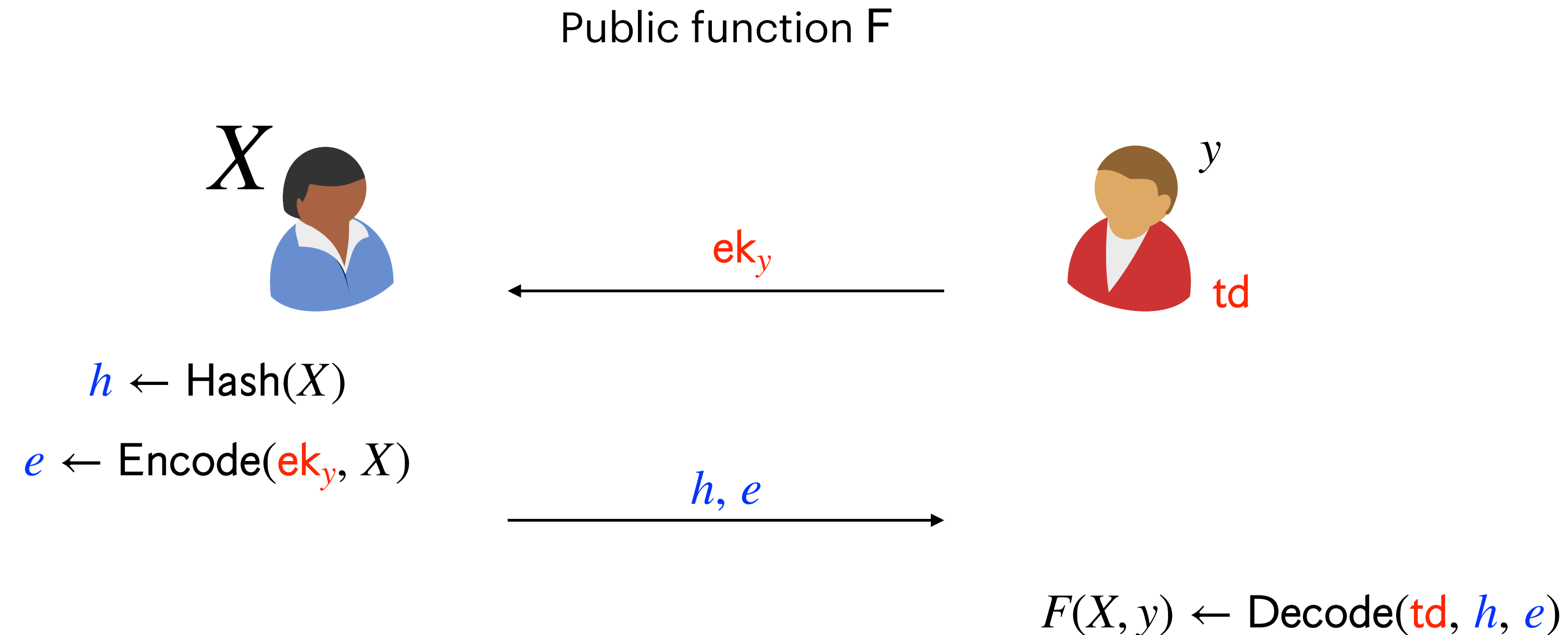
Privacy: ek_y hides y

(h, e) hides X

Efficiency: h is small i.e., $|h| = o(|X|) \cdot \text{poly}(\lambda)$

Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



Privacy: ek_y hides y

(h, e) hides X

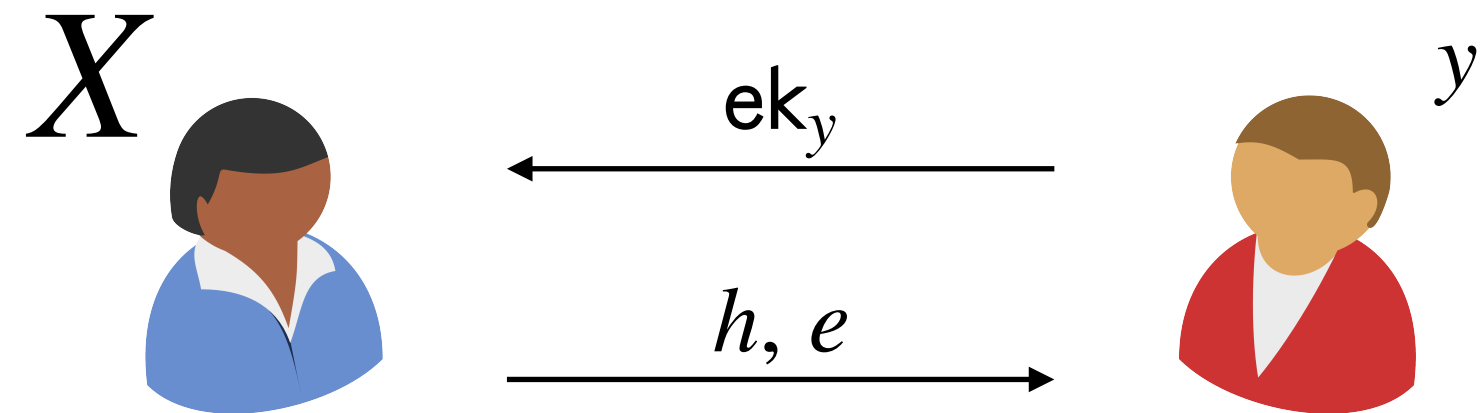
Efficiency: h is small i.e., $|h| = o(|X|) \cdot \text{poly}(\lambda)$

e has high rate i.e., $|e| \approx |F(X, y)|$

Rate: $\frac{|F(X, y)|}{|e|}$

Trapdoor Hash Functions (TDH)

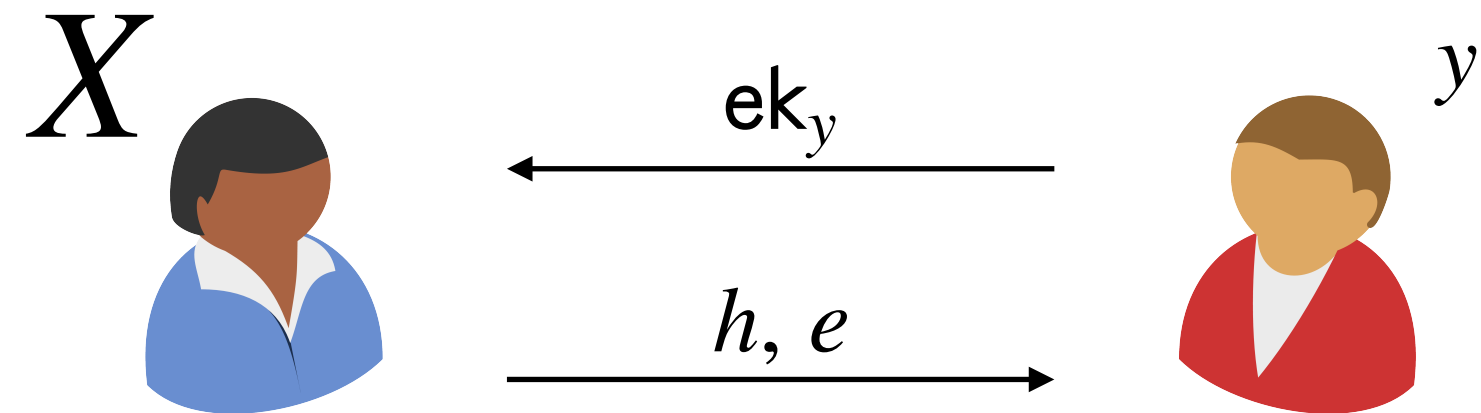
[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

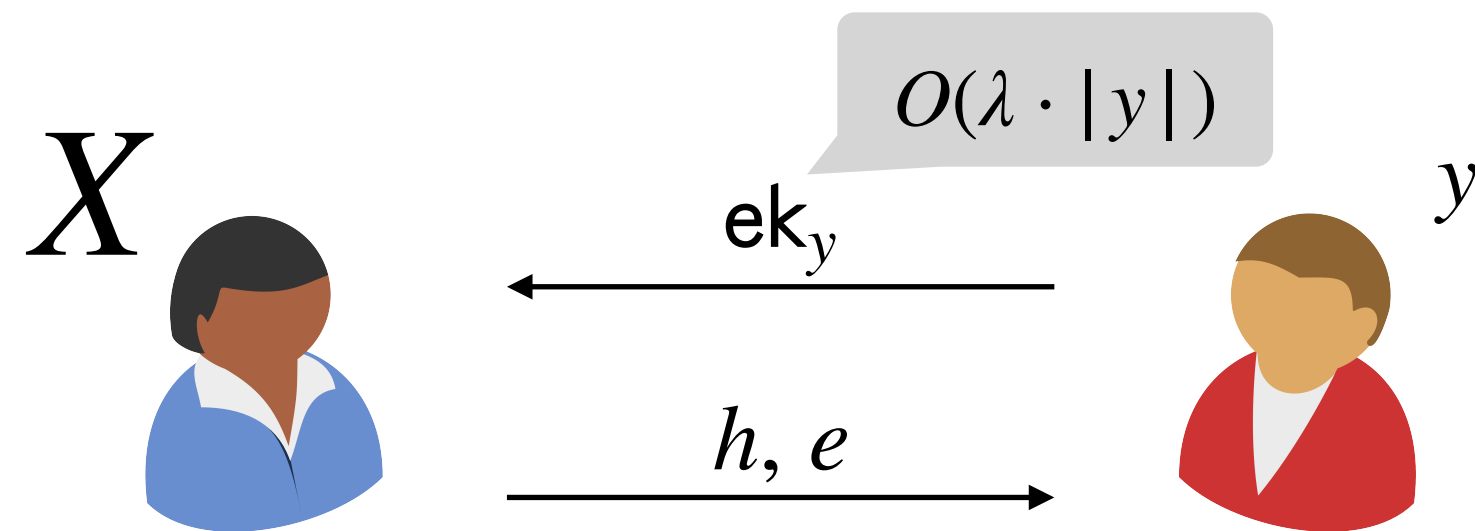
$$F(X, y) = \sum_i x_i \cdot y_i$$



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

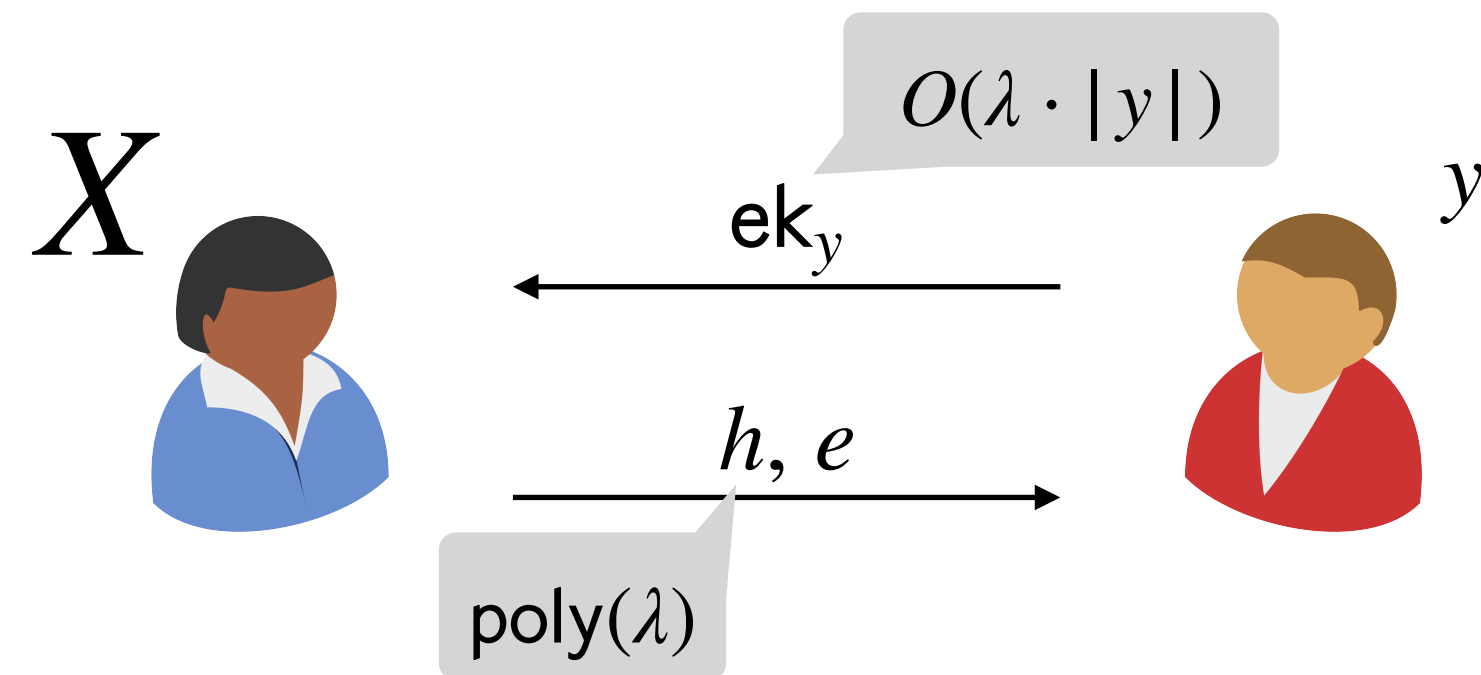
$$F(X, y) = \sum_i x_i \cdot y_i$$



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

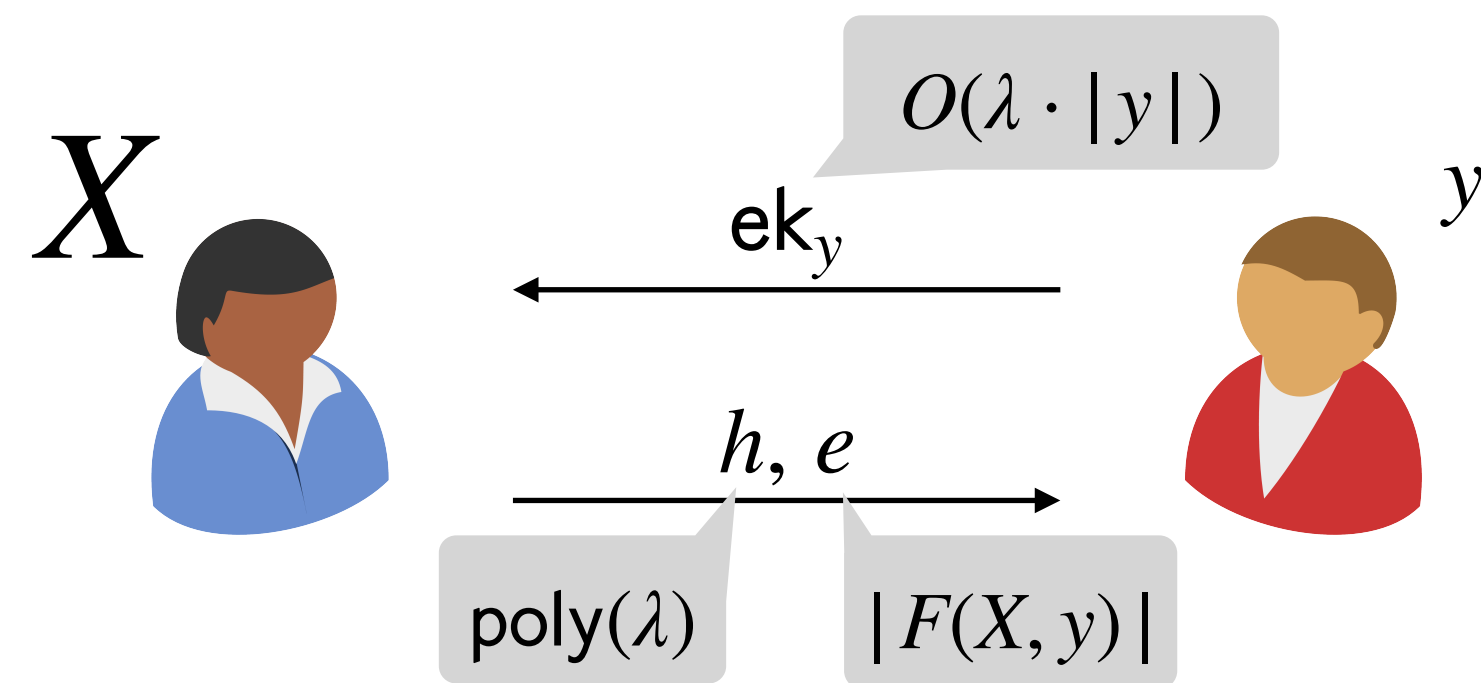
$$F(X, y) = \sum_i x_i \cdot y_i$$



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

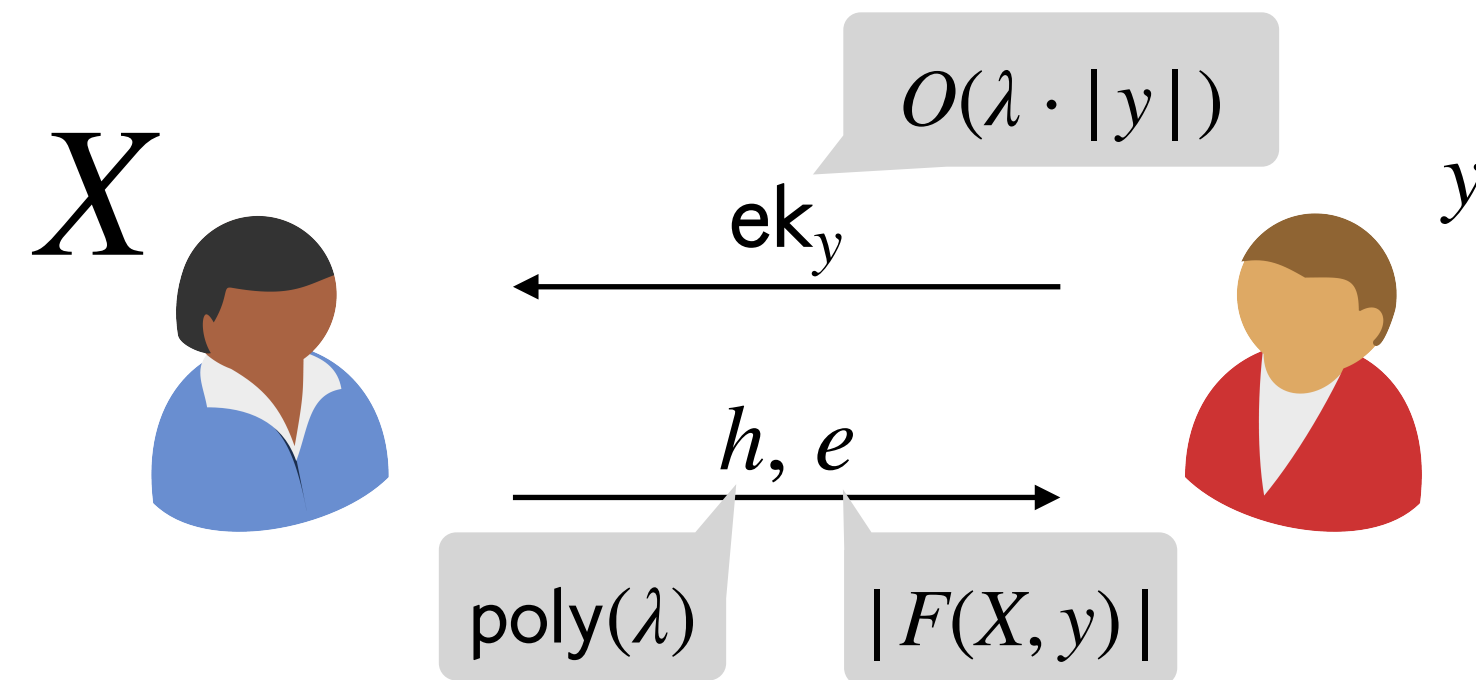
$$F(X, y) = \sum_i x_i \cdot y_i$$



Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

$$F(X, y) = \sum_i x_i \cdot y_i$$

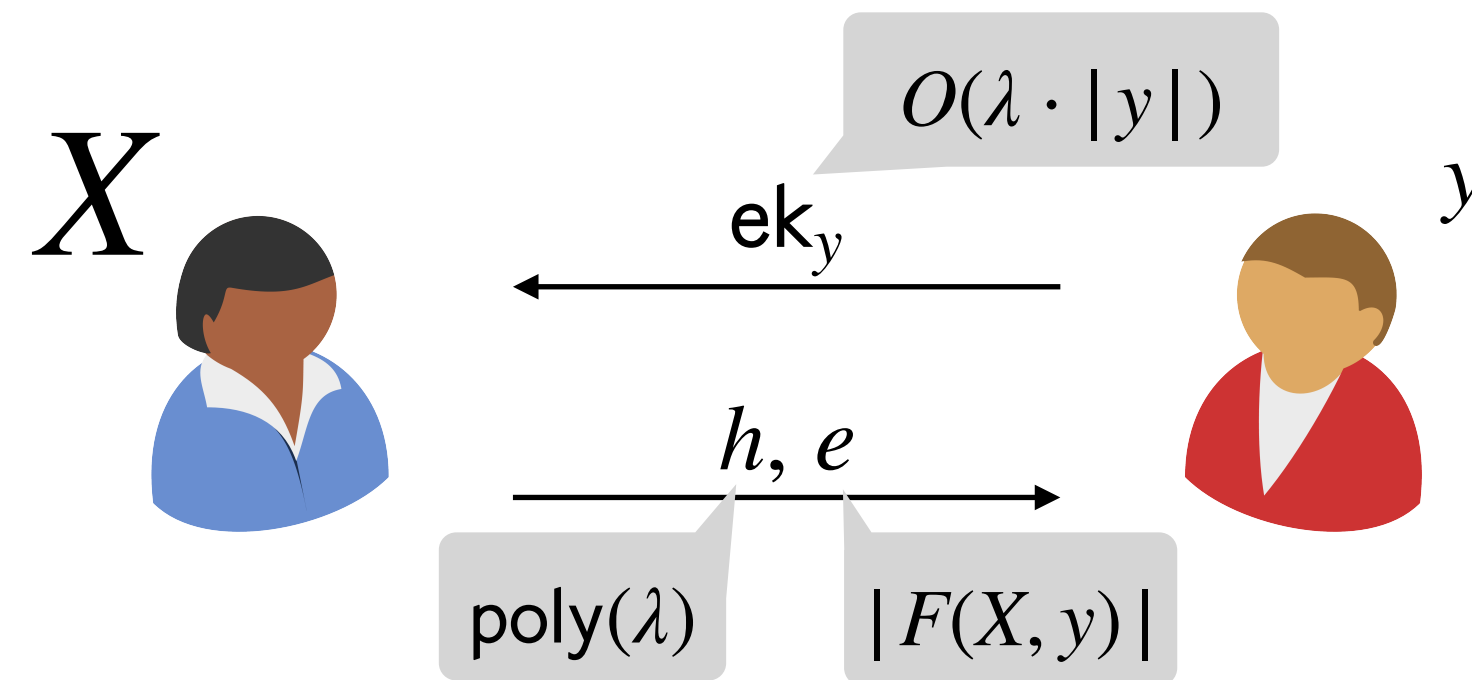


Assumptions: DCR, DDH, QR, LWE

Trapdoor Hash Functions (TDH)

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

$$F(X, y) = \sum_i x_i \cdot y_i$$



Assumptions: DCR, DDH, QR, LWE

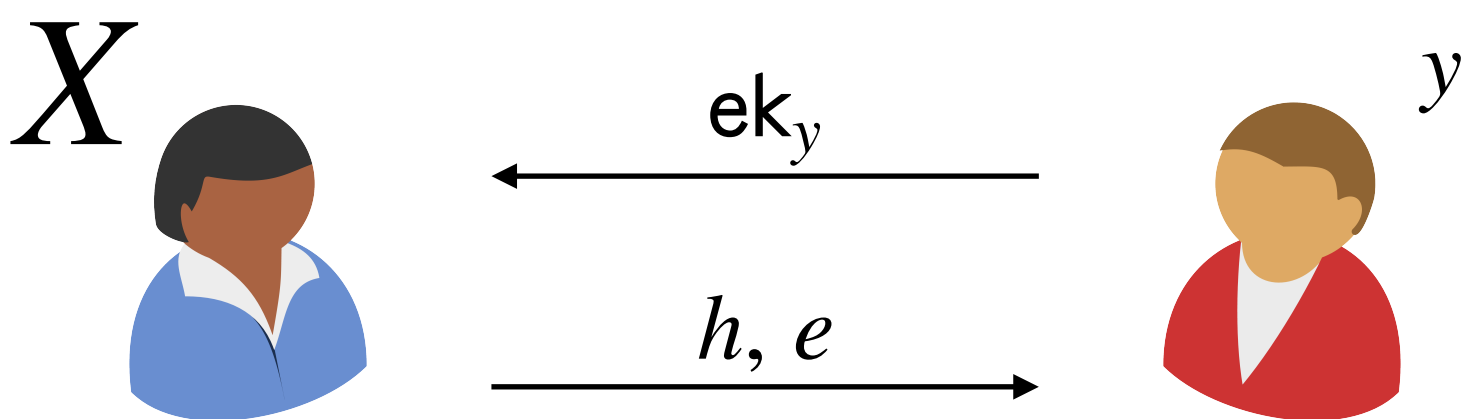
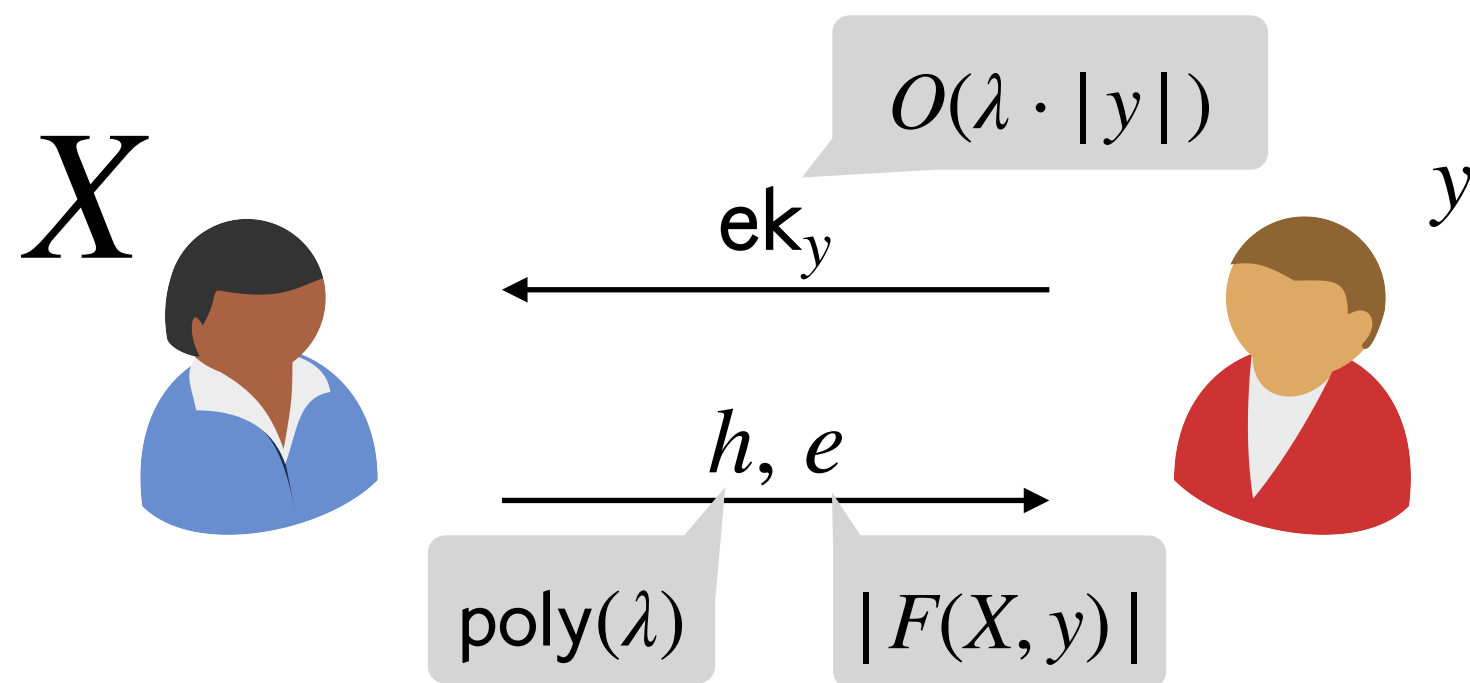
Can we improve the functionality of TDH from **group-based** assumptions?

Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$



Enhanced Trapdoor Hash Functions from DDH and DCR

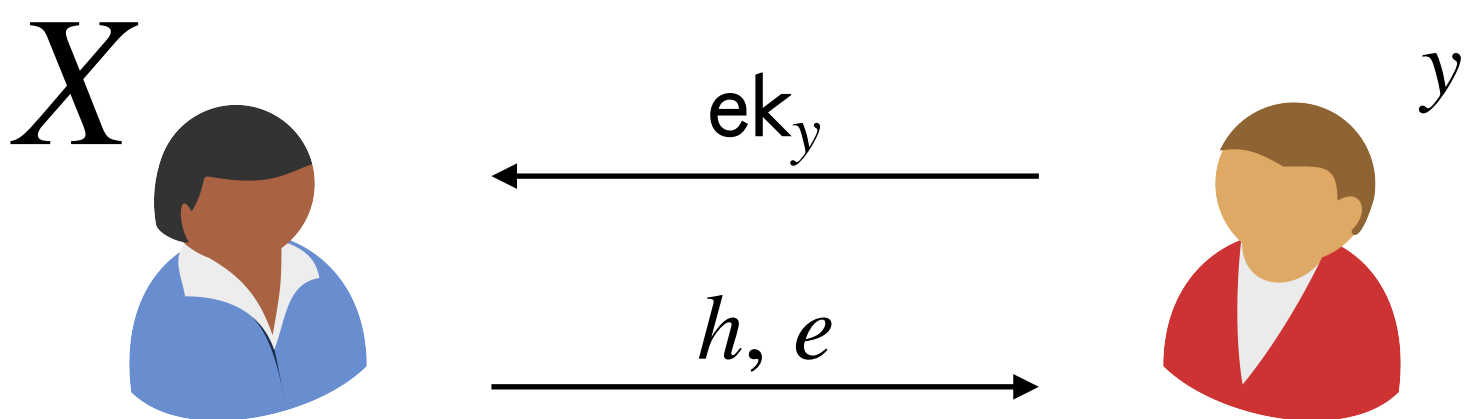
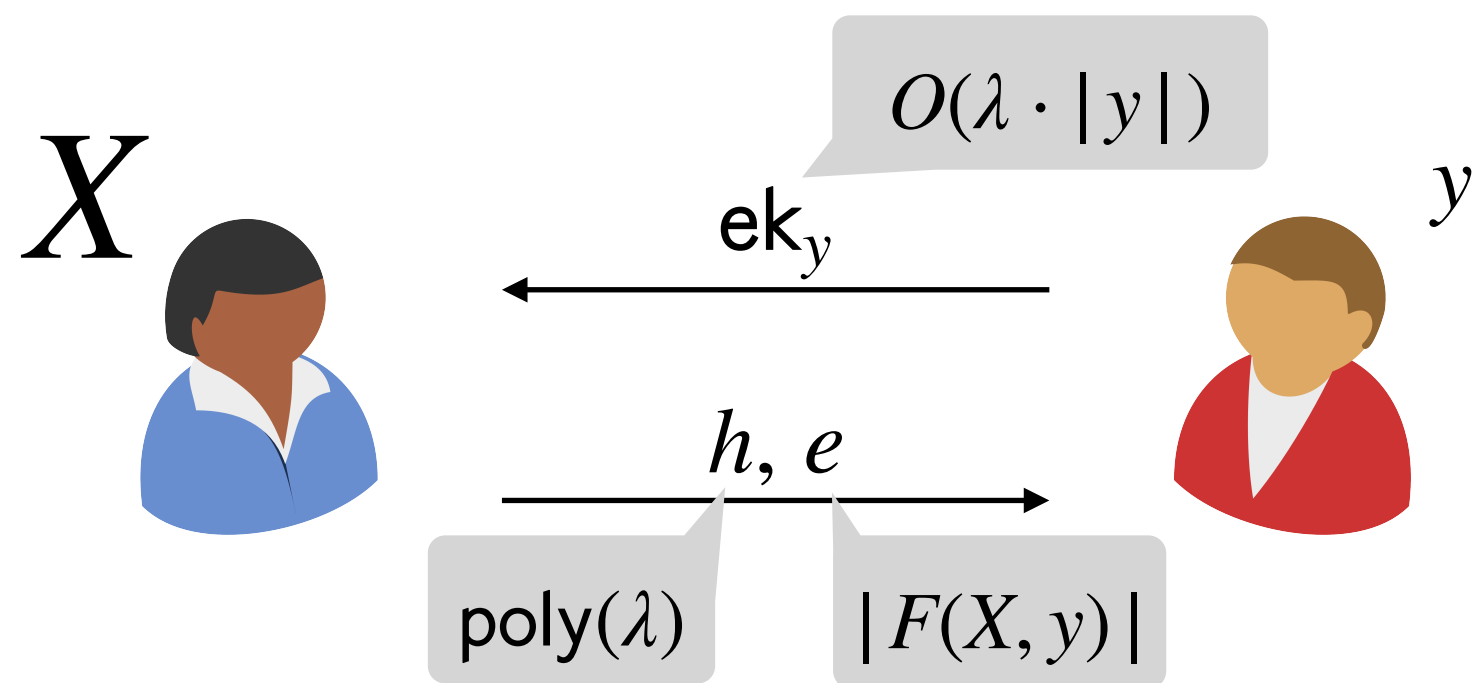
[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F(X, y) = \sum_{i=1}^n f_i(X) \cdot g_i(y)$$

NC¹



Expressivity

Supports computing
Bilinear-NC¹ programs

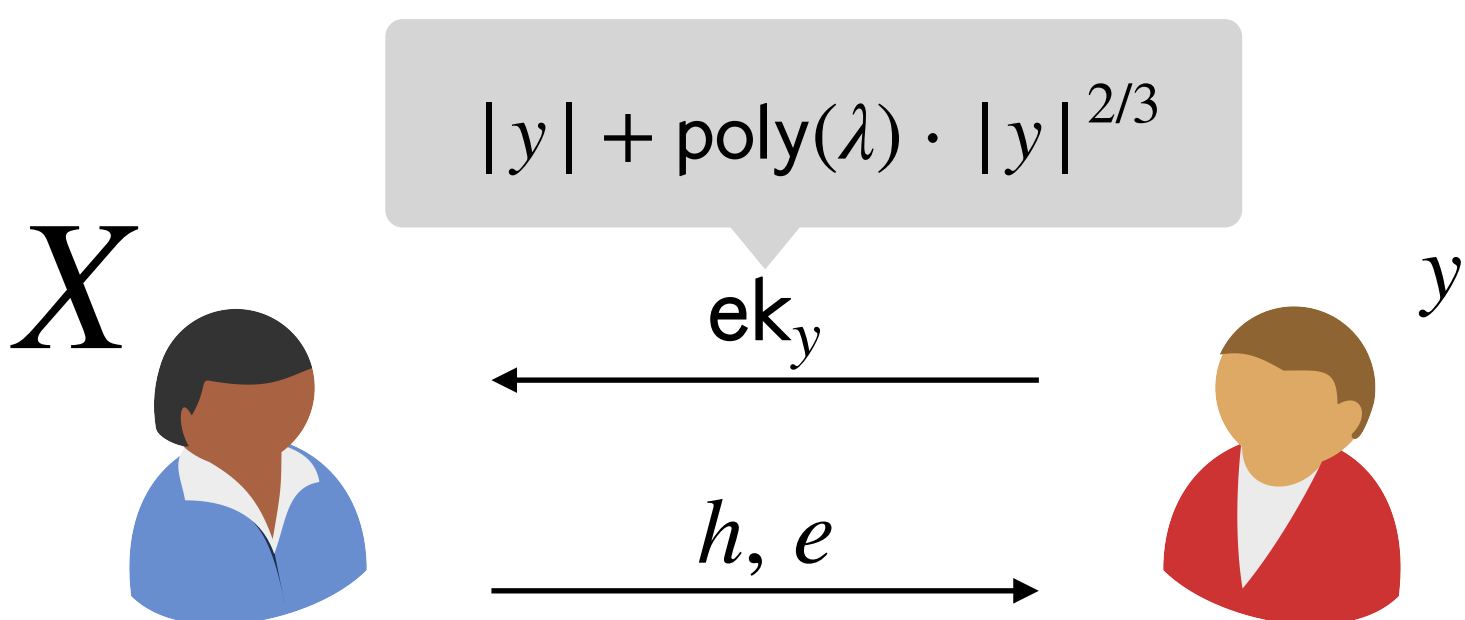
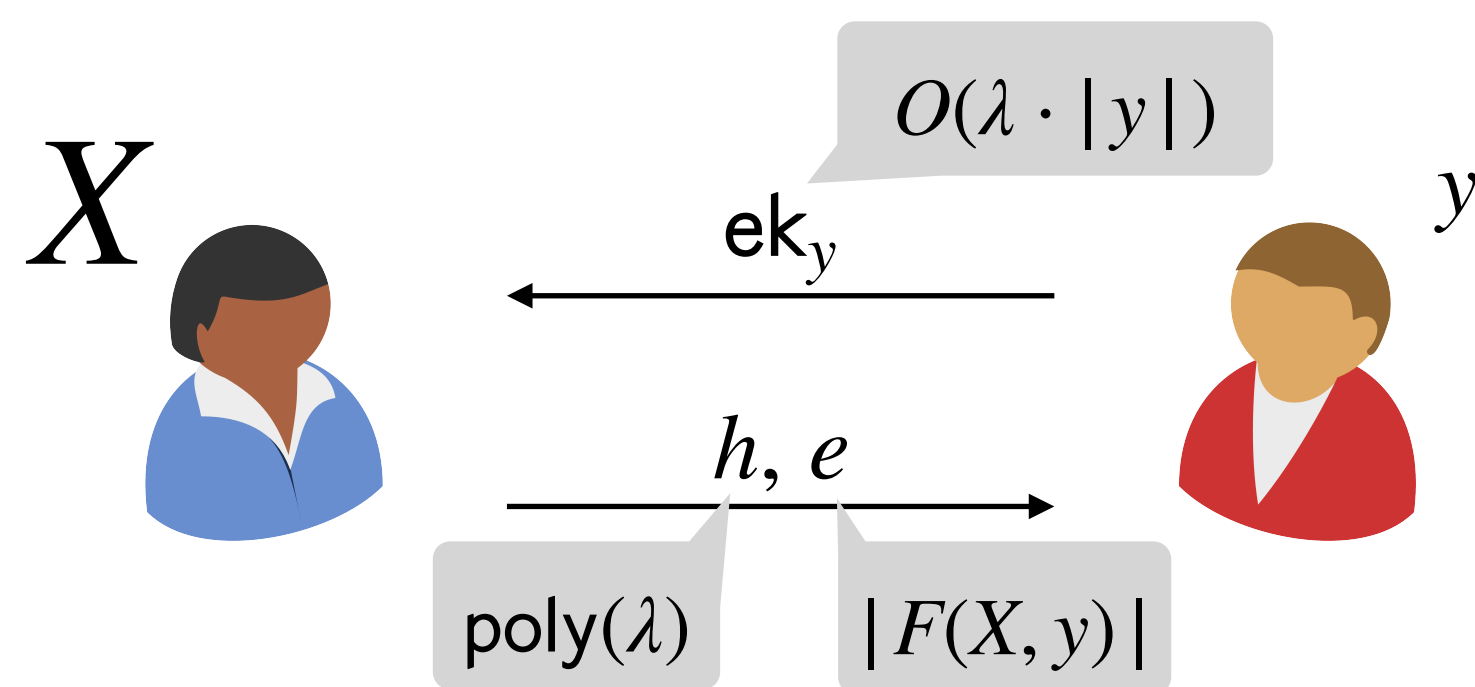
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F(X, y) = \sum_{i=1}^n f_i(X) \cdot g_i(y)$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

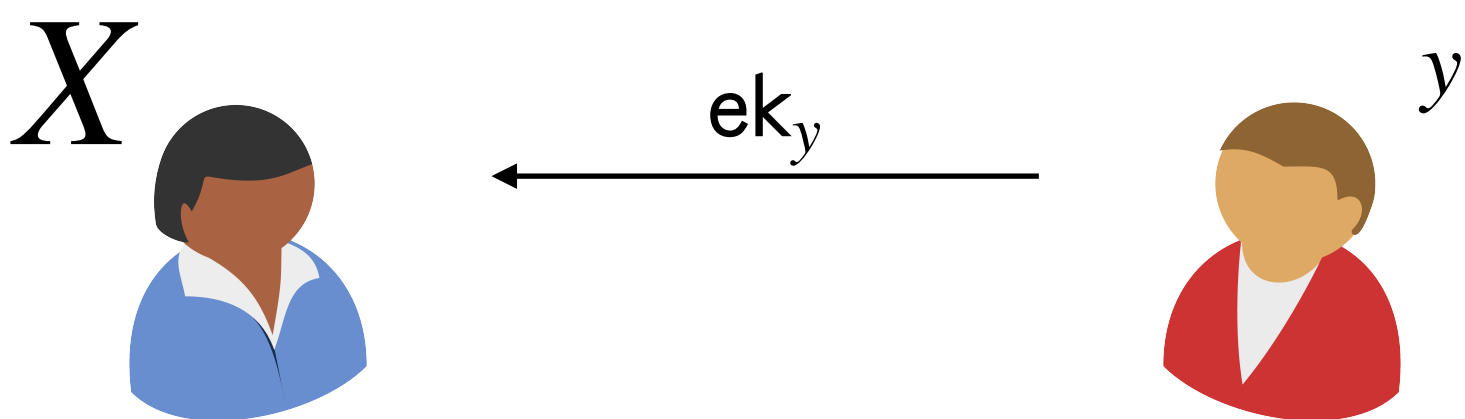
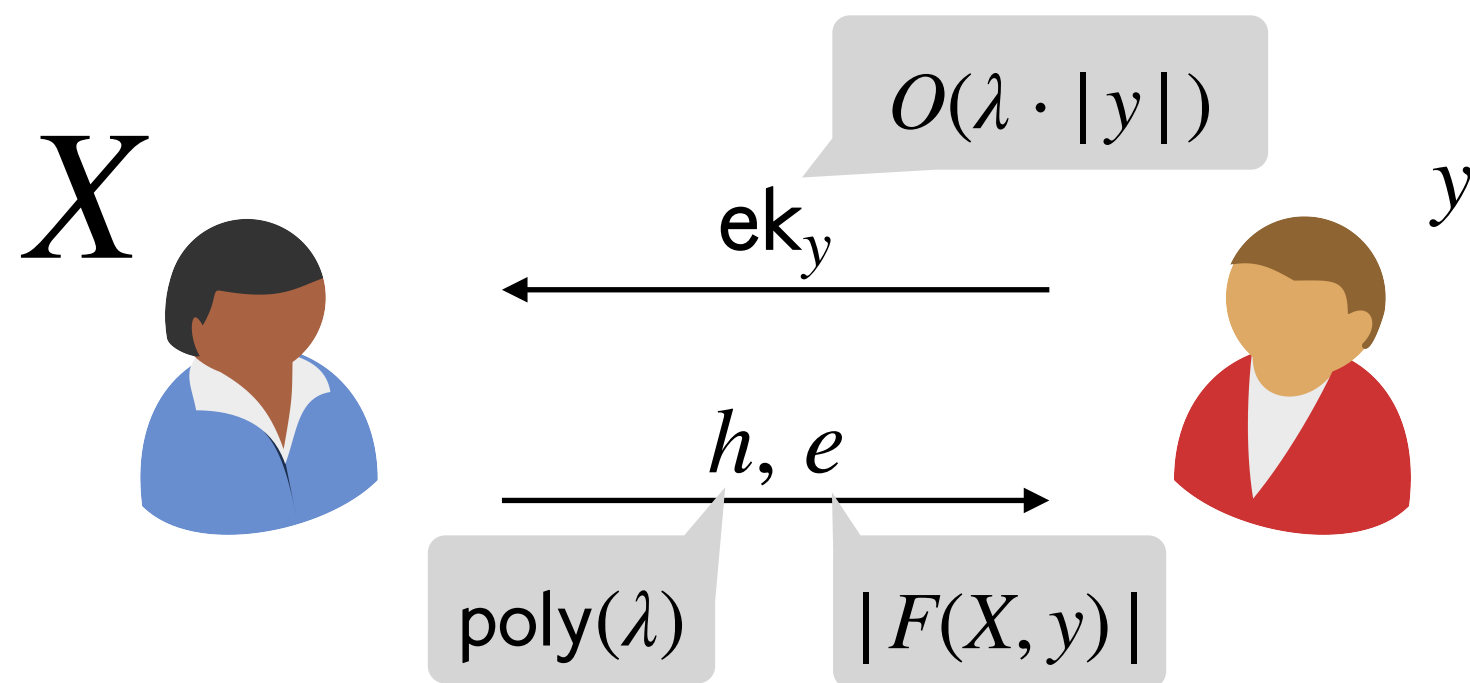
Reusable encoding key with
functions chosen on-the-fly

Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

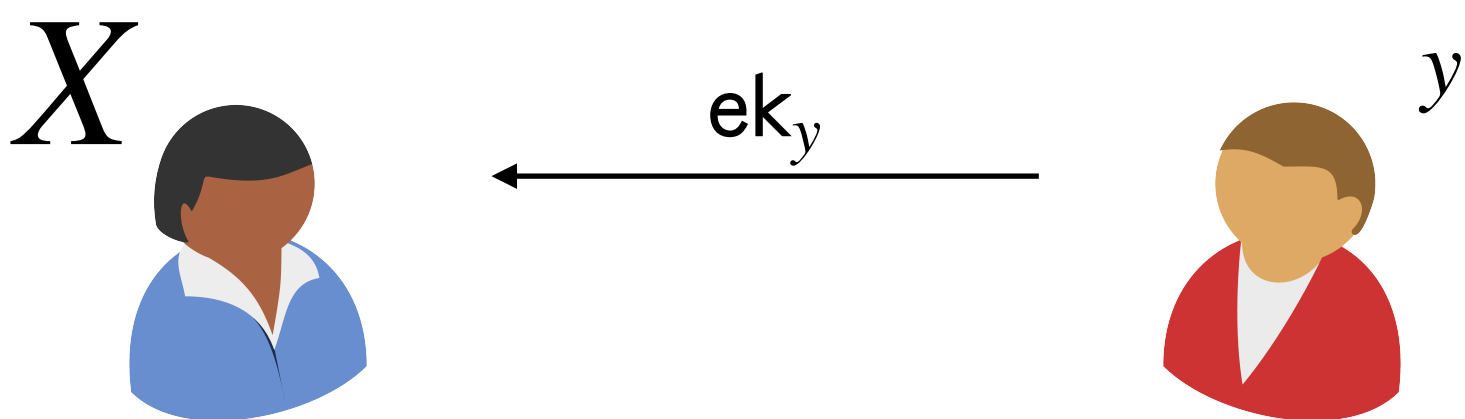
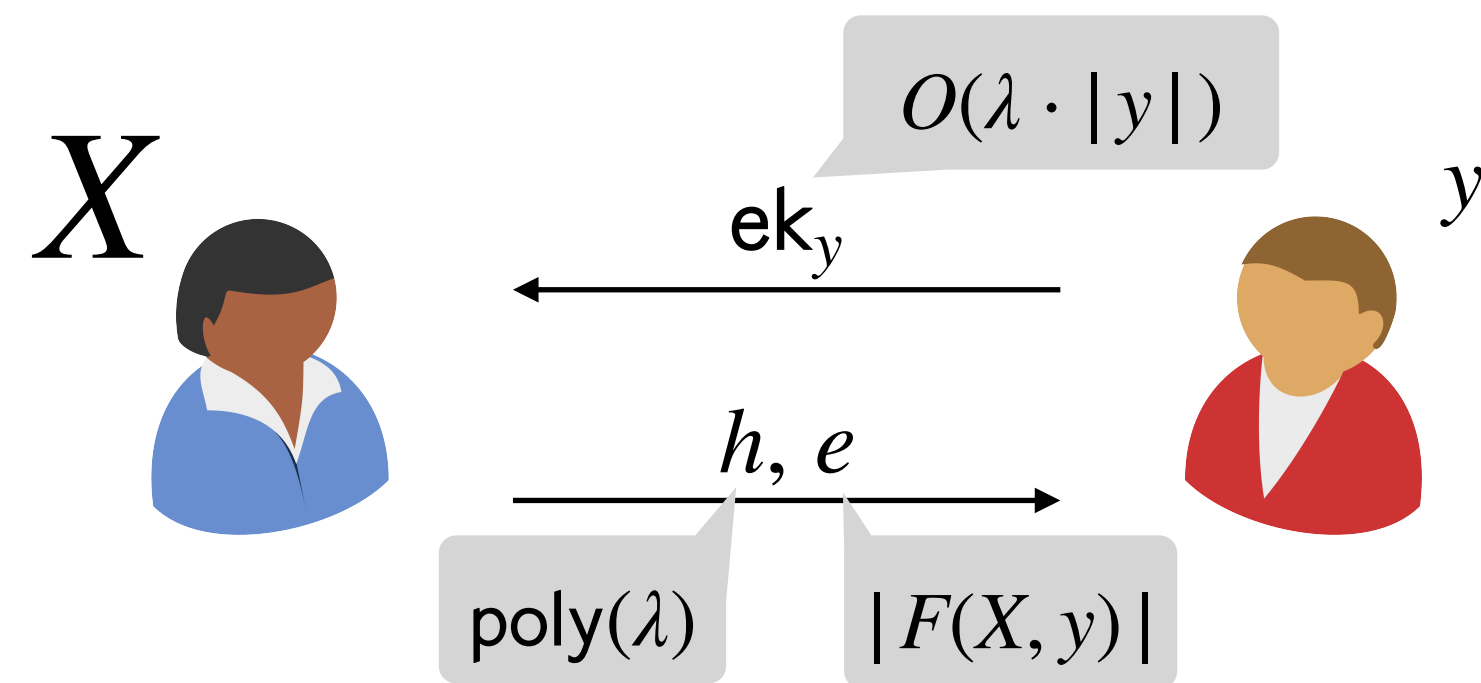
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F_1 \quad F_2 \quad F_3$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

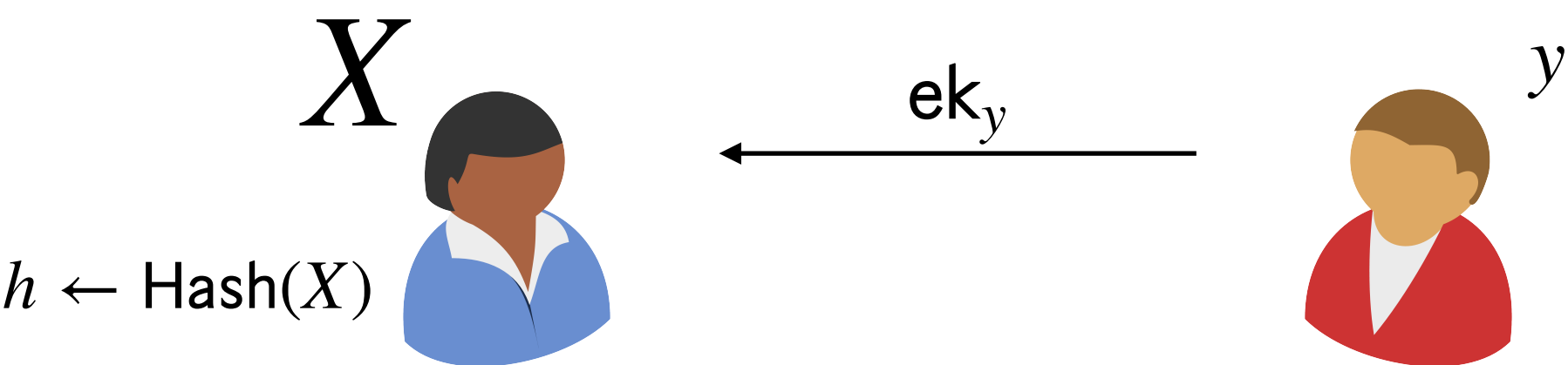
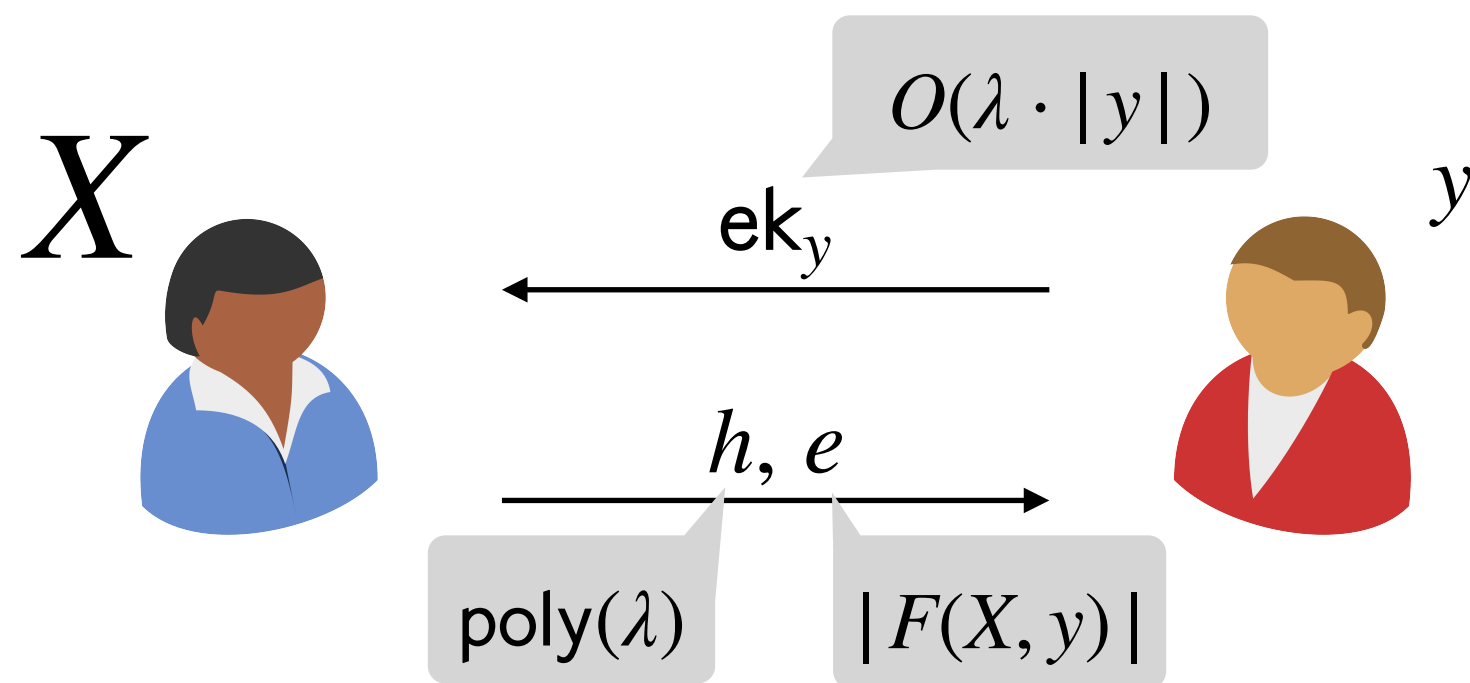
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F_1 \quad F_2 \quad F_3$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

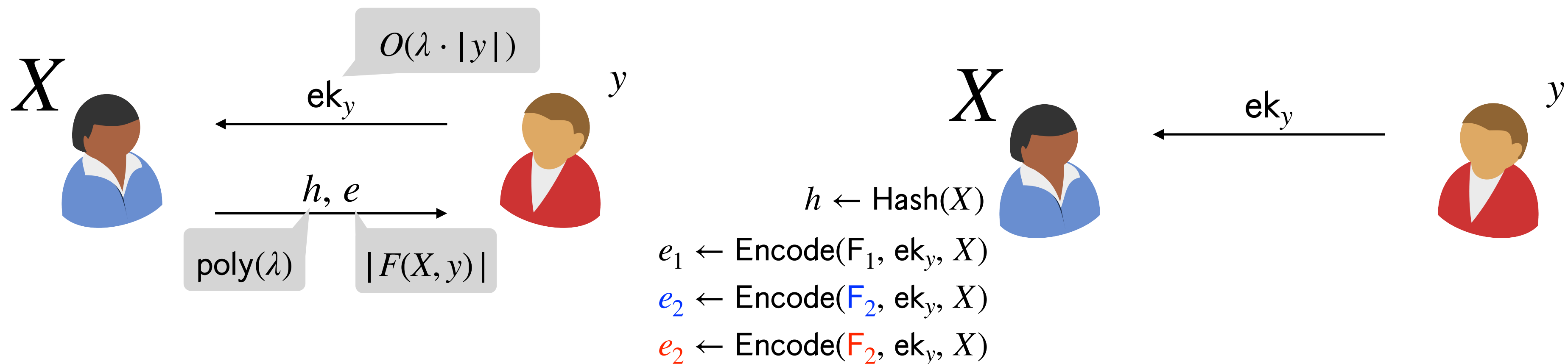
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

F_1 F_2 F_3



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

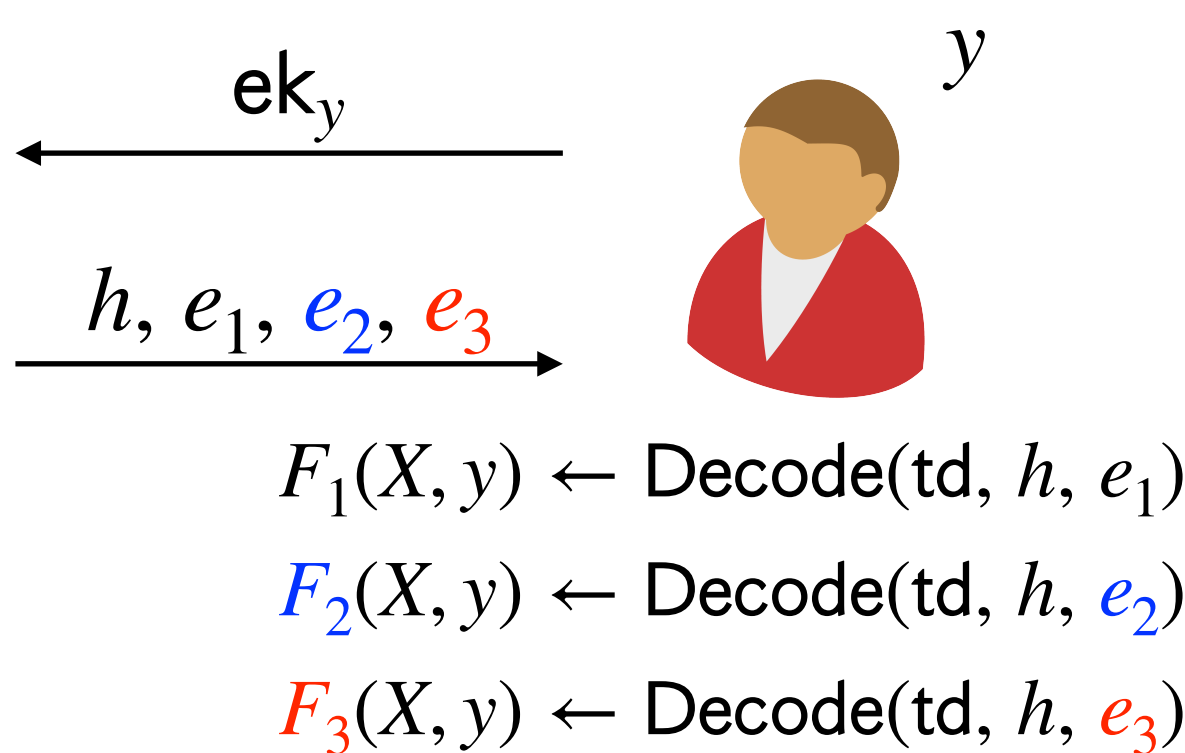
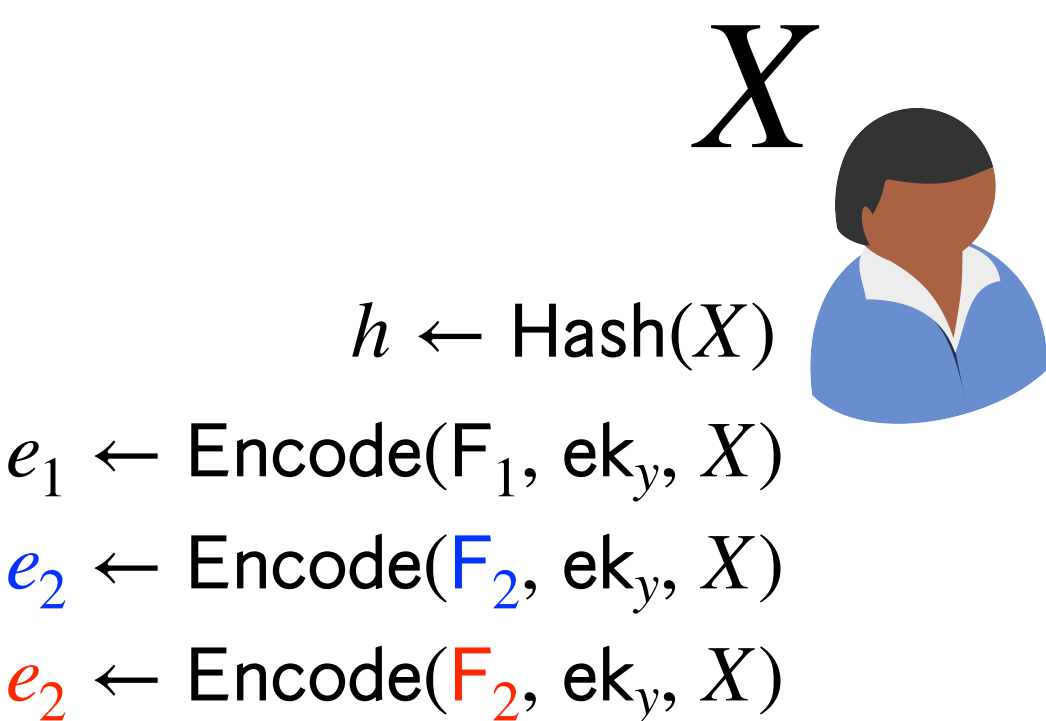
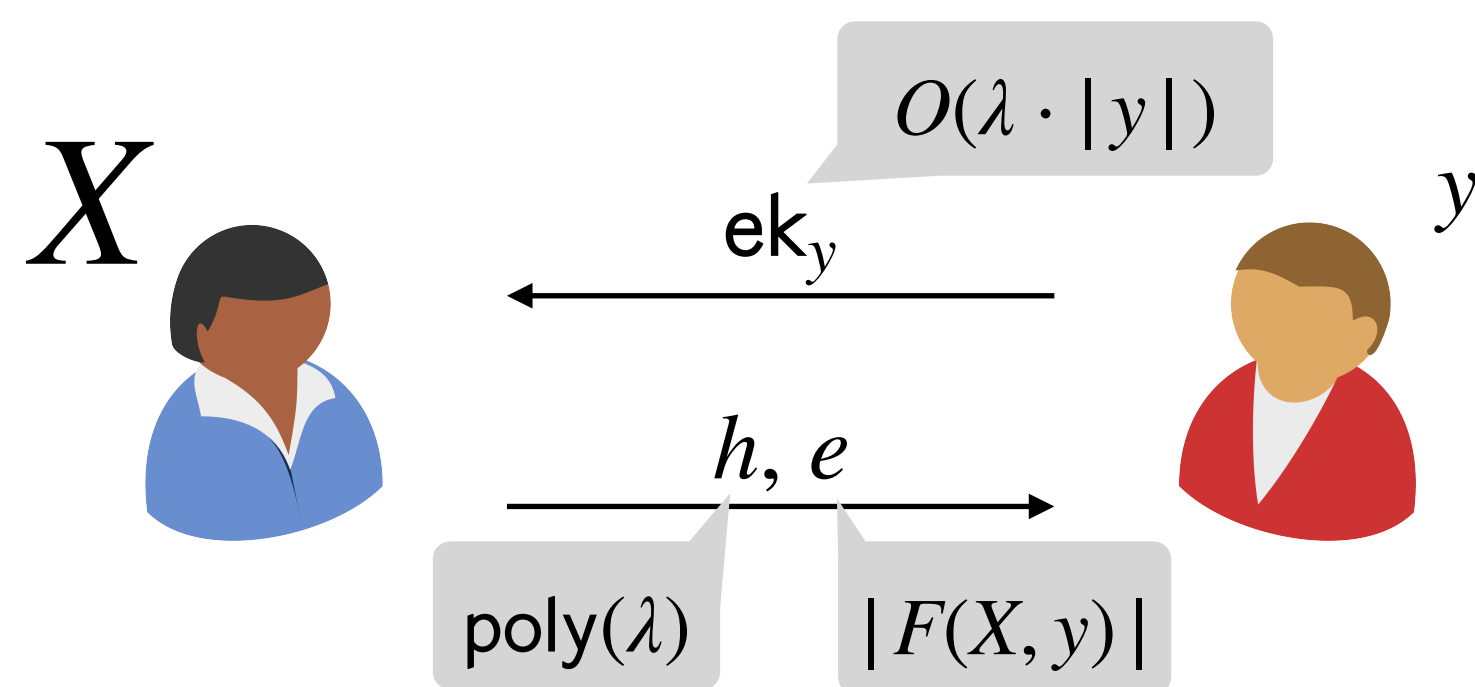
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F_1 \quad F_2 \quad F_3$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

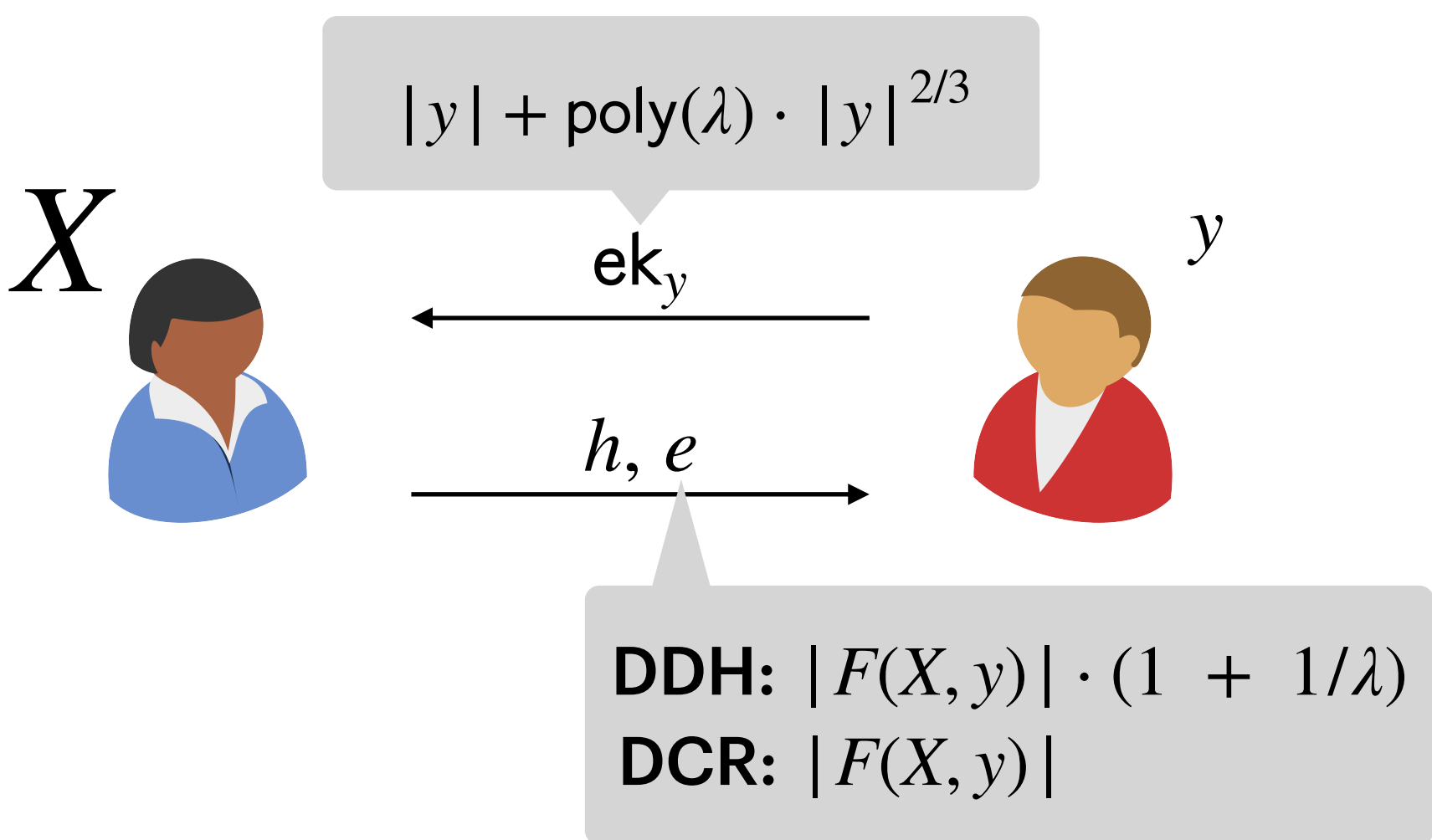
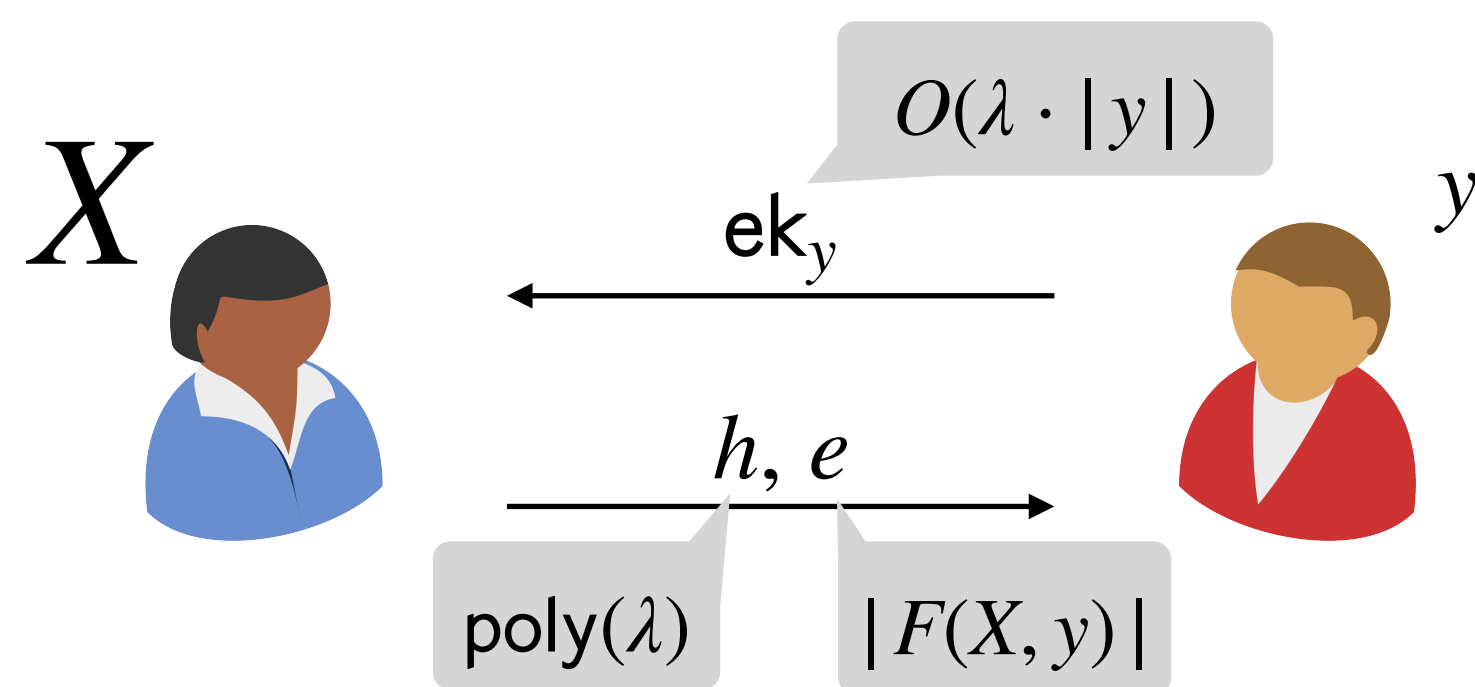
Enhanced Trapdoor Hash Functions from DDH and DCR

[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

This work

$$F(X, y) = \sum_i x_i \cdot y_i$$

$$F(X, y) = \sum_{i=1}^n f_i(X) \cdot g_i(y)$$



Expressivity

Supports computing
Bilinear-NC¹ programs

Compactness

Encoding keys of size
 $|y|(1 + o(1))$

Reusability

Reusable encoding key with
functions chosen on-the-fly

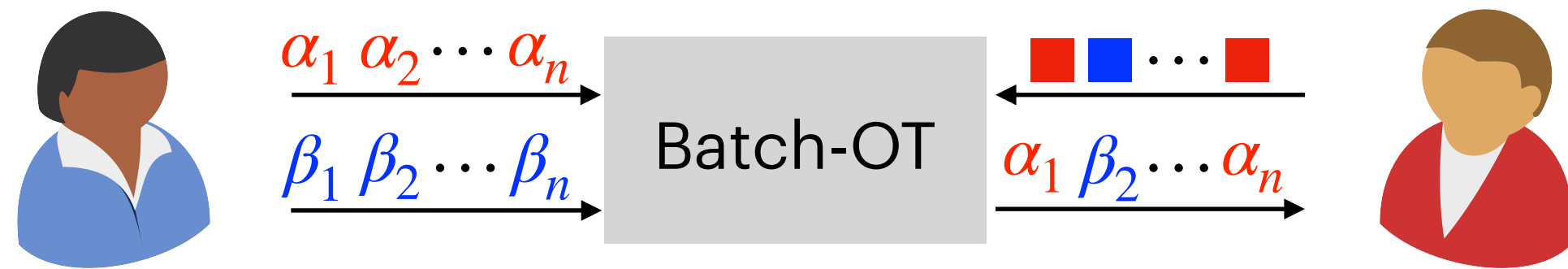
Applications 1: Compactness

Batch-OT with [optimal rate](#) from [DDH](#)



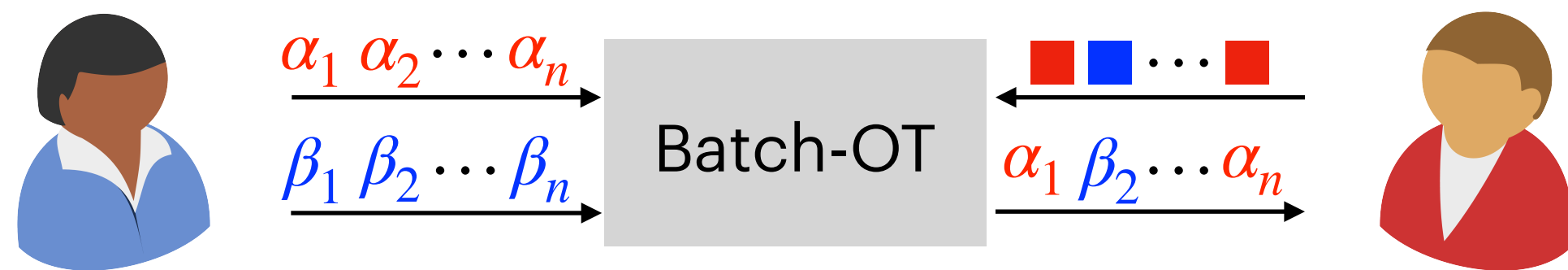
Applications 1: Compactness

Batch-OT with [optimal rate](#) from [DDH](#)



Applications 1: Compactness

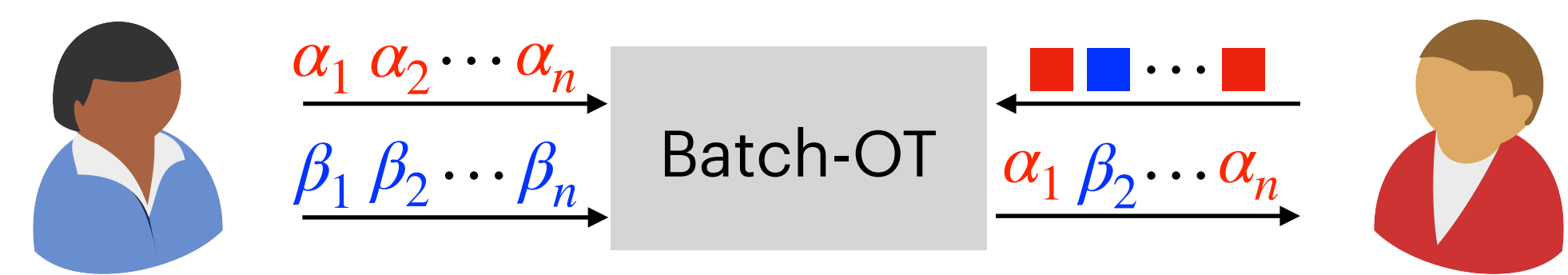
Batch-OT with [optimal rate](#) from [DDH](#)



Ideal World Communication: $2n$ bits

Applications 1: Compactness

Batch-OT with optimal rate from DDH

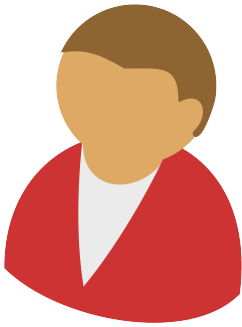


Ideal World Communication: $2n$ bits

$$X = \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$$

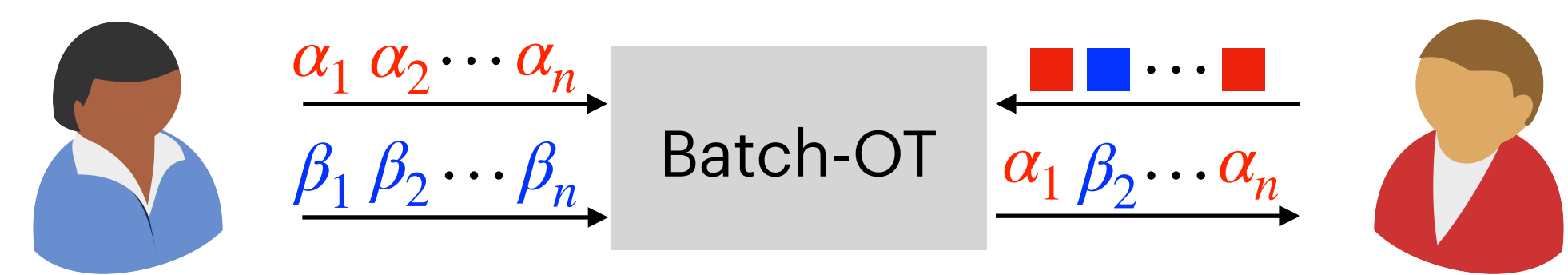


$$y = \text{red square}, \text{blue square}, \dots, \text{red square}$$

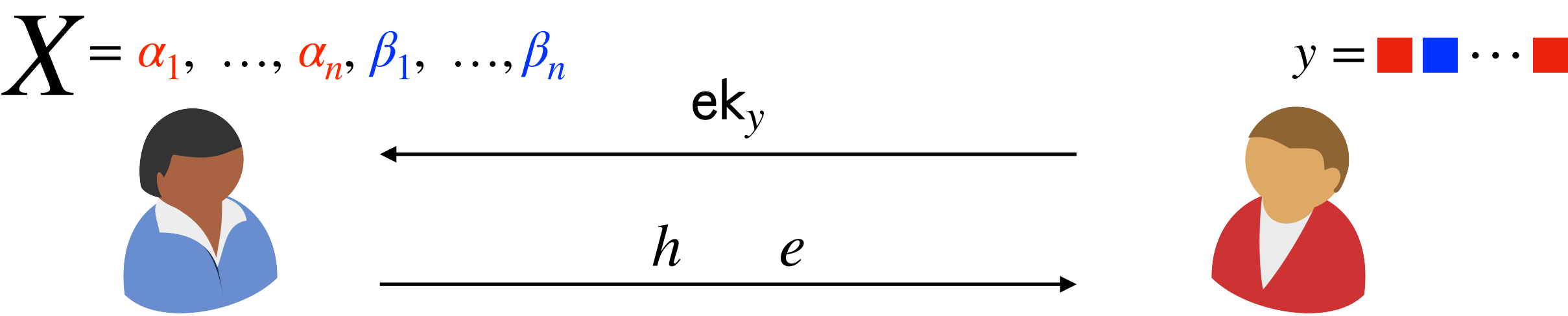


Applications 1: Compactness

Batch-OT with optimal rate from DDH

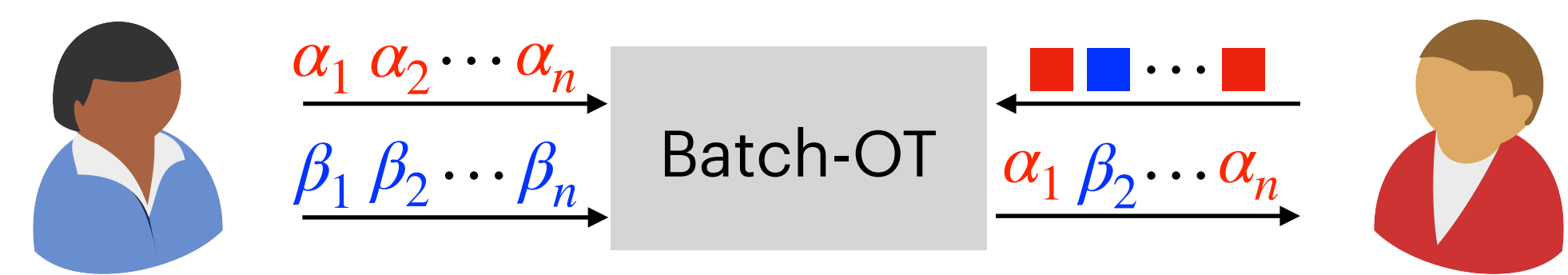


Ideal World Communication: $2n$ bits

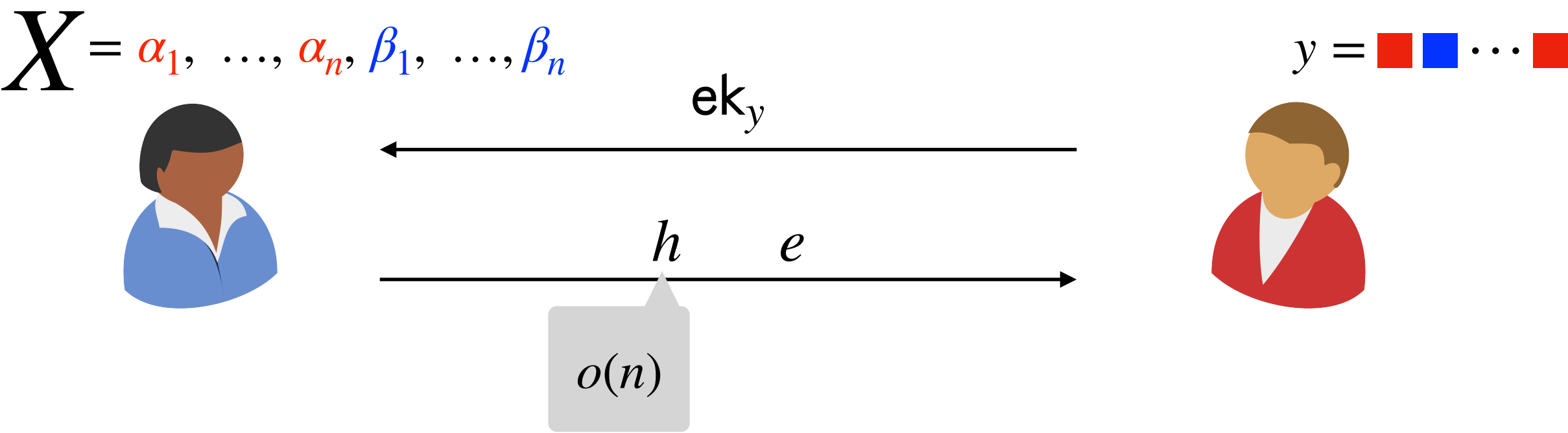


Applications 1: Compactness

Batch-OT with optimal rate from DDH

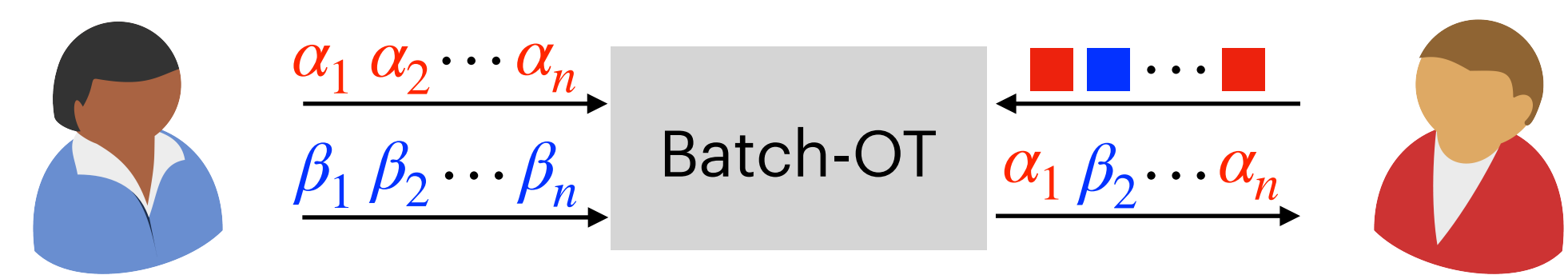


Ideal World Communication: $2n$ bits

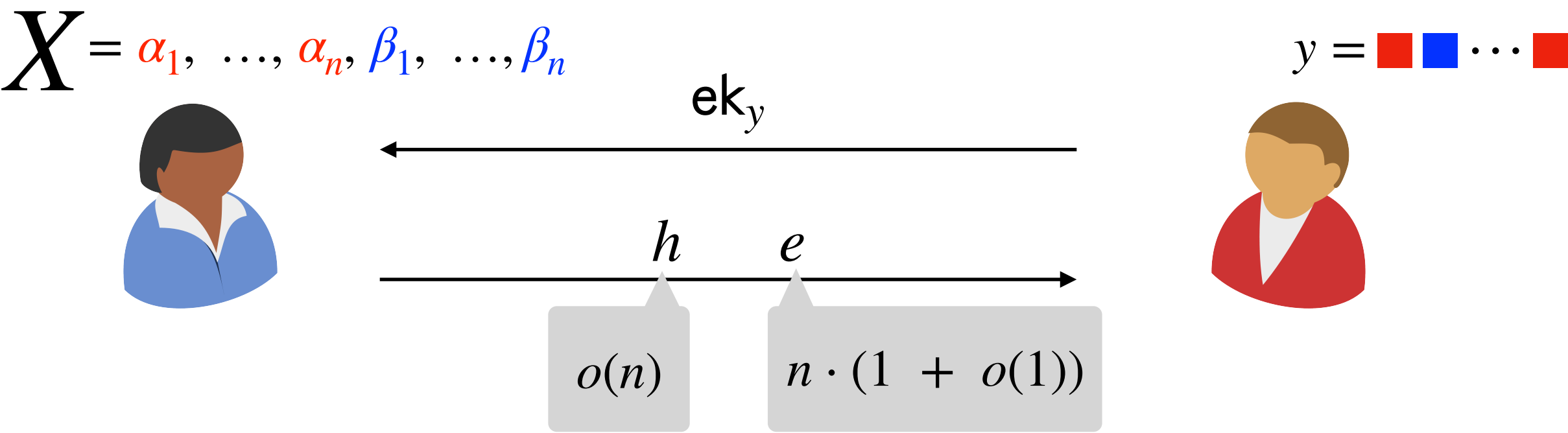


Applications 1: Compactness

Batch-OT with optimal rate from DDH

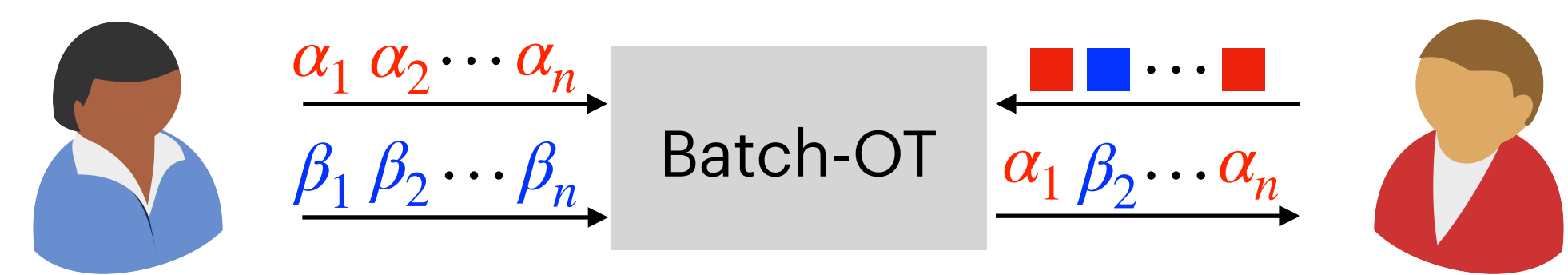


Ideal World Communication: $2n$ bits

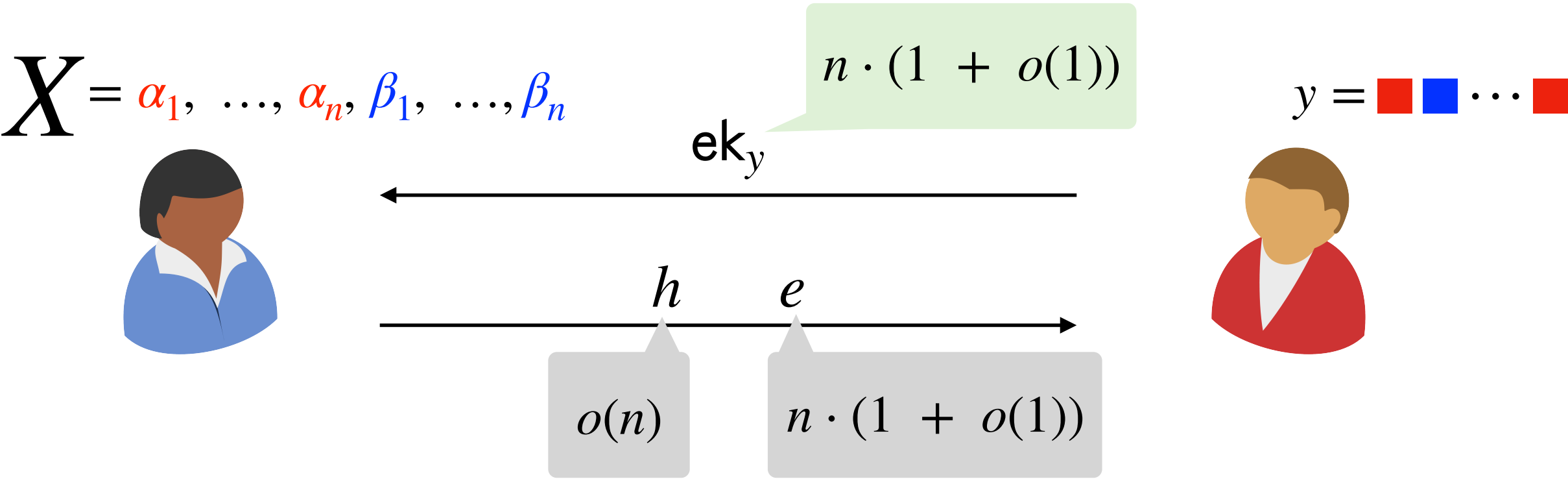


Applications 1: Compactness

Batch-OT with optimal rate from DDH

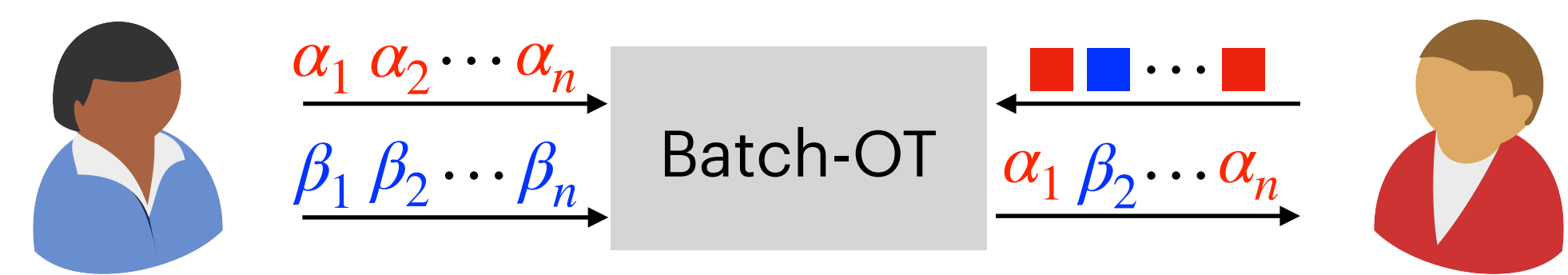


Ideal World Communication: $2n$ bits

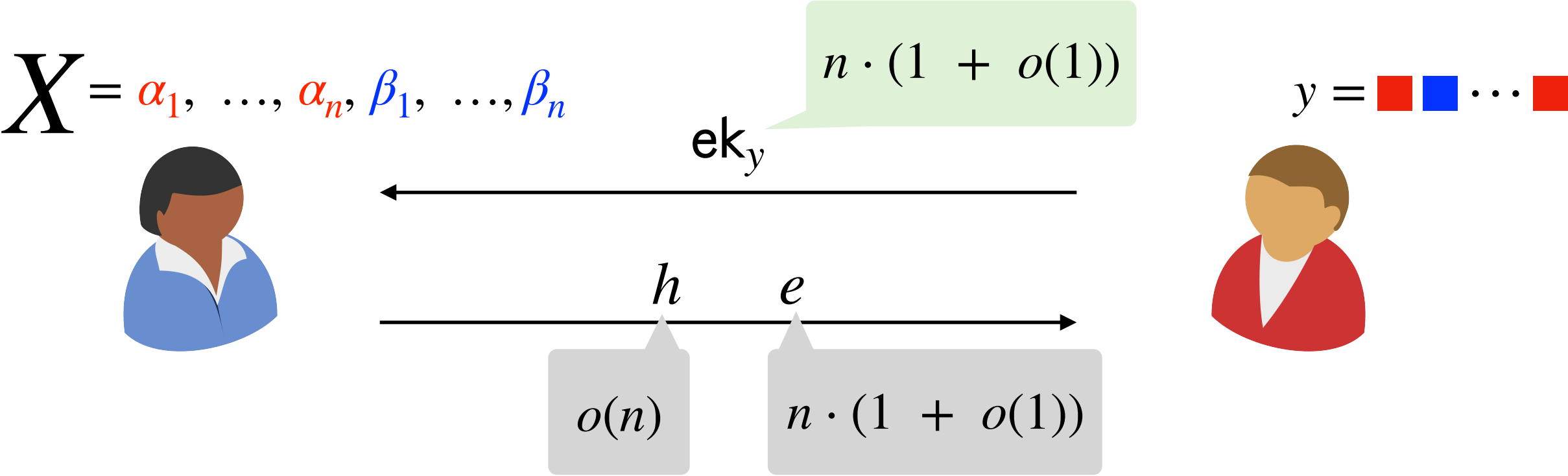


Applications 1: Compactness

Batch-OT with optimal rate from DDH



Ideal World Communication: $2n$ bits



Communication: $2 \cdot n \cdot (1 + o(1))$ bits

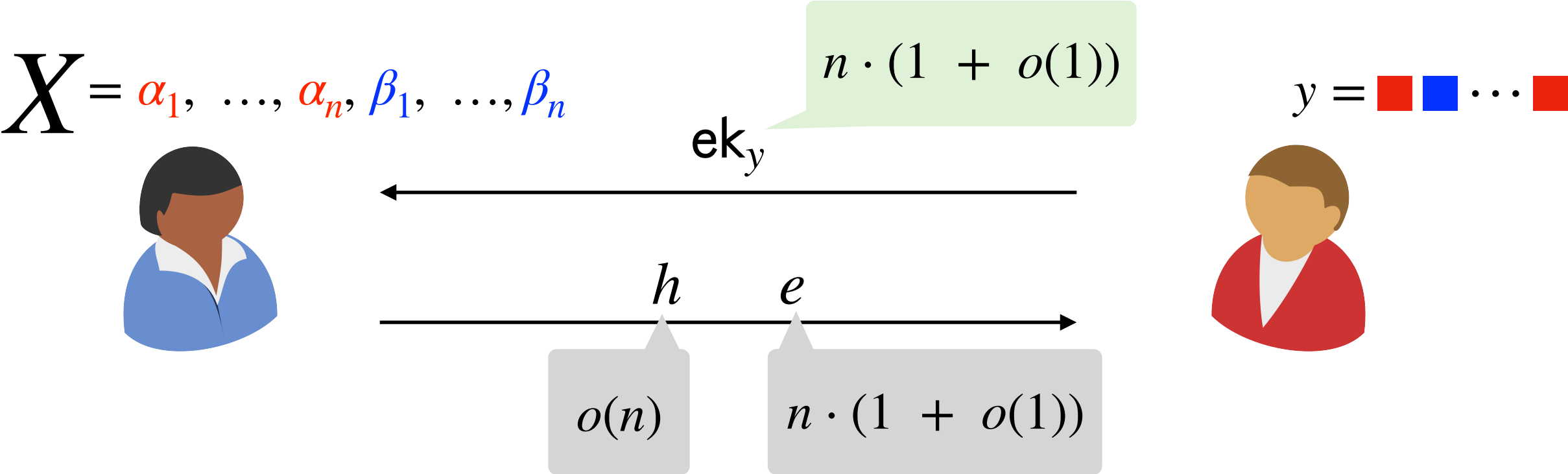
Applications 1: Compactness

Batch-OT with optimal rate from DDH



This Work: $1 + o(1)$ rate
Semi-honest statistical sender privacy

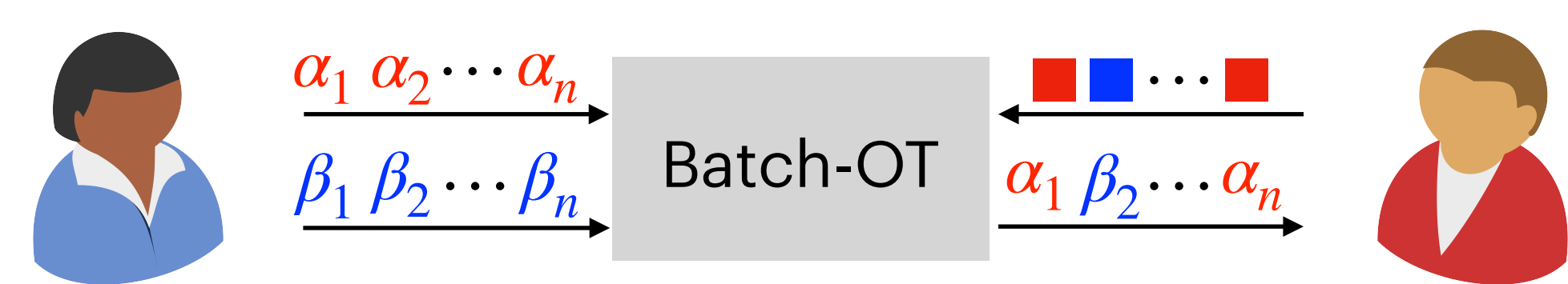
Ideal World Communication: $2n$ bits



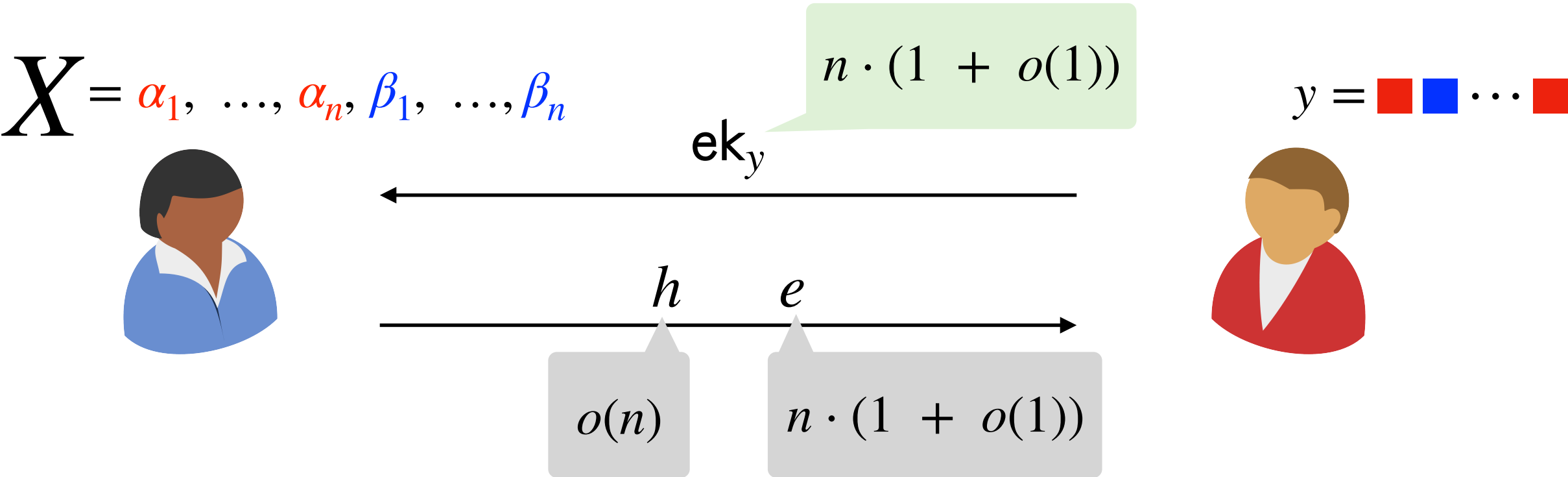
Communication: $2 \cdot n \cdot (1 + o(1))$ bits

Applications 1: Compactness

Batch-OT with optimal rate from DDH



Ideal World Communication: $2n$ bits



Communication: $2 \cdot n \cdot (1 + o(1))$ bits

This Work: $1 + o(1)$ rate
Semi-honest statistical sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]:
 $1 + o(1)$ rate
DDH + LPN

[Boyle-Giboa-Ishai'17]:
DDH
 $n \cdot (4 + o(1))$ bits communication
PKI setup

Applications 1: Compactness

Batch-OT with **optimal rate** from **DDH**

This Work: $1 + o(1)$ rate

Semi-honest **statistical** sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + **LPN**

[Boyle-Giboa-Ishai'17]:

DDH

$n \cdot (4 + o(1))$ bits communication

PKI setup

Applications 1: Compactness

Batch-OT with **optimal rate** from **DDH**

This Work: $1 + o(1)$ rate

Semi-honest **statistical** sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + **LPN**

[Boyle-Giboa-Ishai'17]:

DDH

$n \cdot (4 + o(1))$ bits communication

PKI setup

Implications of Batch-OT with optimal rate

Applications 1: Compactness

Batch-OT with optimal rate from DDH

This Work: $1 + o(1)$ rate

Semi-honest statistical sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + LPN

[Boyle-Giboa-Ishai'17]: DDH

$n \cdot (4 + o(1))$ bits communication

PKI setup

Implications of Batch-OT with optimal rate

String OT: $o(n)$ bits sender-to-receiver communication and $n \cdot (1 + o(1))$ bits receiver-to-sender communication

Applications 1: Compactness

Batch-OT with **optimal rate** from **DDH**

This Work: $1 + o(1)$ rate

Semi-honest **statistical** sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + **LPN**

[Boyle-Giboa-Ishai'17]: **DDH**

$n \cdot (4 + o(1))$ bits communication

PKI setup

Implications of Batch-OT with optimal rate

String OT: $o(n)$ bits sender-to-receiver communication and $n \cdot (1 + o(1))$ bits receiver-to-sender communication

Lossy Trapdoor Functions (LTDF): Rate-1 LTDF with public key size $n \cdot (1 + o(1))$ bits

Rate-1 LTDF with public key size $o(n)$ bits with **CRS**

Applications 1: Compactness

Batch-OT with **optimal rate** from **DDH**

This Work: $1 + o(1)$ rate

Semi-honest **statistical** sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + **LPN**

[Boyle-Giboa-Ishai'17]: **DDH**

$n \cdot (4 + o(1))$ bits communication

PKI setup

Implications of Batch-OT with optimal rate

String OT: $o(n)$ bits **sender-to-receiver communication** and $n \cdot (1 + o(1))$ bits receiver-to-sender communication

Lossy Trapdoor Functions (LTDF): Rate-1 LTDF with public key size $n \cdot (1 + o(1))$ bits

Rate-1 LTDF with public key size $o(n)$ bits with **CRS**

Private Information Retrieval: Client computation $\text{poly}(n, \lambda)$

Database size: 2^n

Upload communication $n + \text{poly}(\lambda)$ bits

Download communication $n \cdot \text{poly}(\lambda)$ bits

Applications 1: Compactness

Batch-OT with **optimal rate** from **DDH**

This Work: $1 + o(1)$ rate

Semi-honest **statistical** sender privacy

Before:

[Brakerski-Branco-Döttling-Pu'22]: $1 + o(1)$ rate

DDH + **LPN**

[Boyle-Giboa-Ishai'17]: **DDH**

$n \cdot (4 + o(1))$ bits communication

PKI setup

Implications of Batch-OT with optimal rate

String OT: $o(n)$ bits **sender-to-receiver communication** and $n \cdot (1 + o(1))$ bits receiver-to-sender communication

Lossy Trapdoor Functions (LTDF): Rate-1 LTDF with public key size $n \cdot (1 + o(1))$ bits

Rate-1 LTDF with public key size $o(n)$ bits with **CRS**

Private Information Retrieval: Client computation $\text{poly}(n, \lambda)$

Database size: 2^n

Upload communication $n + \text{poly}(\lambda)$ bits

Download communication $n \cdot \text{poly}(\lambda)$ bits

Other Implications

Branching programs over encrypted data

Correlated symmetric PIR

Applications 2: Compactness + **Expressivity**

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Applications 2: Compactness + **Expressivity**

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

Applications 2: Compactness + **Expressivity**

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

This Work: $|x| + (2 + o(1)) \cdot \frac{|C|}{\log \log |C|} + |y|^{2/3} \cdot \text{poly}(\lambda)$ bits communication

Applications 2: Compactness + **Expressivity**

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

Linear communication in
computationally secure input

Sublinear communication in
statistically secure input

This Work: $|x| + (2 + o(1)) \cdot \frac{|C|}{\log \log |C|} + |y|^{2/3} \cdot \text{poly}(\lambda)$ bits communication

Applications 2: Compactness + **Expressivity**

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

Linear communication in
computationally secure input

Sublinear communication in
statistically secure input

This Work: $|x| + (2 + o(1)) \cdot \frac{|C|}{\log \log |C|} + |y|^{2/3} \cdot \text{poly}(\lambda)$ bits communication

Sublinear in size of circuit

Applications 2: Compactness + Expressivity

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

Linear communication in
computationally secure input

Sublinear communication in
statistically secure input

This Work: $|x| + (2 + o(1)) \cdot \frac{|C|}{\log \log |C|} + |y|^{2/3} \cdot \text{poly}(\lambda)$ bits communication

Sublinear in size of circuit

Before: Similar results only known from FHE

Applications 2: Compactness + Expressivity

Sublinear 2PC from DCR, with one-sided statistical security for layered circuits

Bilinear-NC¹ $\supseteq$ log log-depth circuits

Linear communication in
computationally secure input

Sublinear communication in
statistically secure input

This Work: $|x| + (2 + o(1)) \cdot \frac{|C|}{\log \log |C|} + |y|^{2/3} \cdot \text{poly}(\lambda)$ bits communication

Sublinear in size of circuit

Before: Similar results only known from FHE

[Couteau-Meyer-Passelégue-Riahinia'23]: $|x| + |y| + \frac{|C|}{\log \log |C|} + \text{poly}(\lambda)$ bits communication

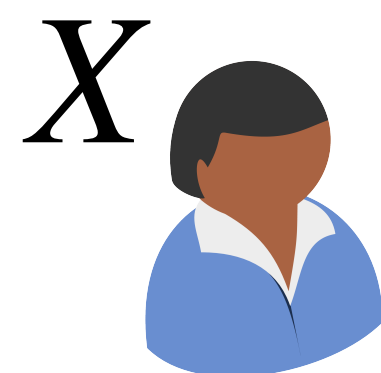
Circular security of Paillier

Layered circuits over $\mathbb{Z}_N$

Applications 3: Compactness + Expressivity + Reusability

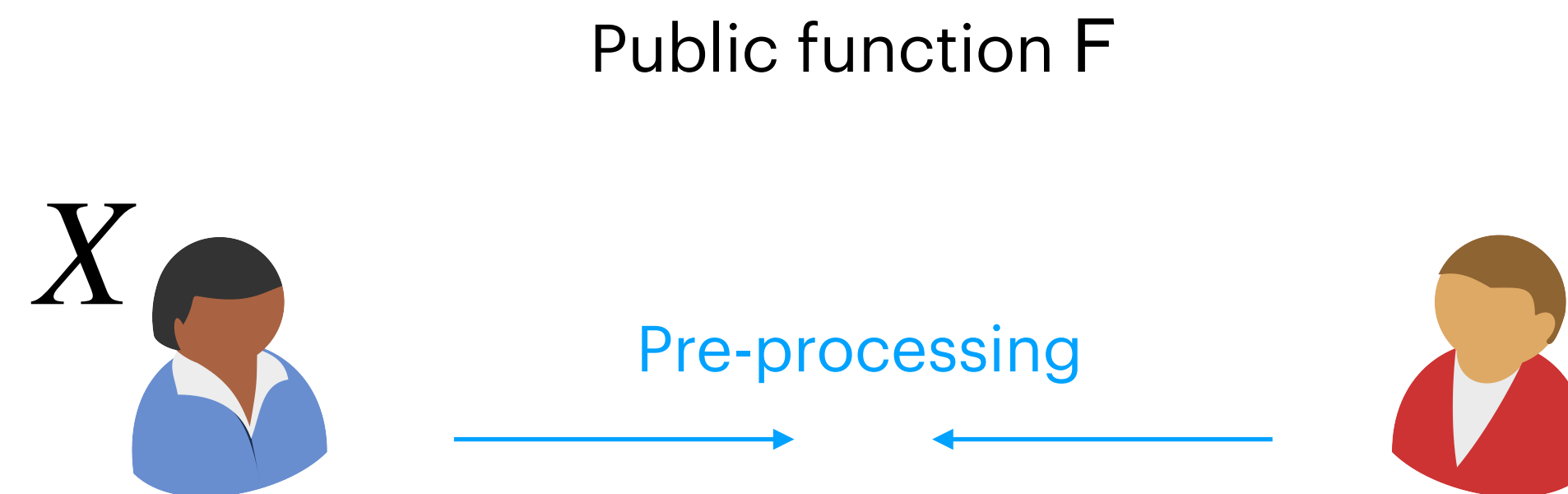
Improving Communication in the Amortized Setting

Public function F



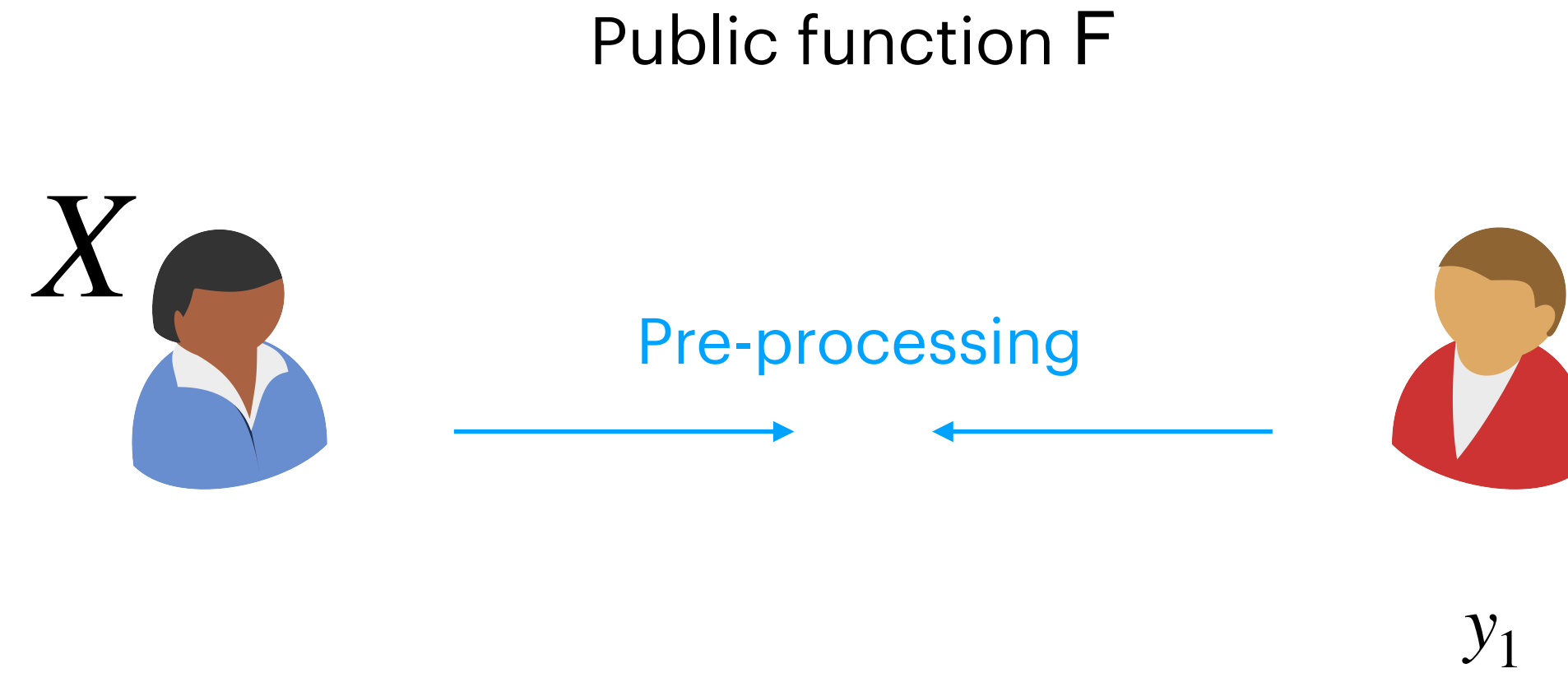
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



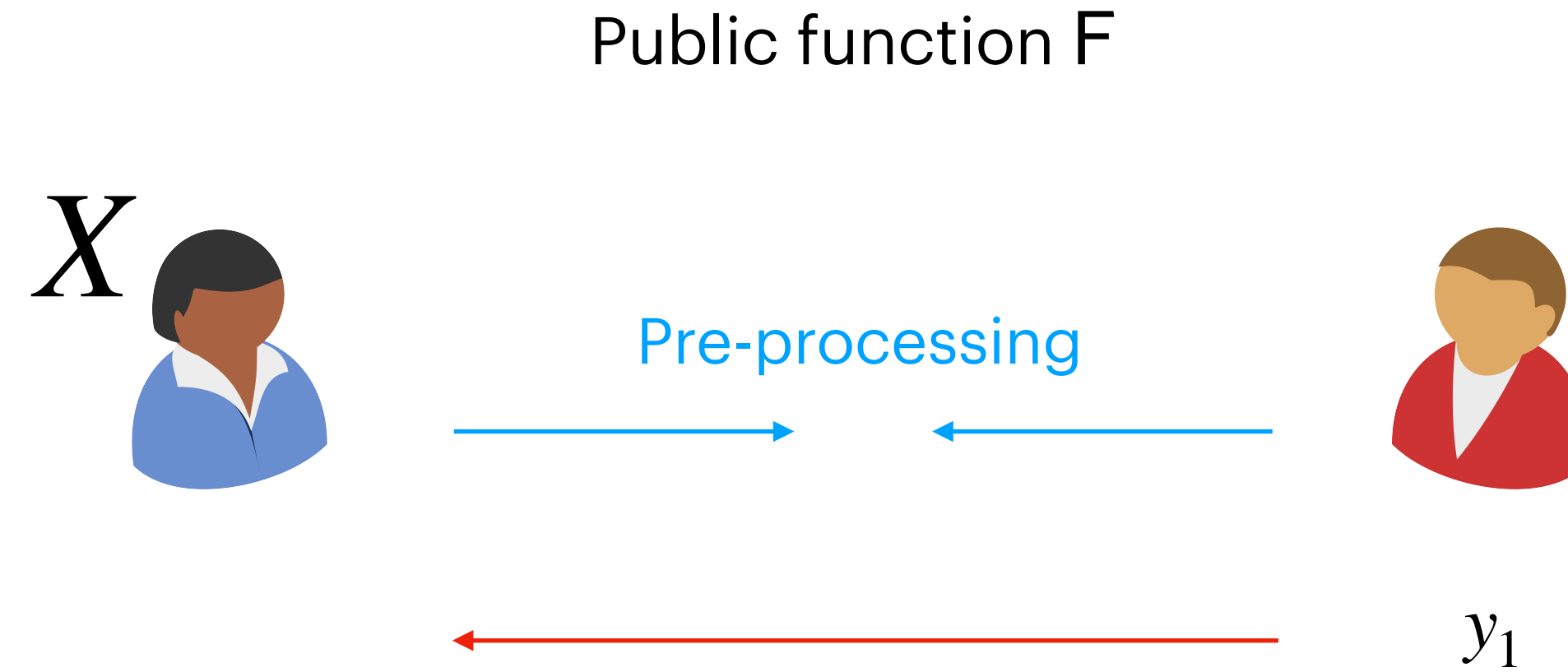
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



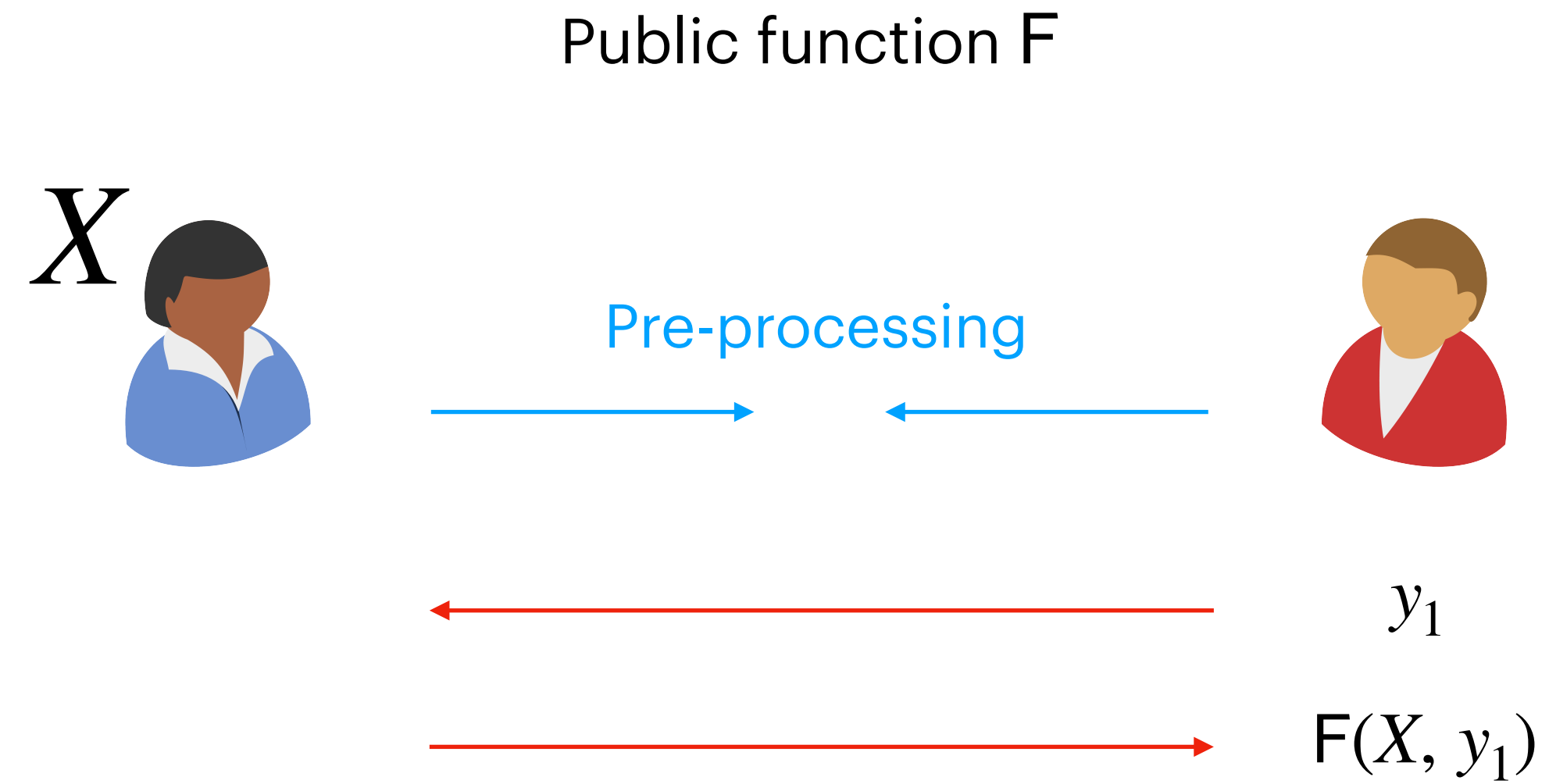
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



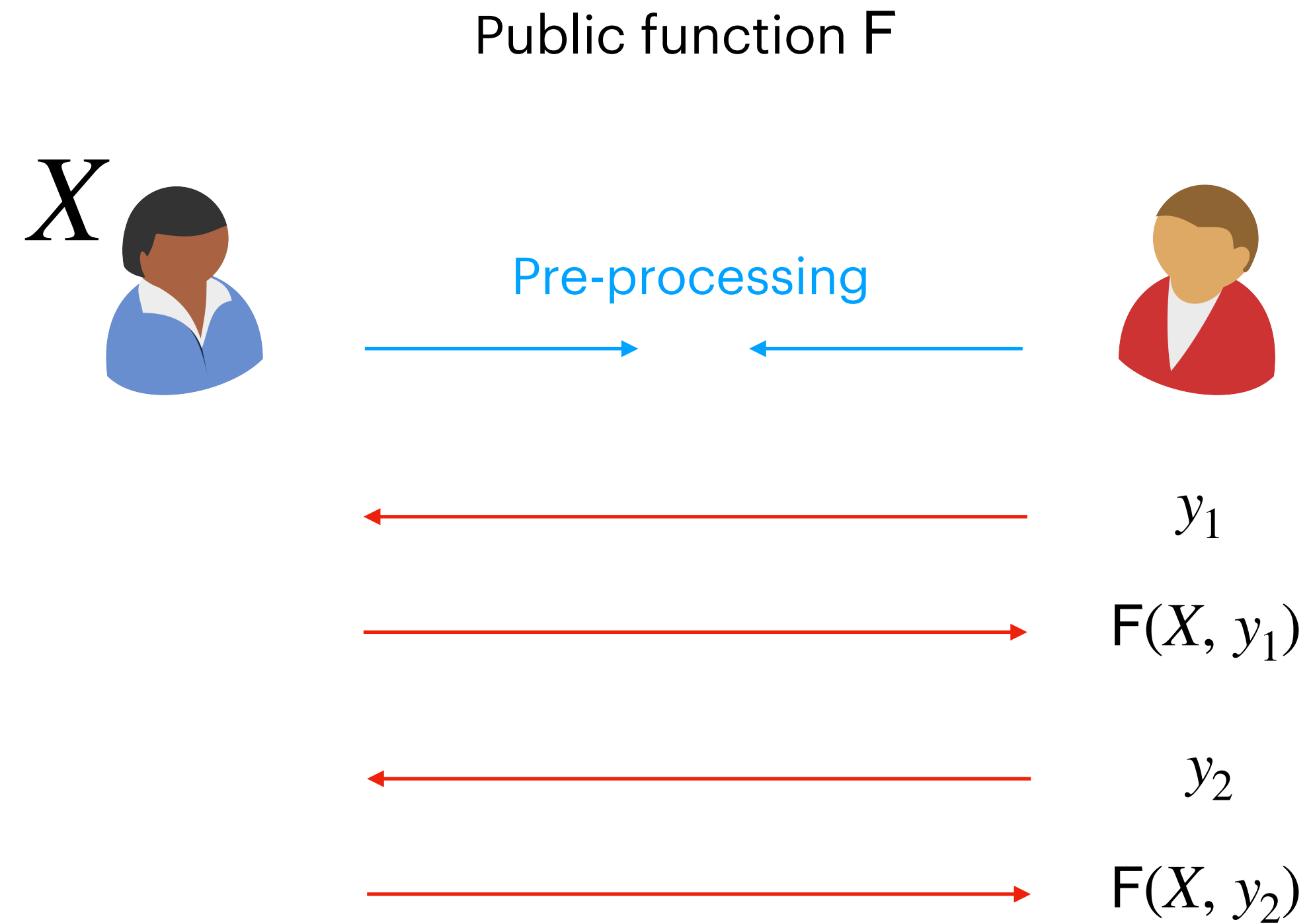
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



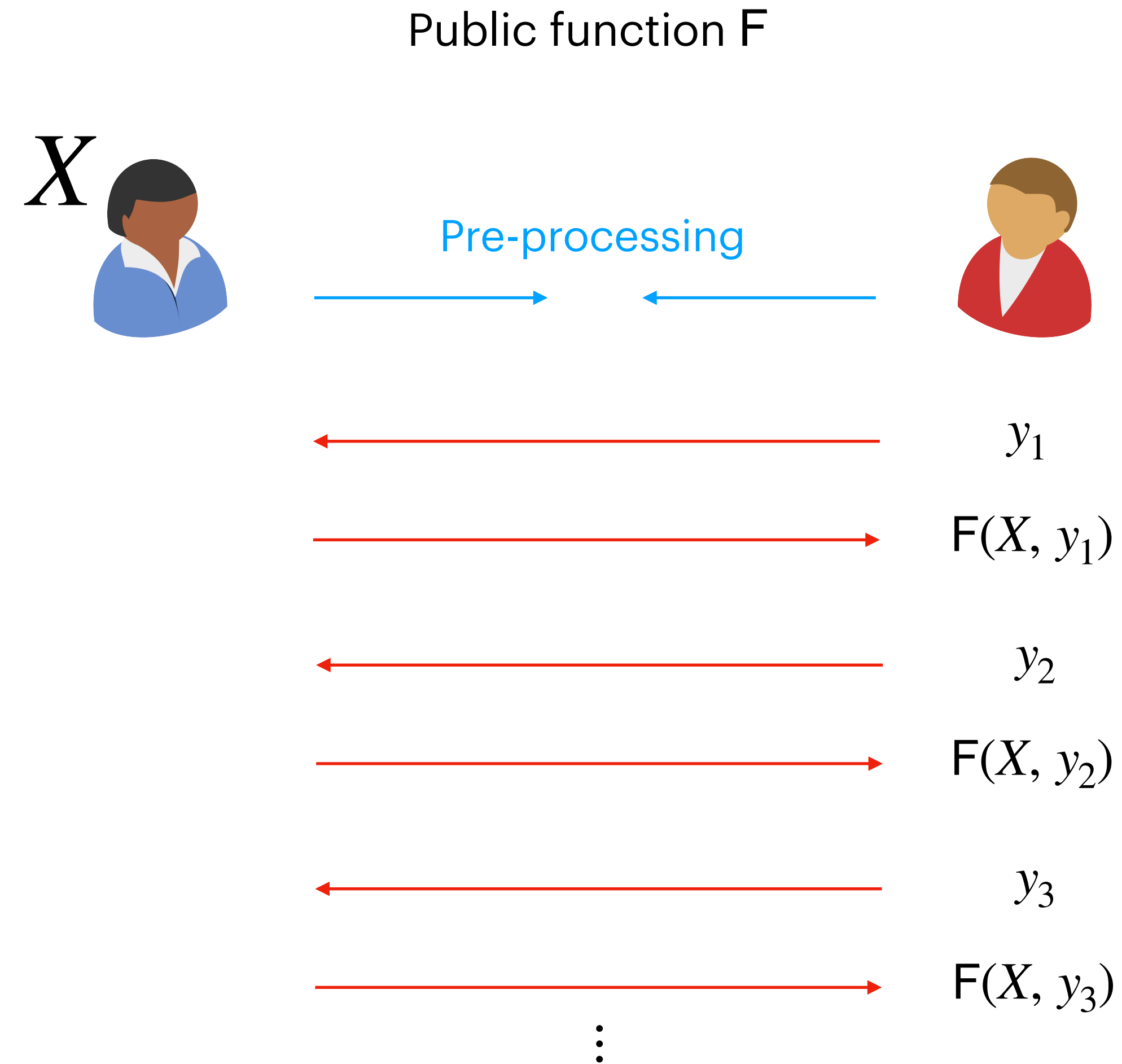
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



Applications 3: Compactness + Expressivity + Reusability

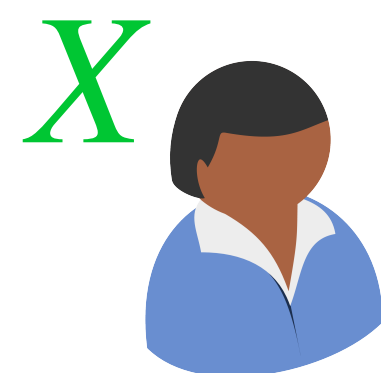
Improving Communication in the Amortized Setting



Applications 3: Compactness + Expressivity + Reusability

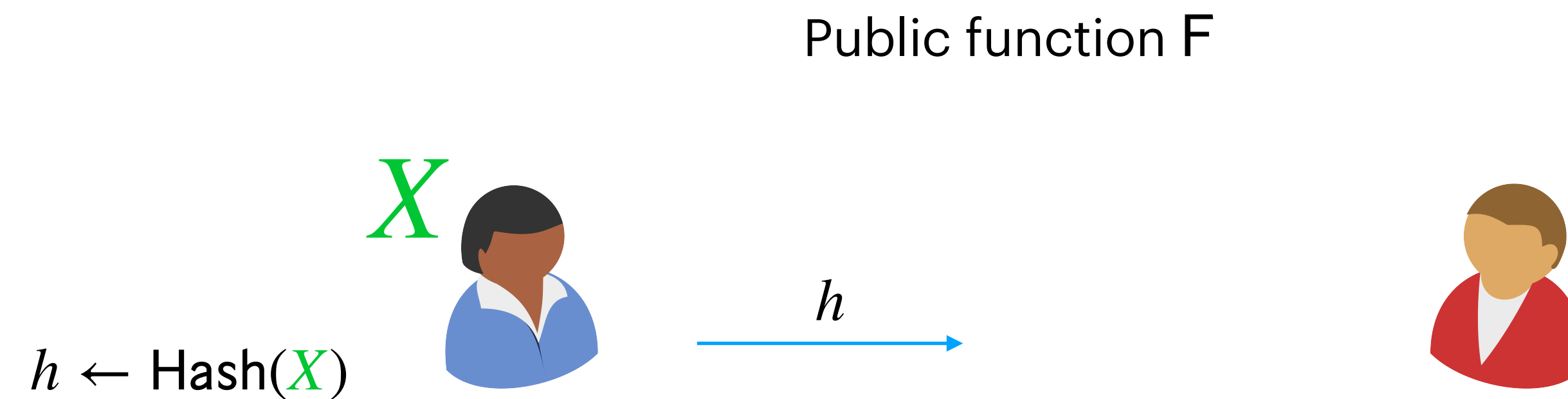
Improving Communication in the Amortized Setting

Public function F



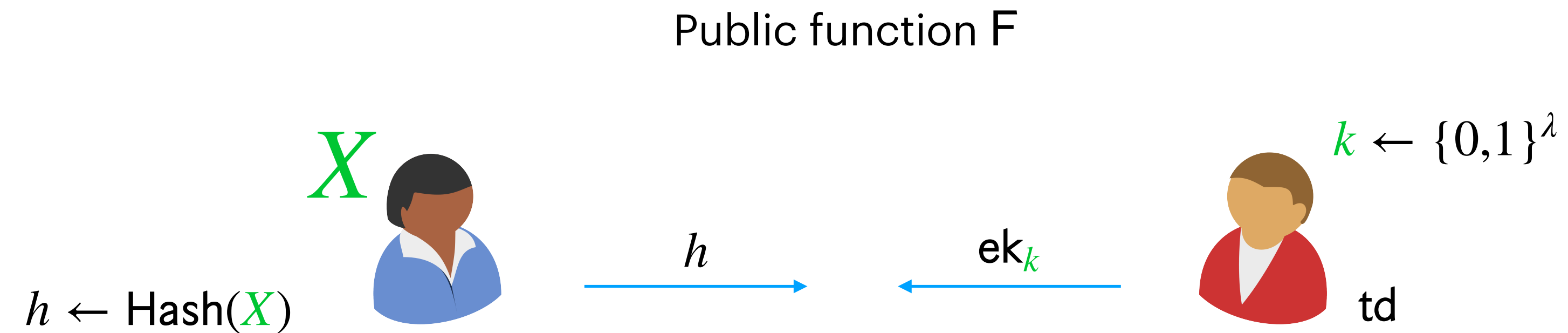
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



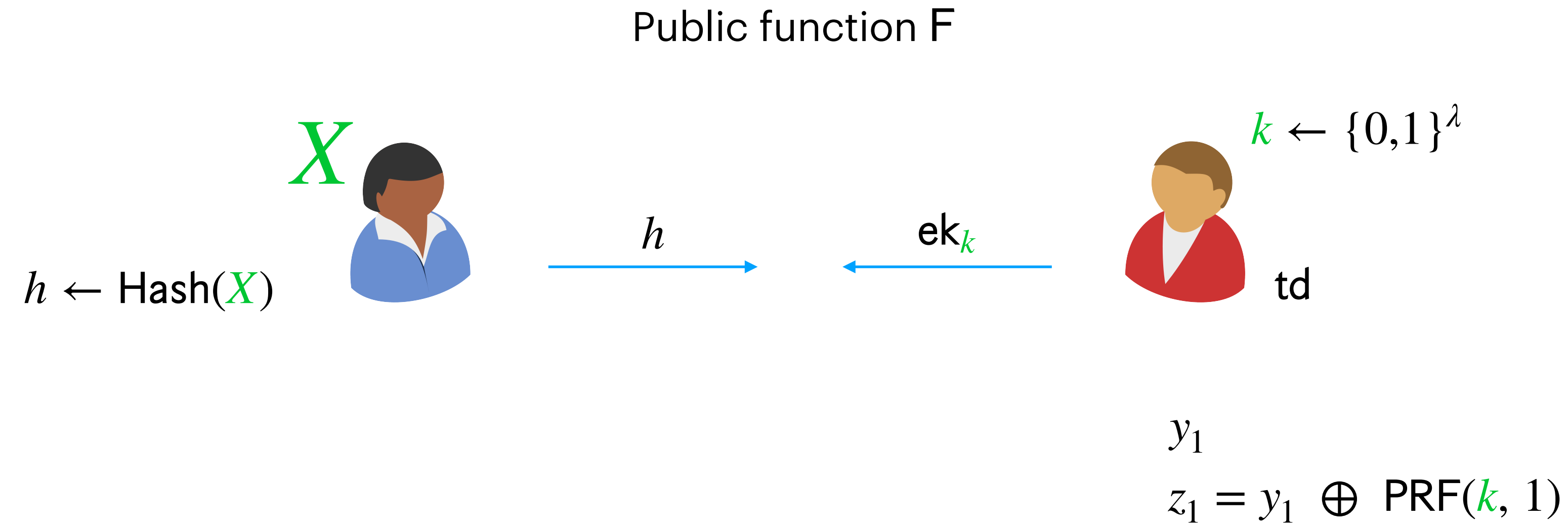
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



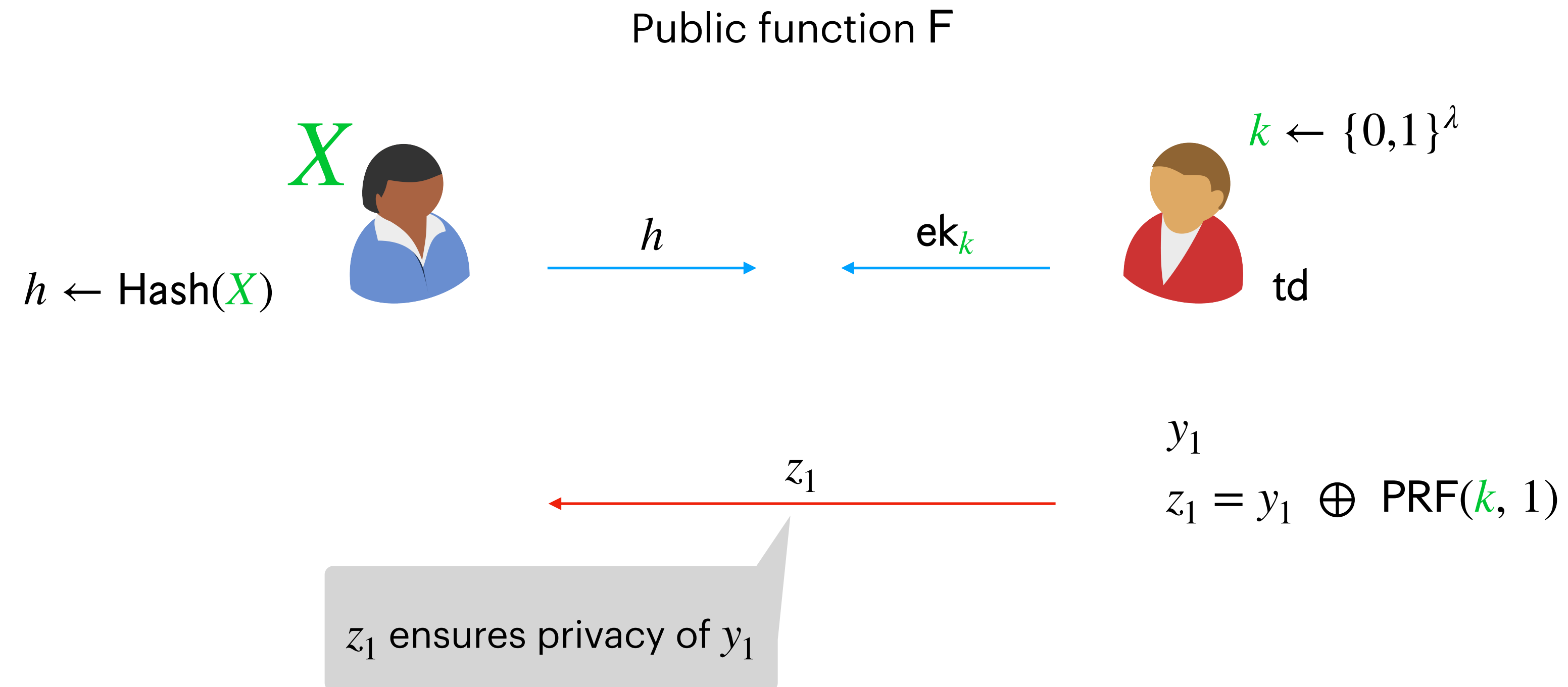
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



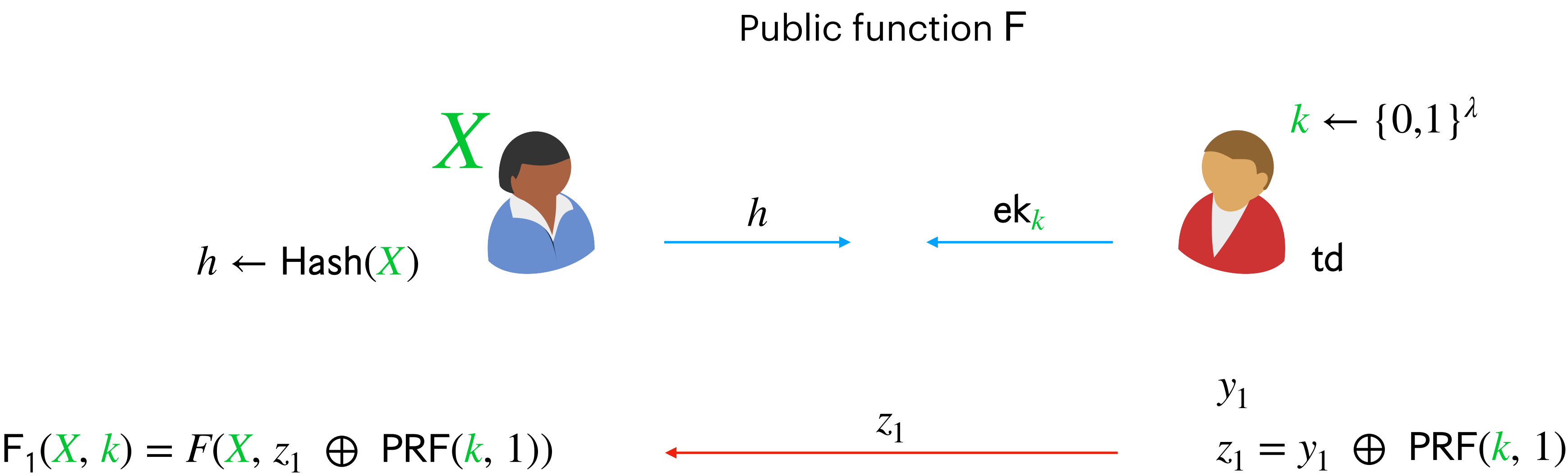
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



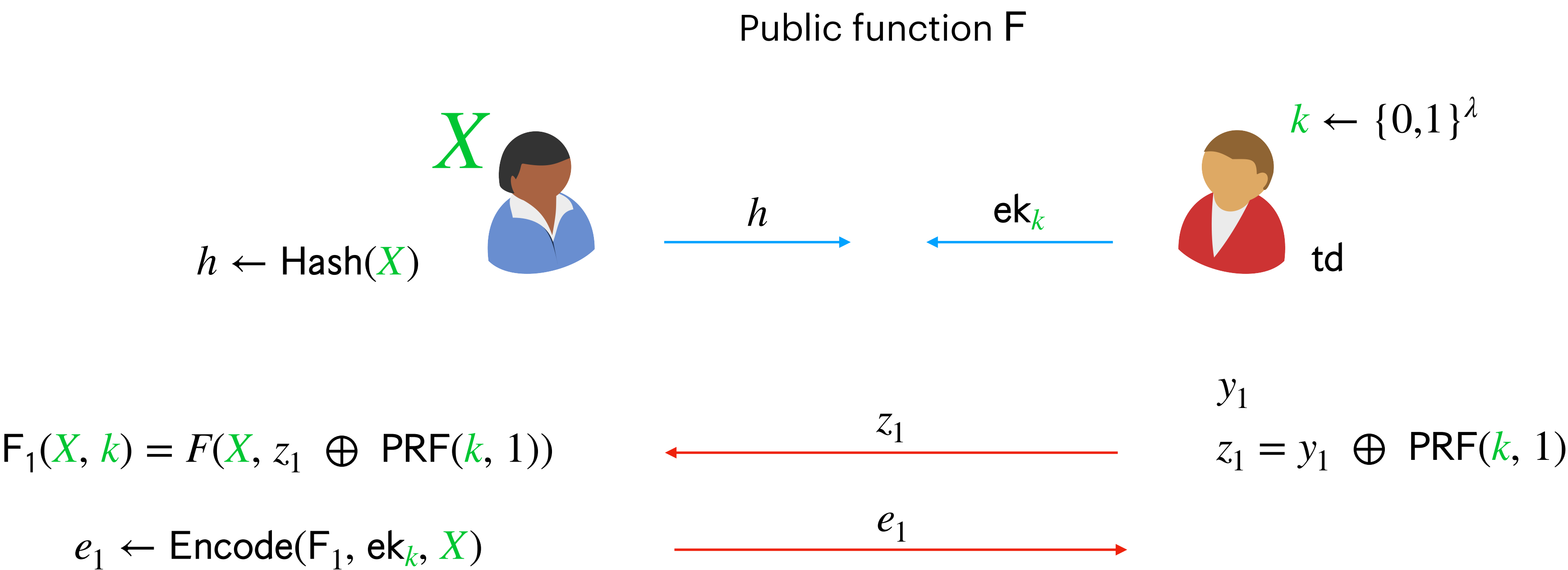
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



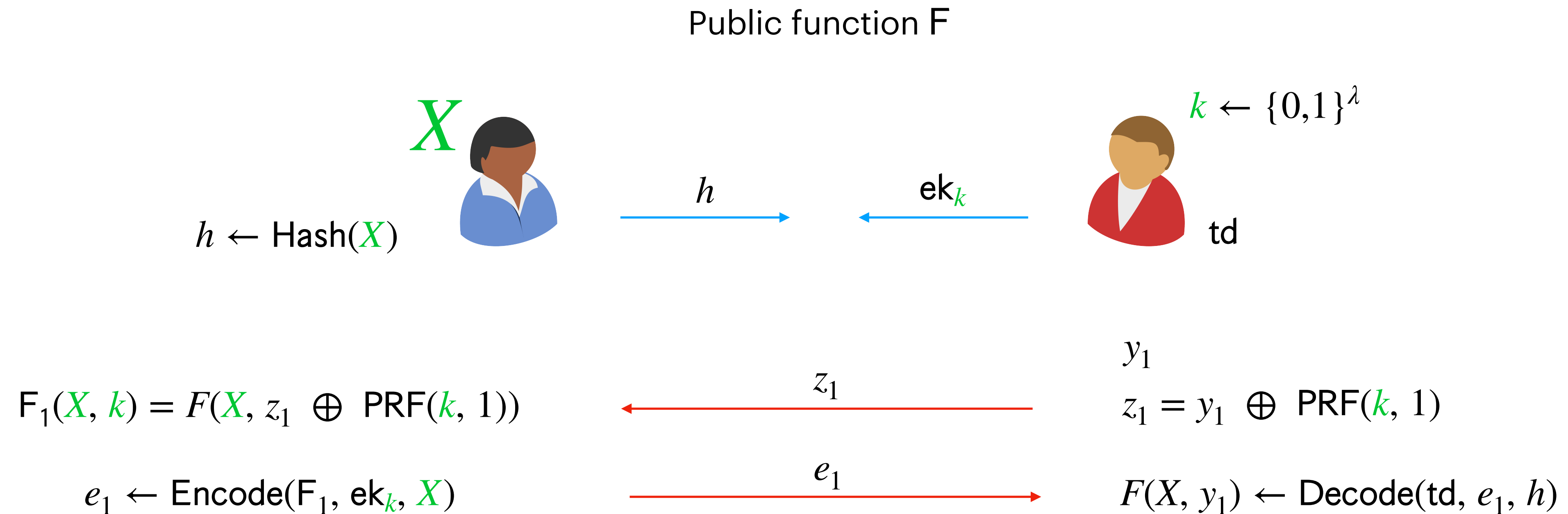
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



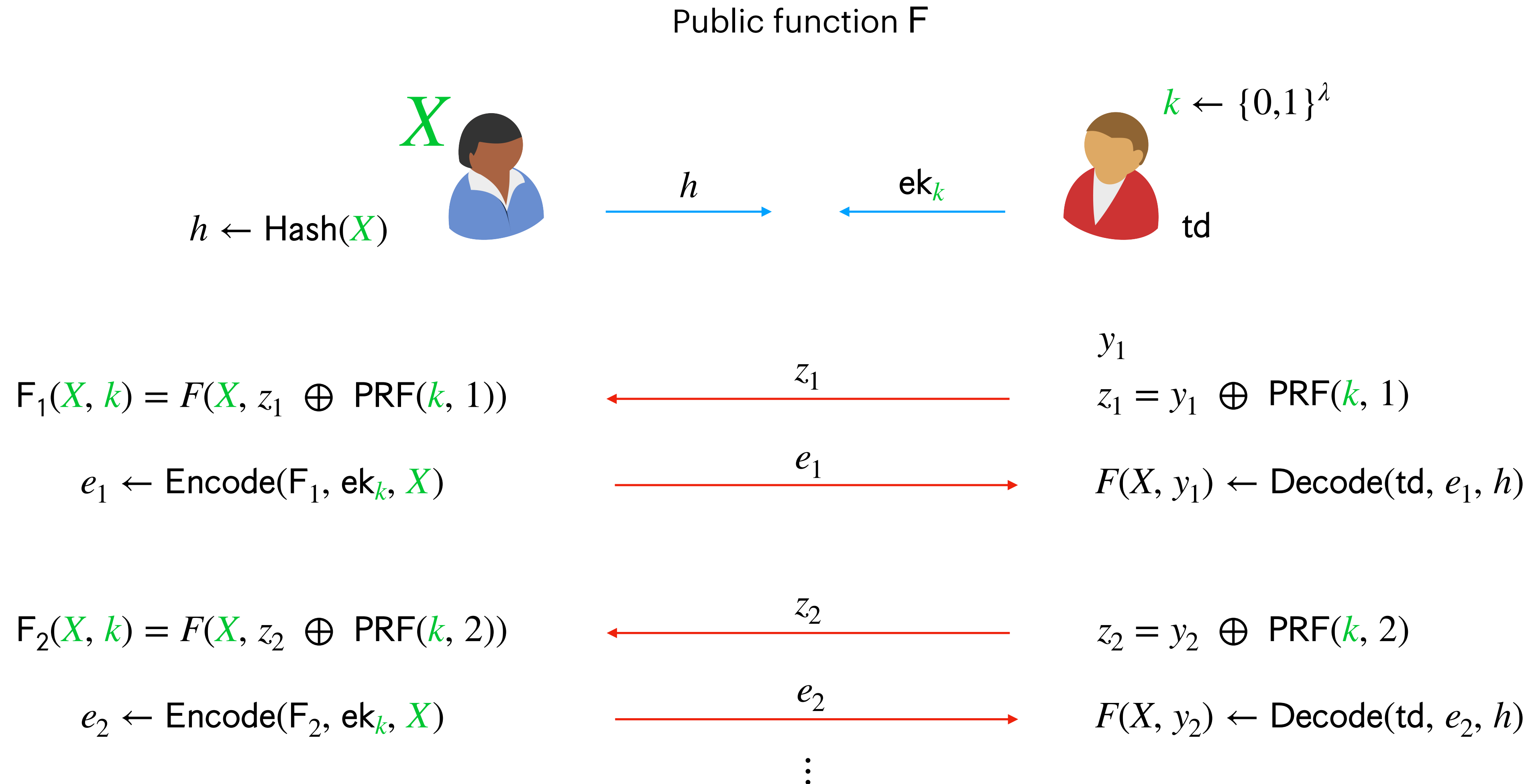
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



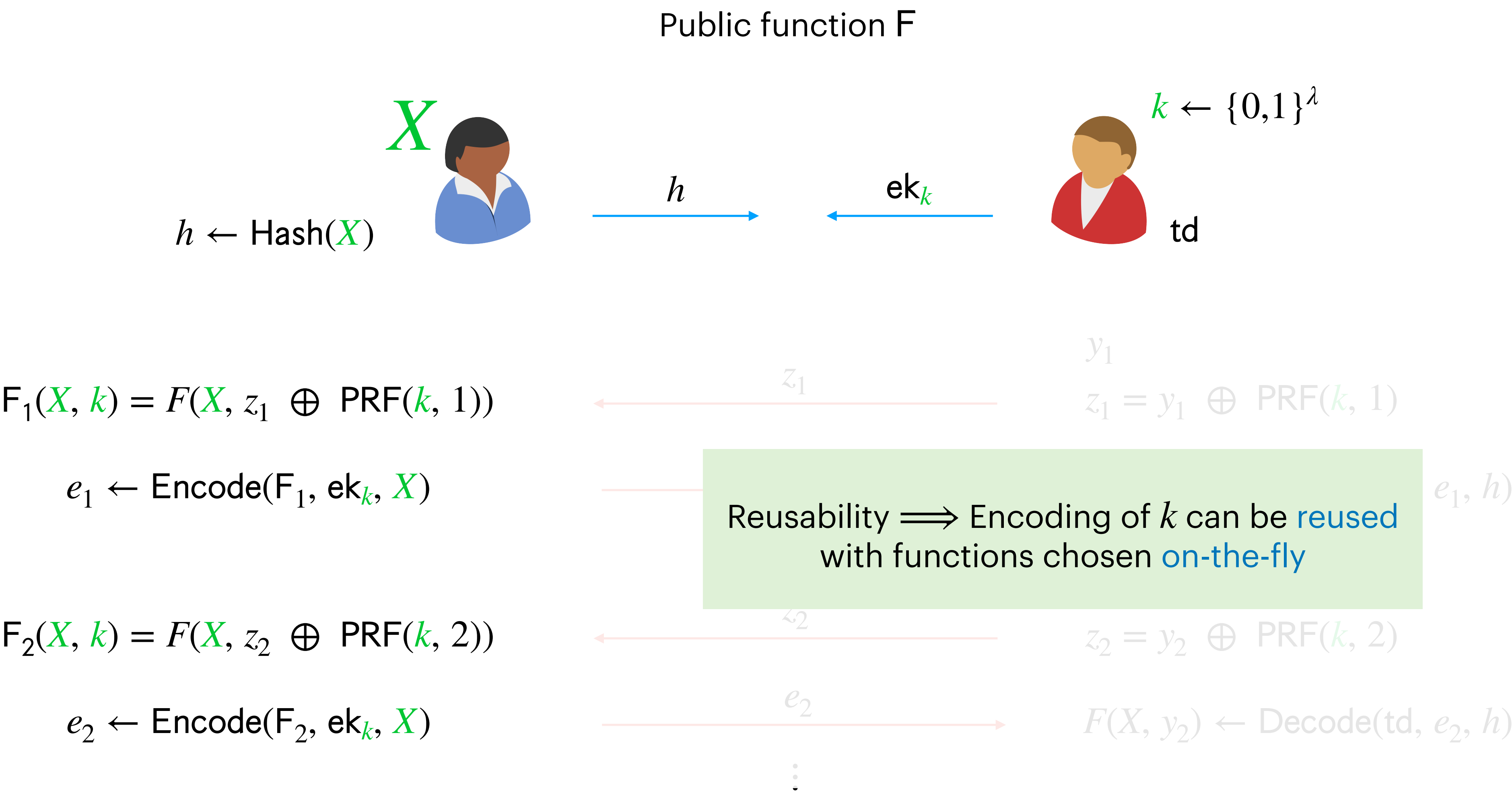
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



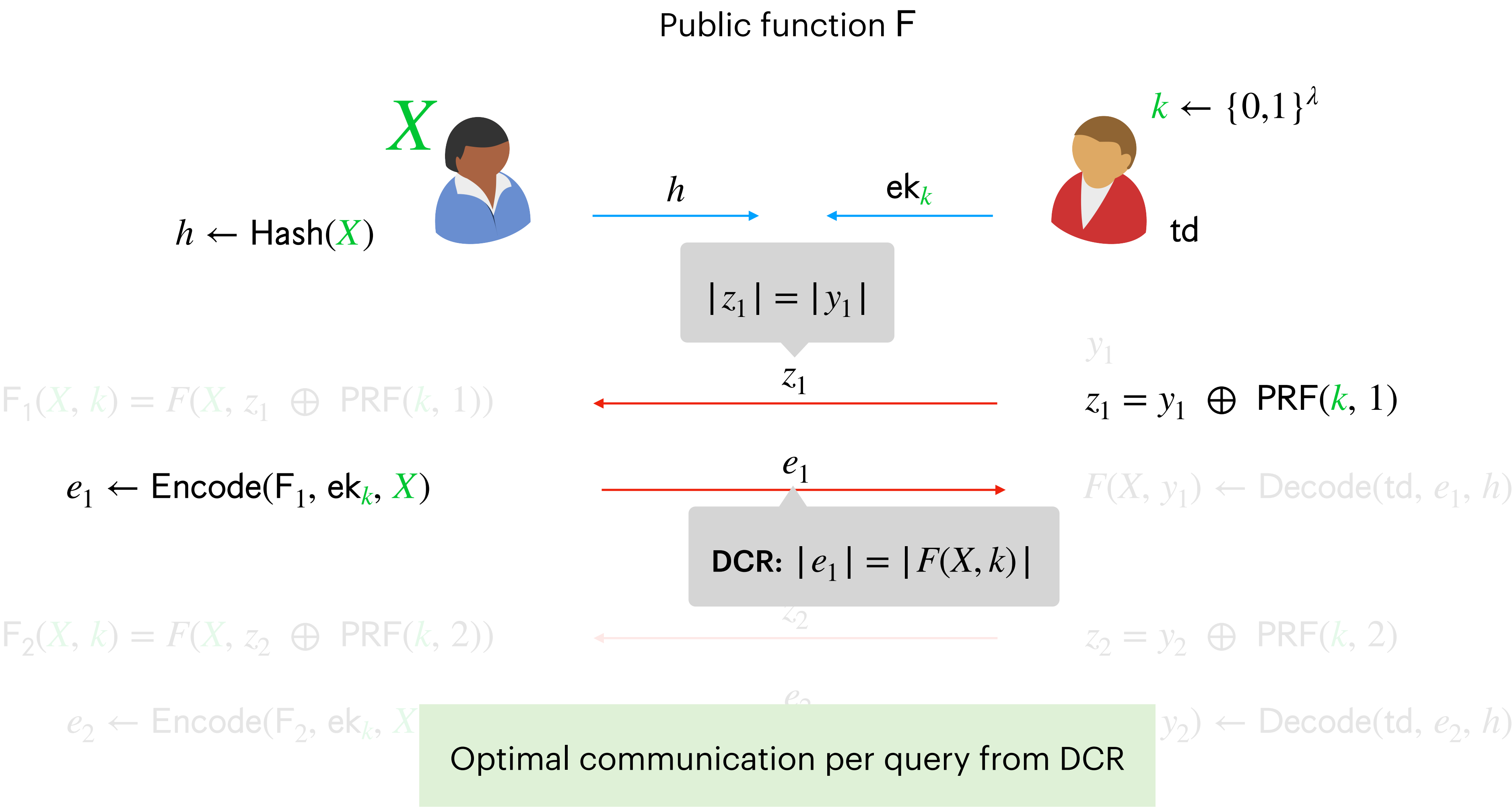
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



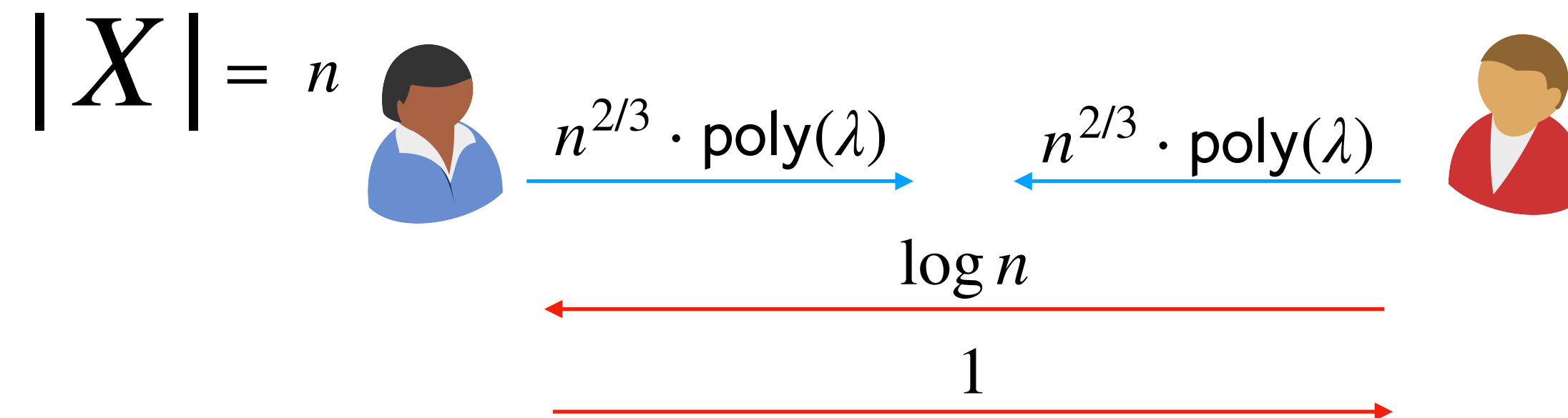
Applications 3: Compactness + Expressivity + Reusability

Improving Communication in the Amortized Setting



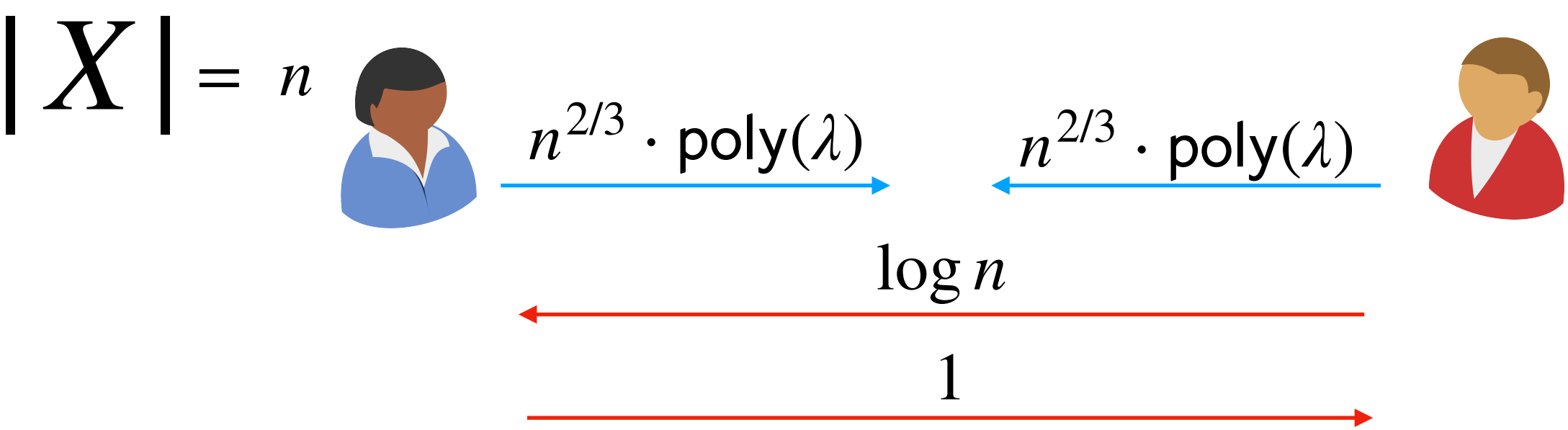
Applications 3: Compactness + Expressivity + Reusability

Optimal preprocessing symmetric Private Information Retrieval from DCR

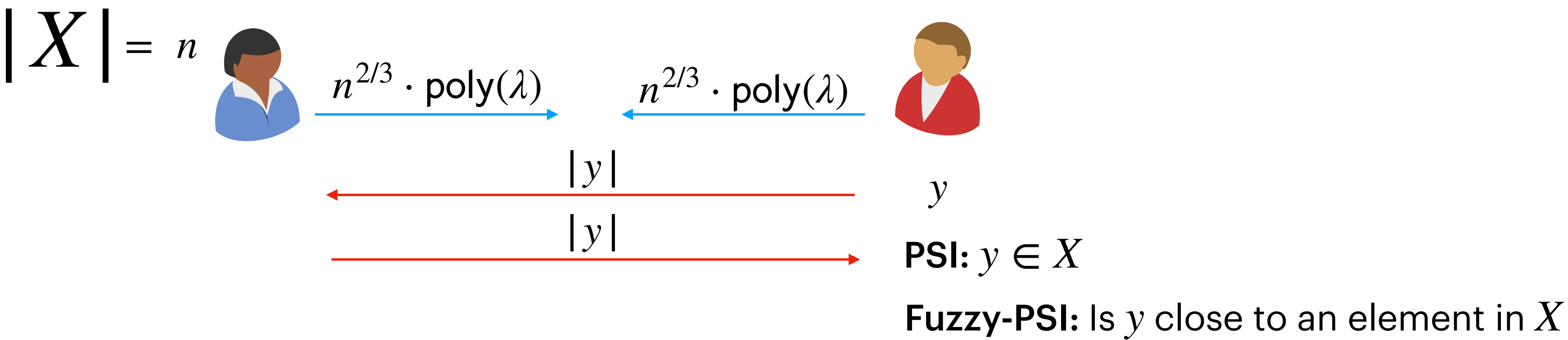


Applications 3: Compactness + Expressivity + Reusability

Optimal preprocessing symmetric Private Information Retrieval from DCR



Rate- $\frac{1}{2}$ Private Set Intersection (PSI) and Fuzzy-PSI from DCR



Constructing Enhanced TDH

Constructing Enhanced TDH

Staged Homomorphic Secret Sharing

[Couteau-Meyer-Passelégue-Riahinia'23]

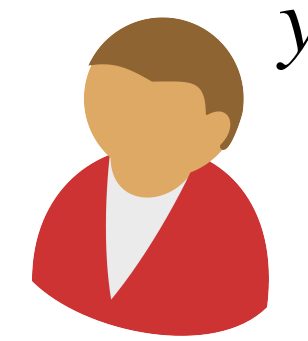
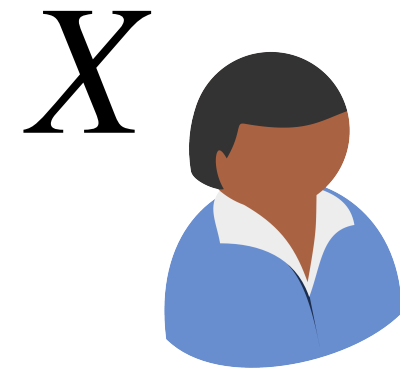
+

Trapdoor Hash Functions

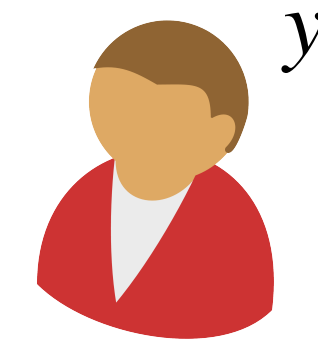
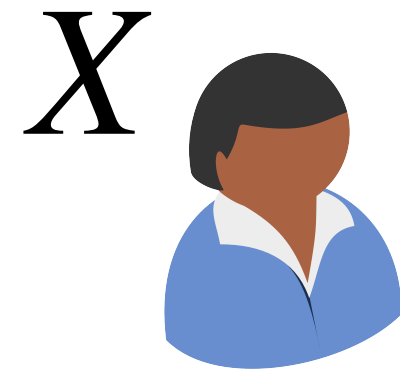
[Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

Alternative view: Extending Succinct HSS [Abram-Roy-Scholl'24] using Staged HSS

Staged Homomorphic Secret Sharing

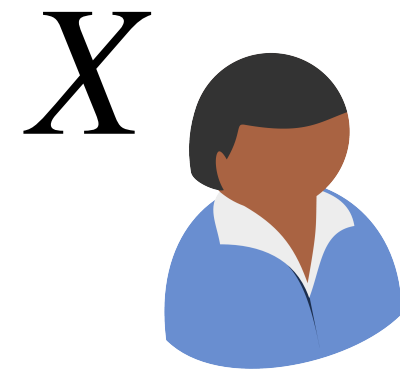


Staged Homomorphic Secret Sharing

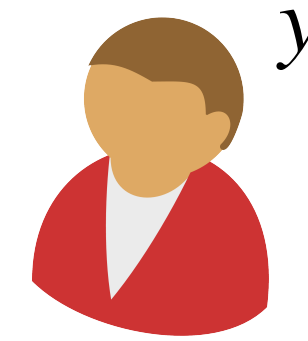


$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

Staged Homomorphic Secret Sharing



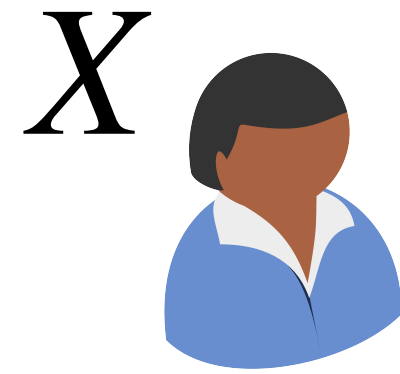
$ct_y \leftarrow \text{Encrypt}(pk, y)$



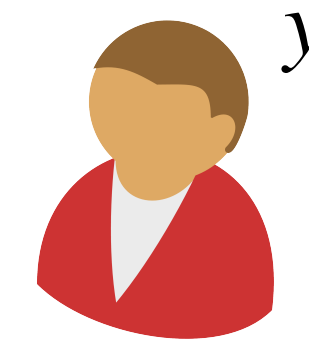
y

$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

Staged Homomorphic Secret Sharing

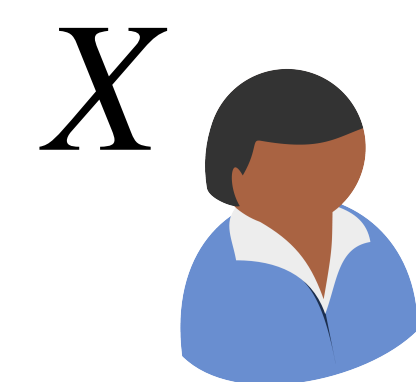


$ct_y \leftarrow \text{Encrypt}(pk, y)$
 $\text{share}_A, \text{share}_B \leftarrow \text{Share}(pk, X)$



y
 $(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

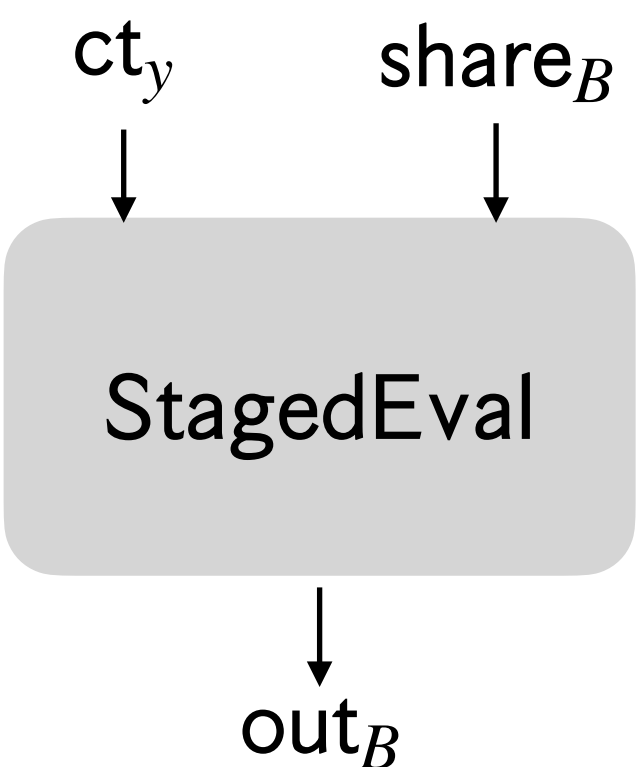
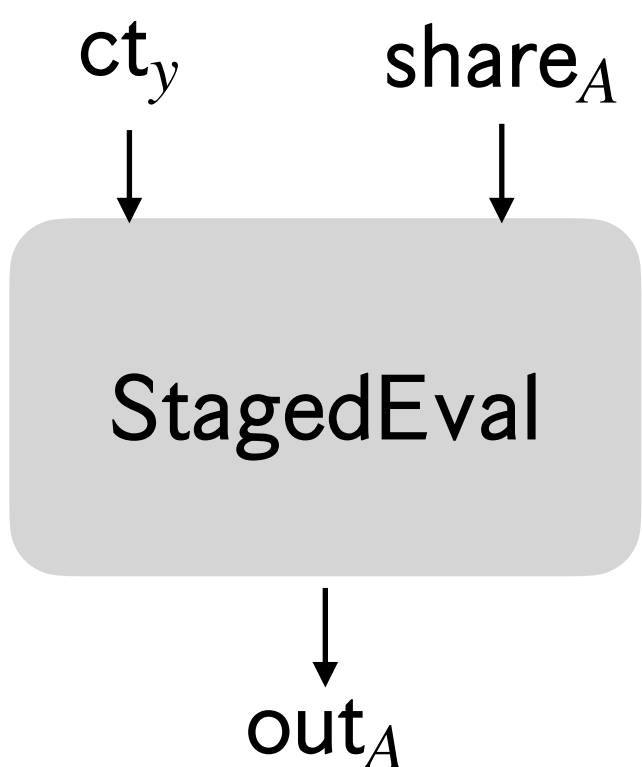
Staged Homomorphic Secret Sharing



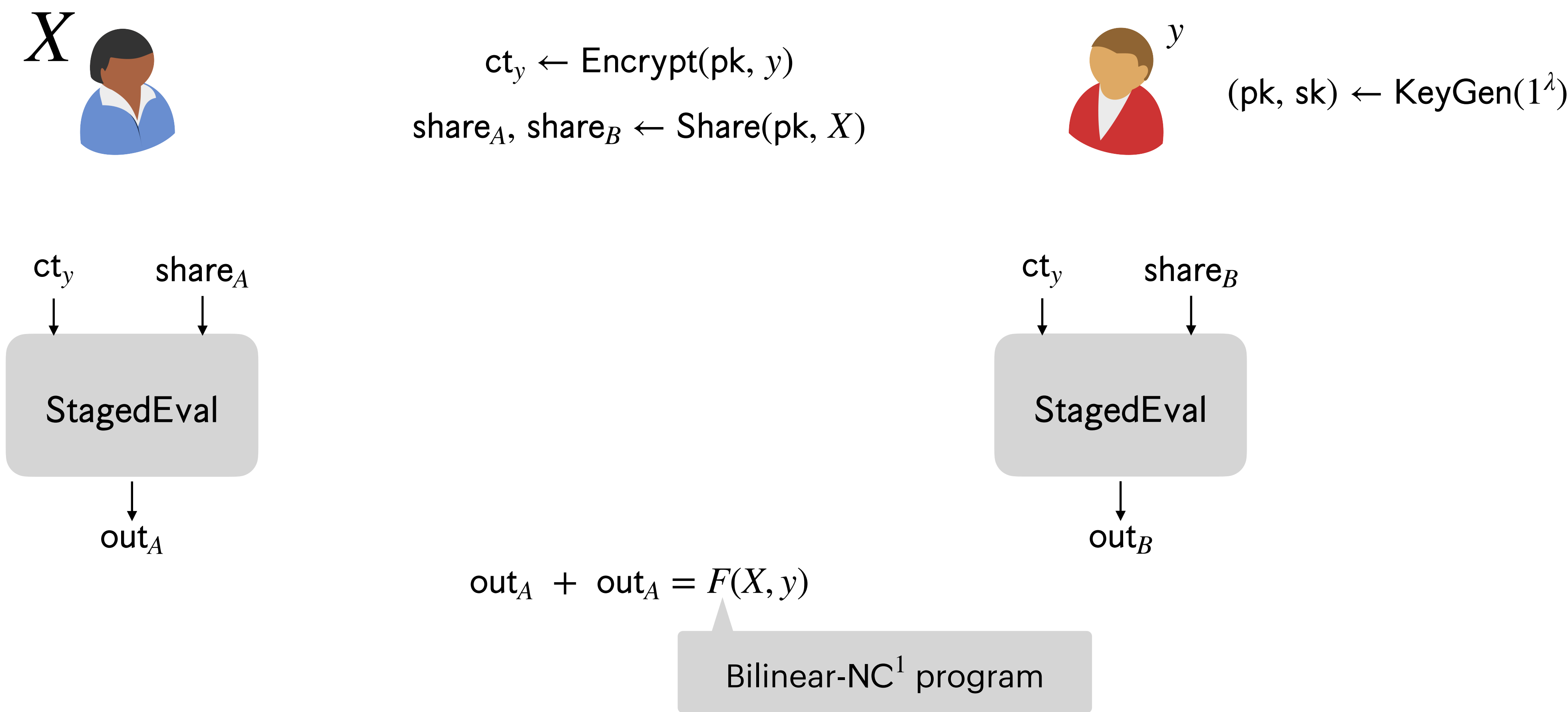
$$\begin{aligned} \text{ct}_y &\leftarrow \text{Encrypt}(\text{pk}, y) \\ \text{share}_A, \text{share}_B &\leftarrow \text{Share}(\text{pk}, X) \end{aligned}$$



$$(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda)$$



Staged Homomorphic Secret Sharing

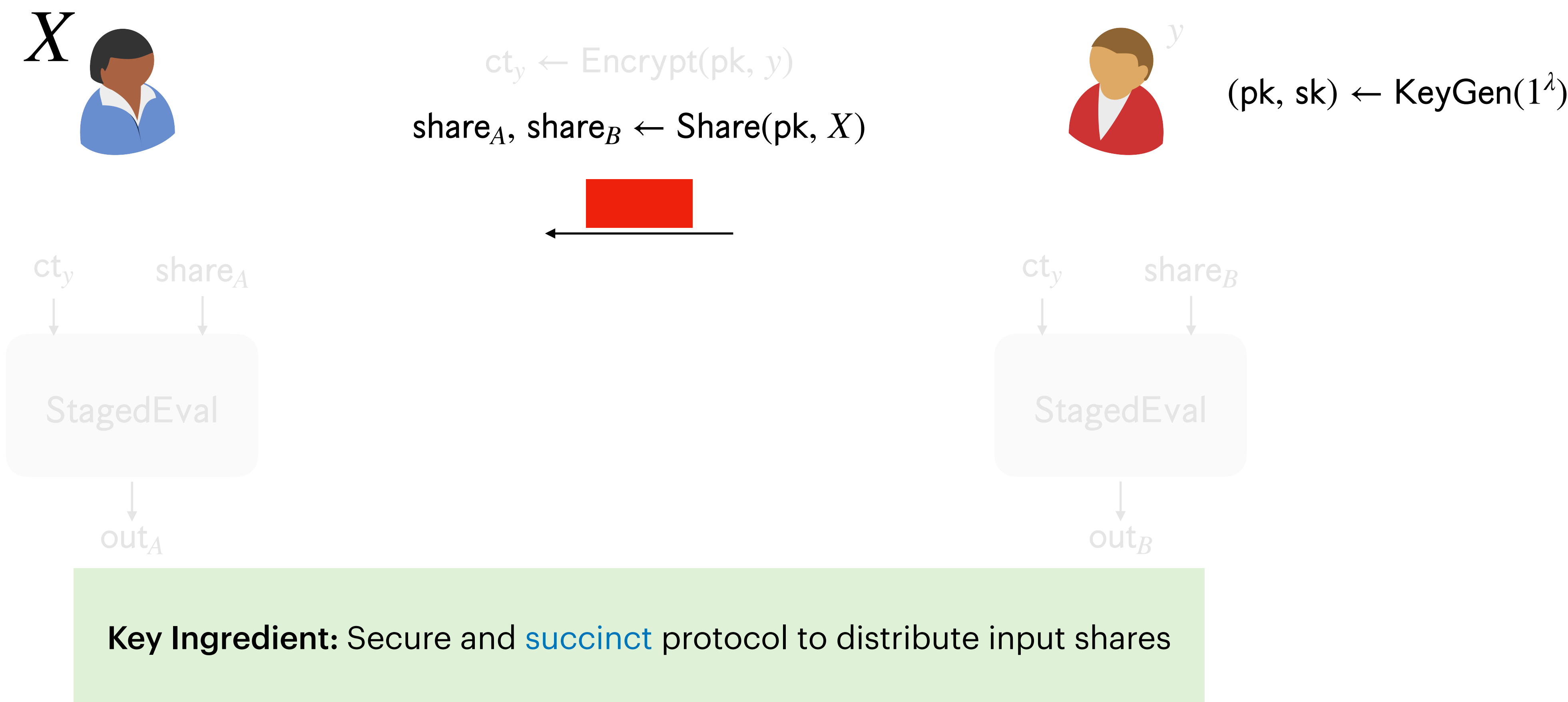


Staged Homomorphic Secret Sharing

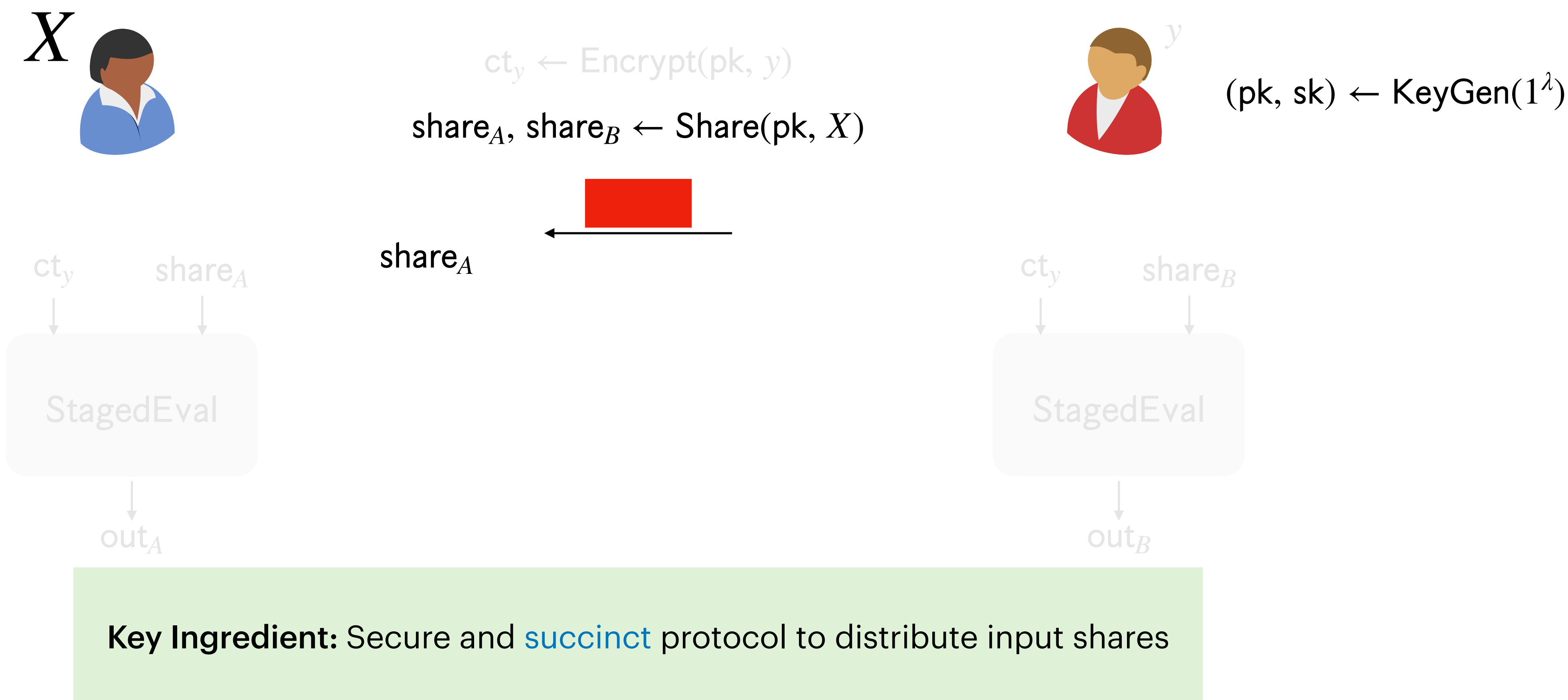


Key Ingredient: Secure and **succinct** protocol to distribute input shares

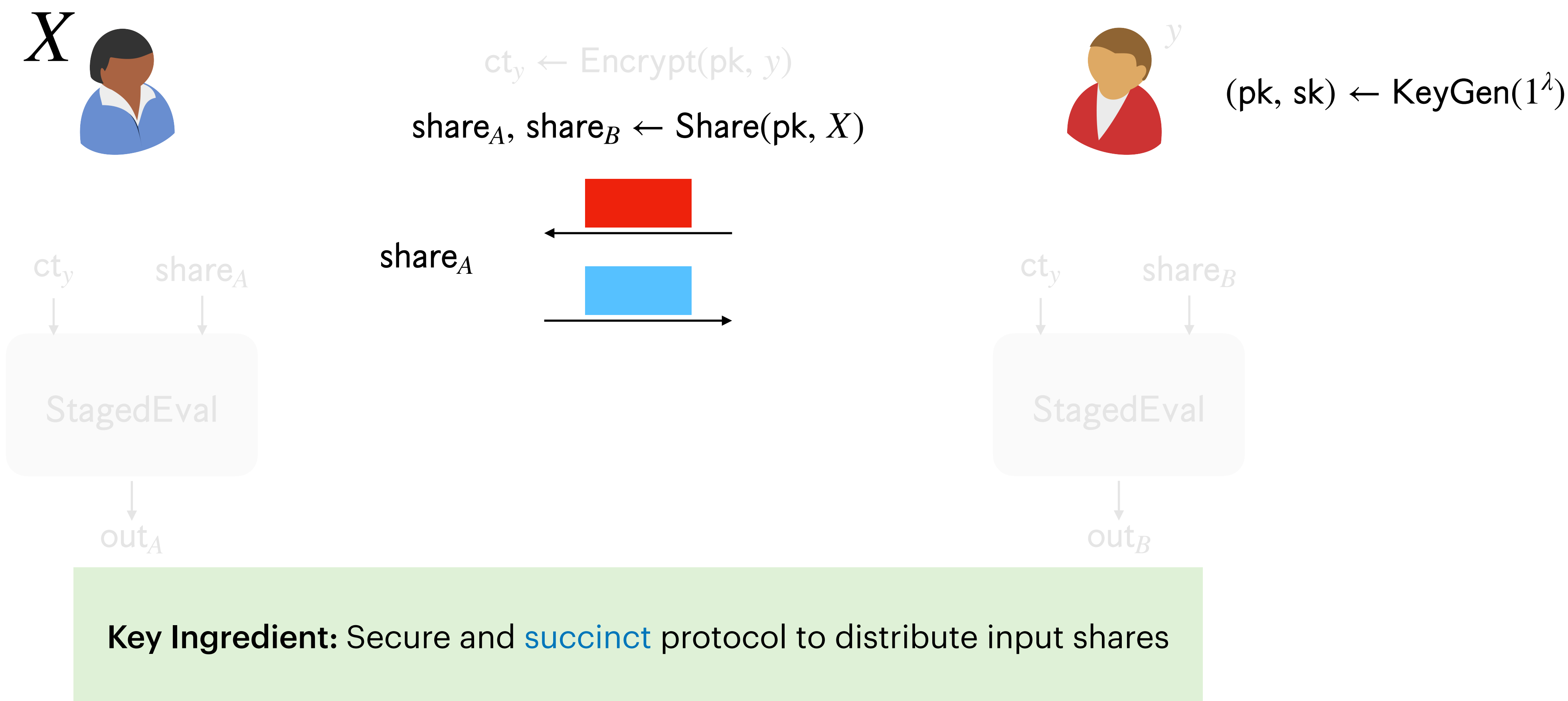
Staged Homomorphic Secret Sharing



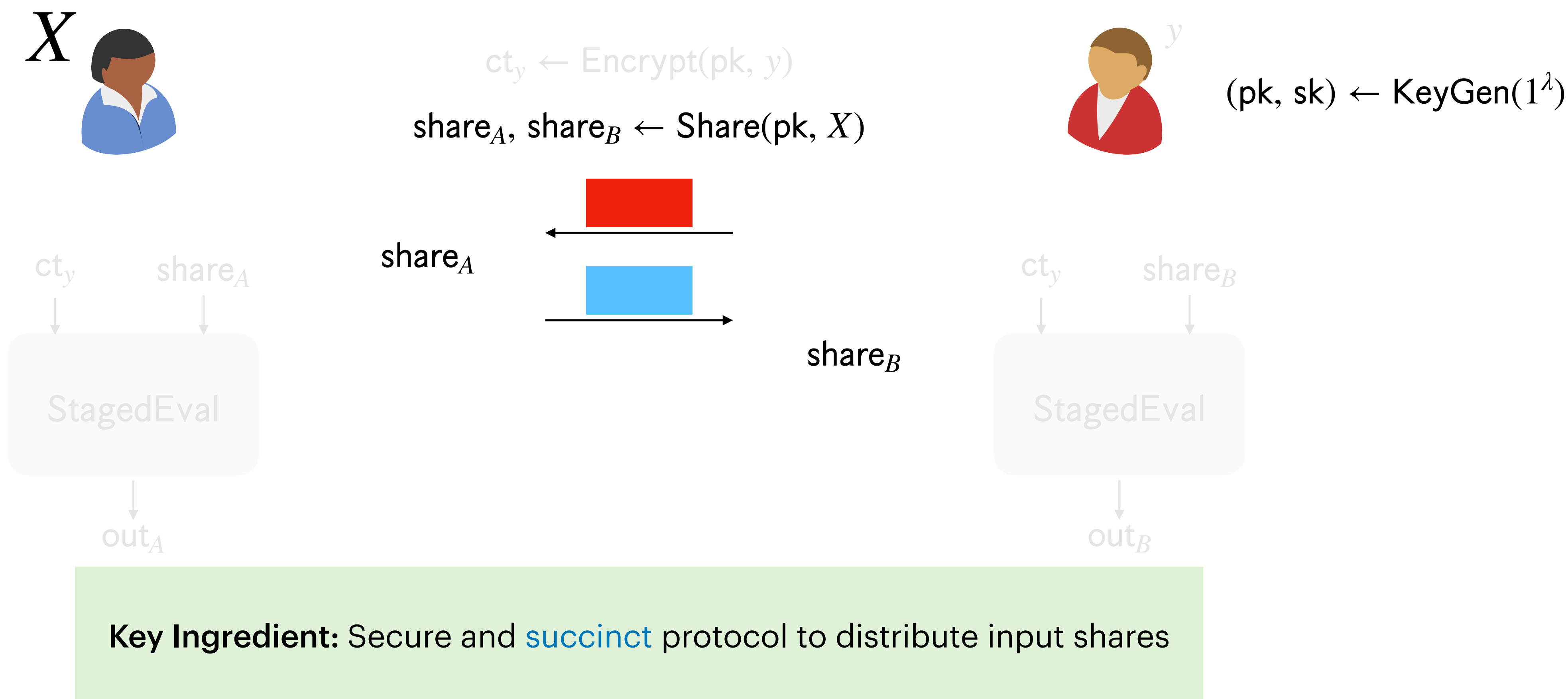
Staged Homomorphic Secret Sharing



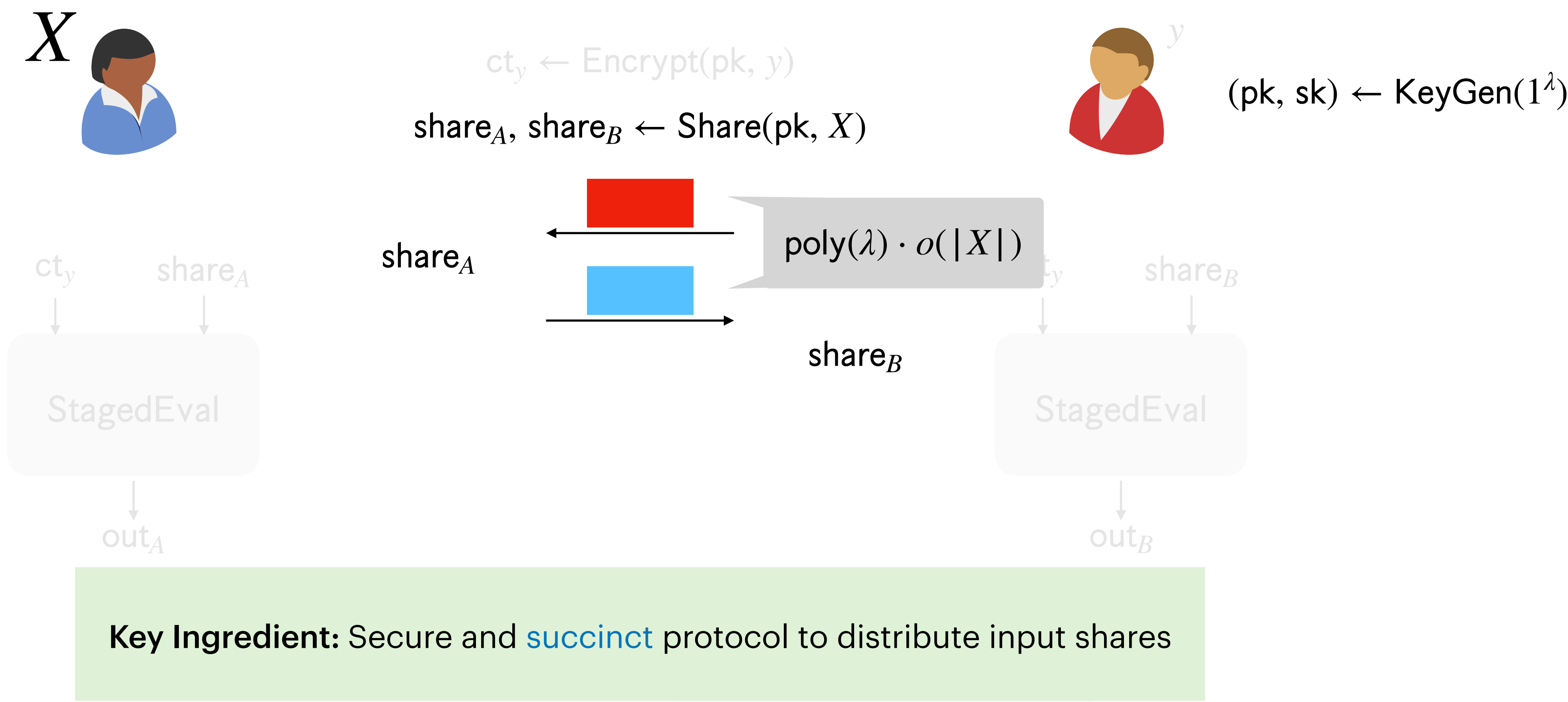
Staged Homomorphic Secret Sharing



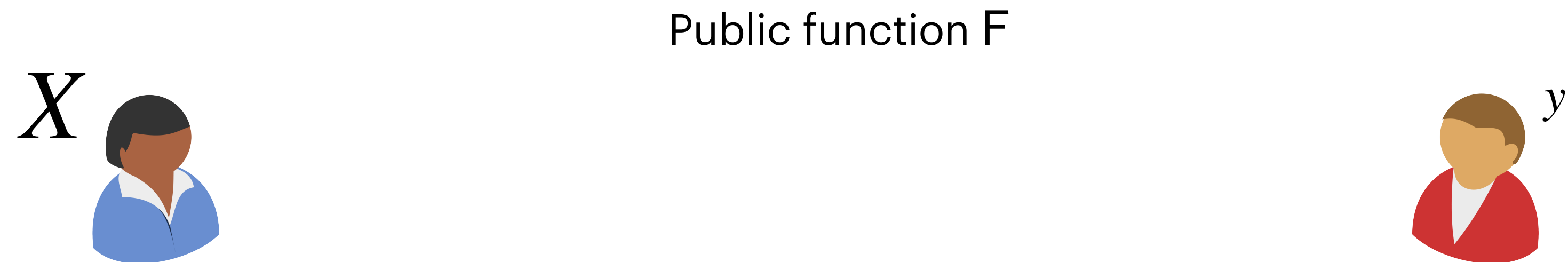
Staged Homomorphic Secret Sharing



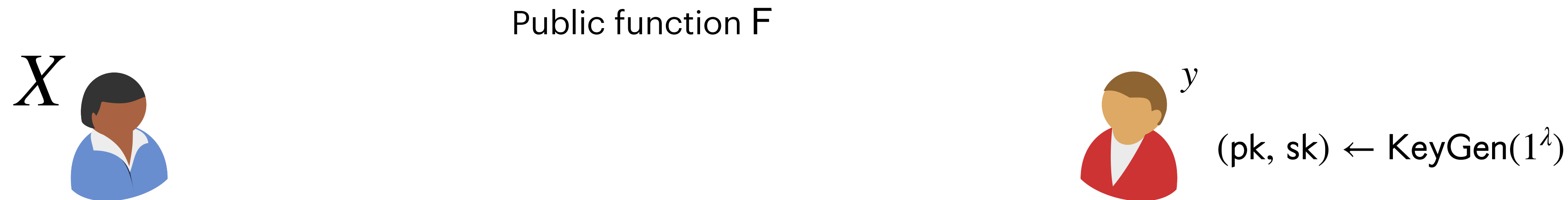
Staged Homomorphic Secret Sharing



Enhanced TDH = Staged HSS + Succinct Distribution of Shares

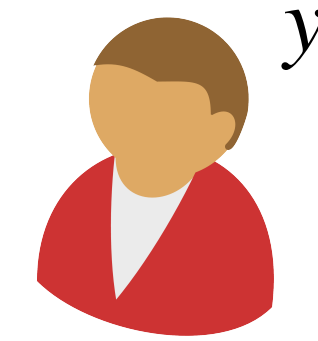
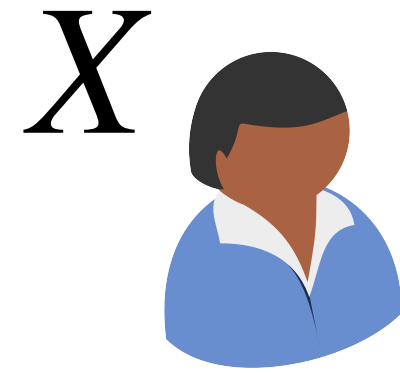


Enhanced TDH = Staged HSS + Succinct Distribution of Shares



Enhanced TDH = Staged HSS + Succinct Distribution of Shares

Public function F



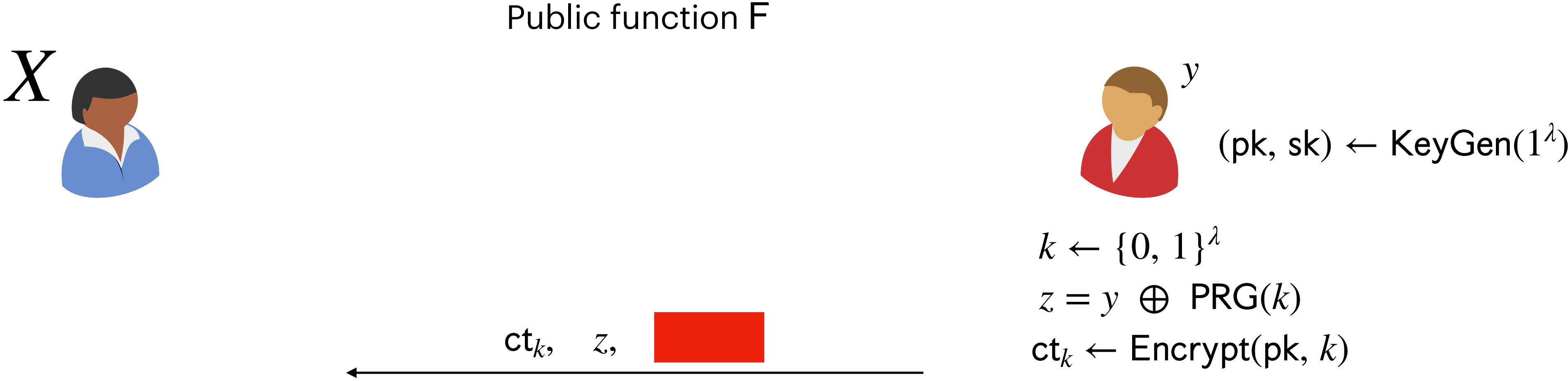
$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

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

$z = y \oplus \text{PRG}(k)$

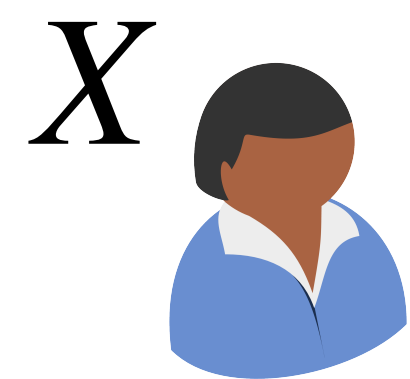
$ct_k \leftarrow \text{Encrypt}(pk, k)$

Enhanced TDH = Staged HSS + Succinct Distribution of Shares



Enhanced TDH = Staged HSS + Succinct Distribution of Shares

Public function F



Compute share_A from



ct_k, z,



y

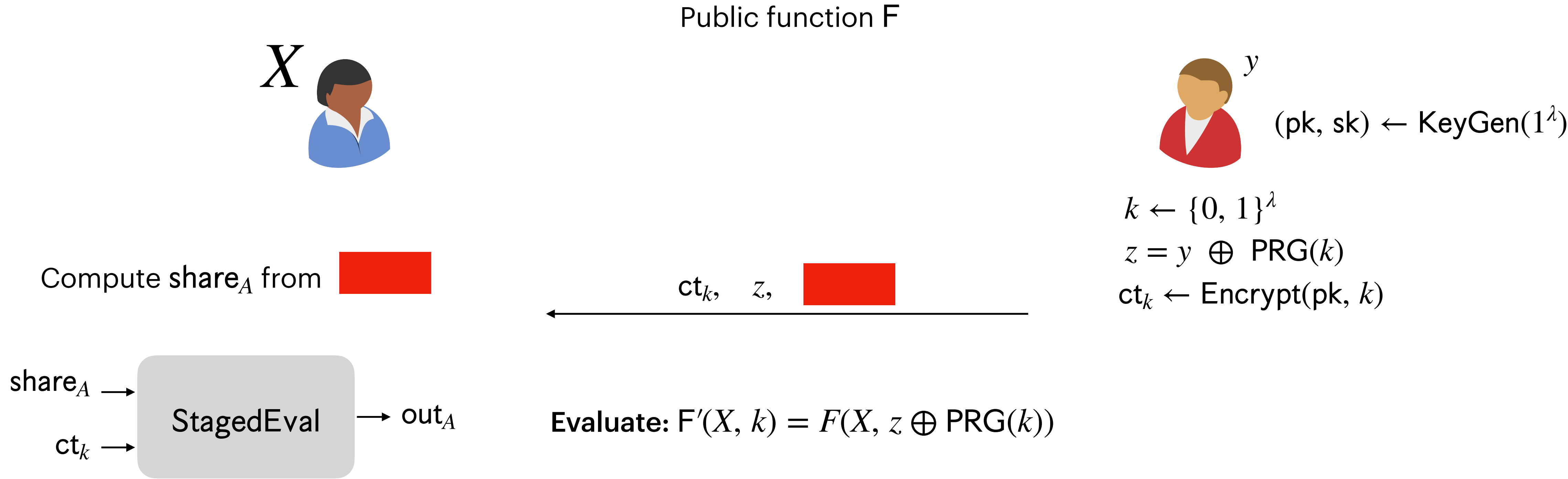
$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

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

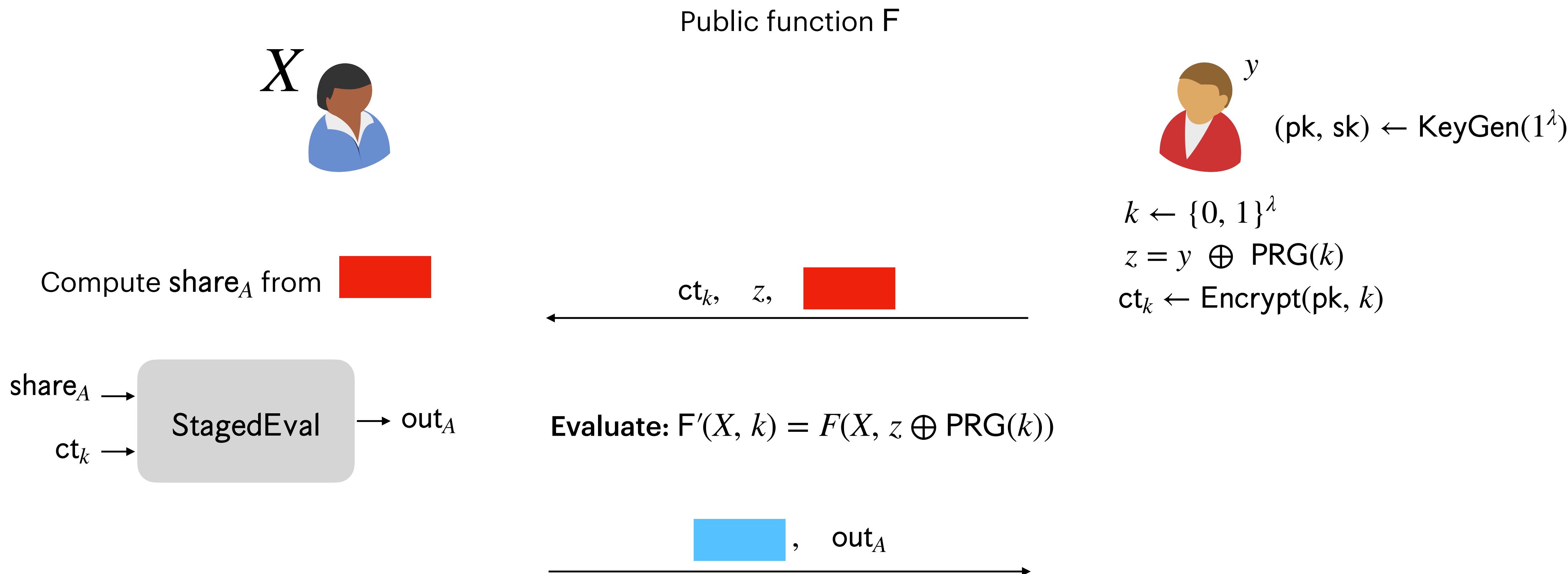
$z = y \oplus \text{PRG}(k)$

$ct_k \leftarrow \text{Encrypt}(pk, k)$

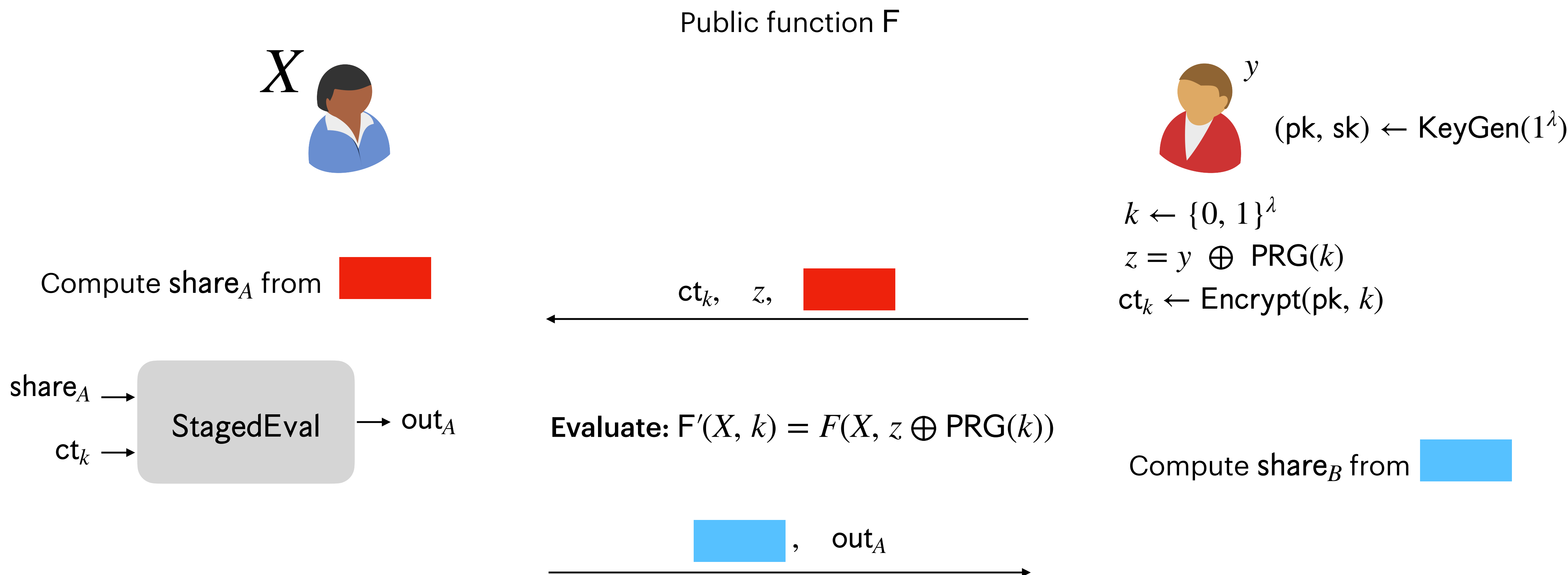
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



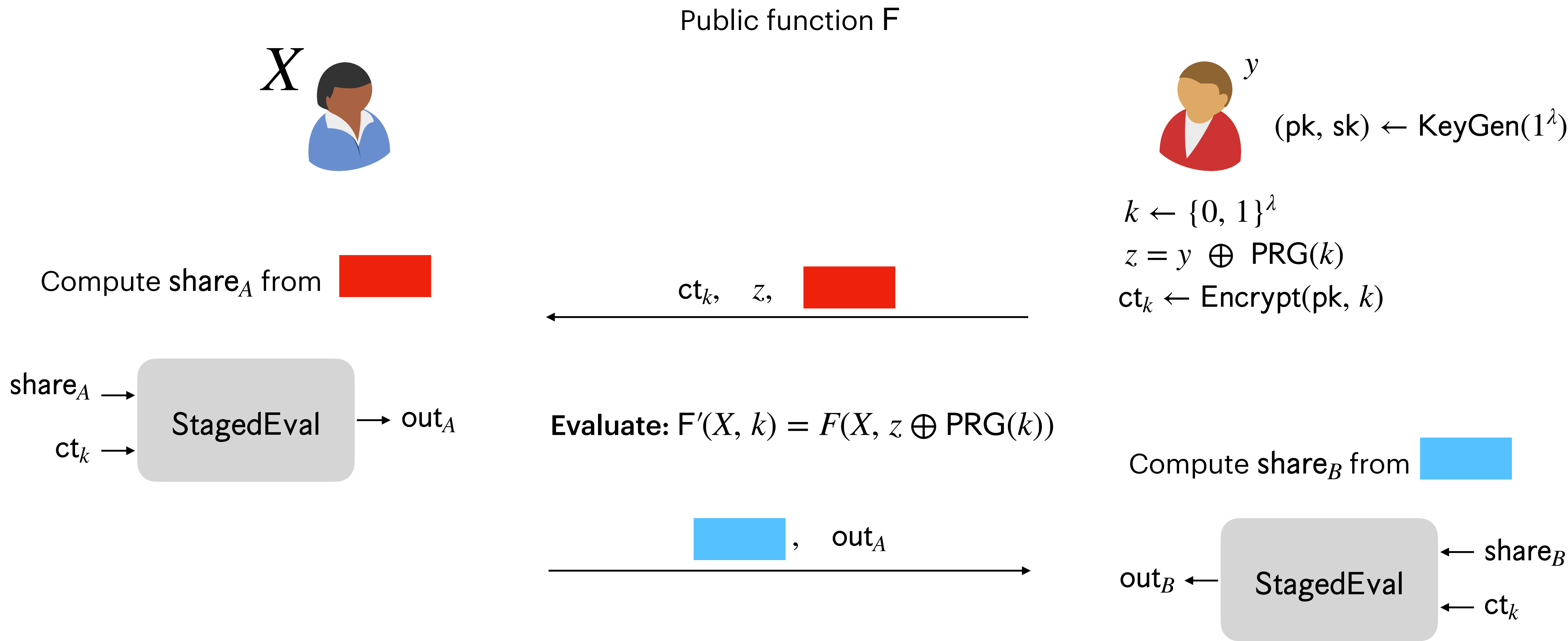
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



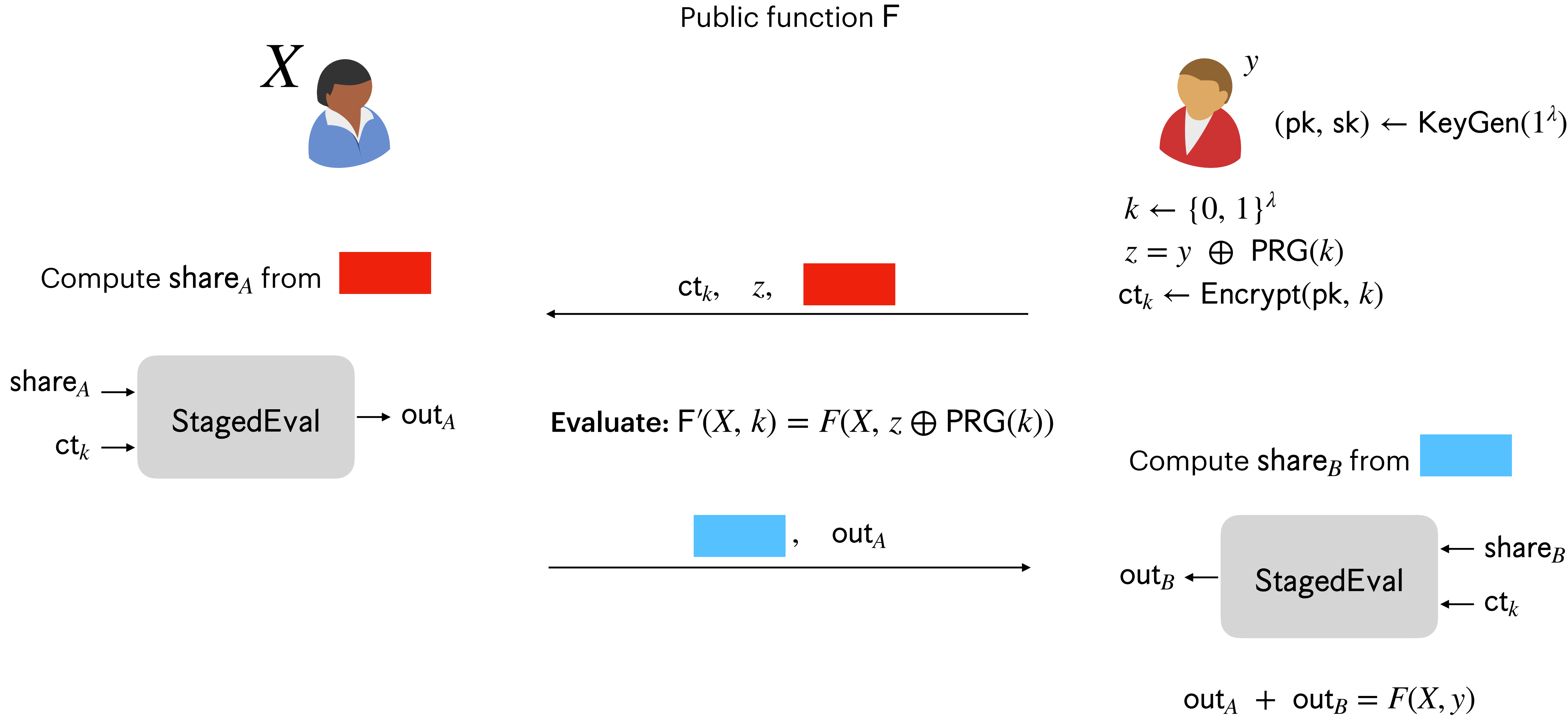
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



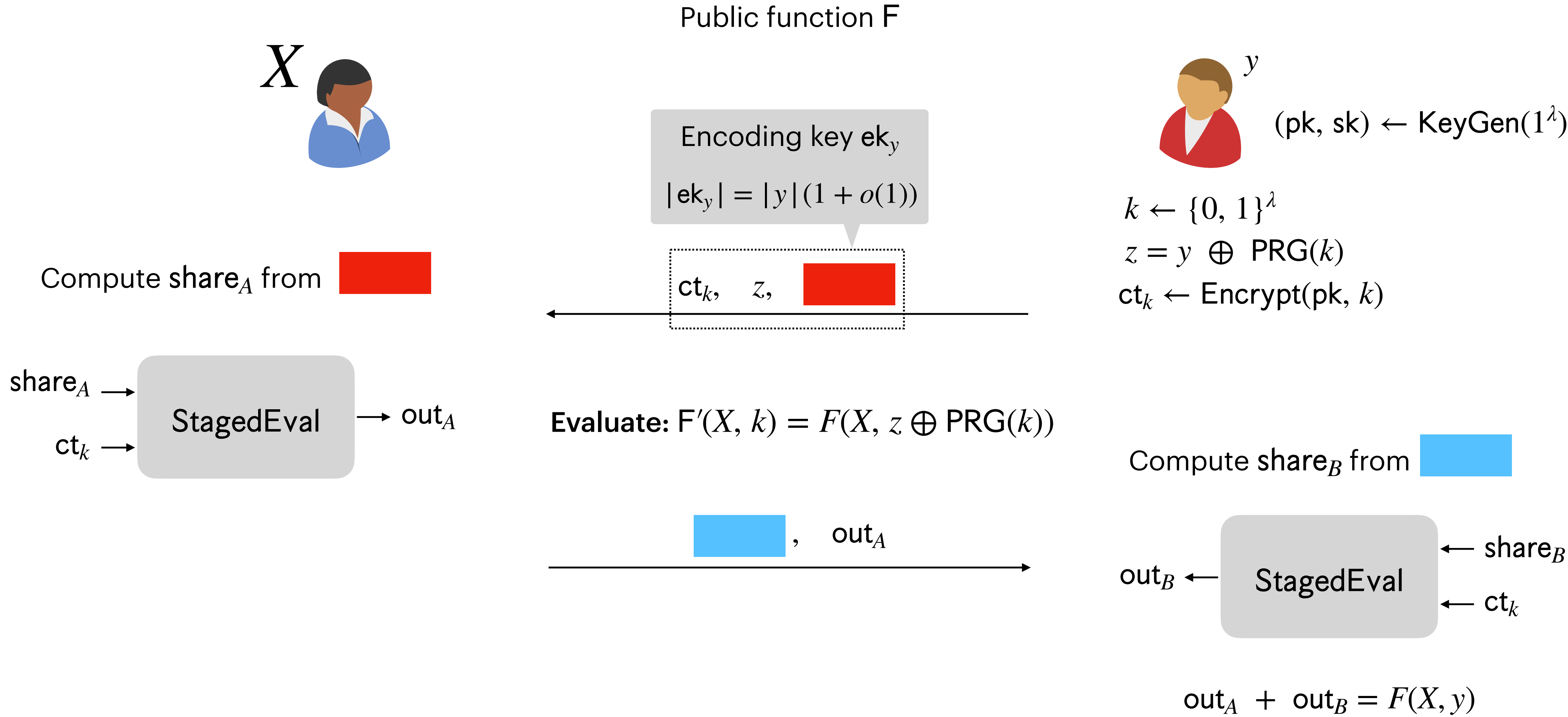
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



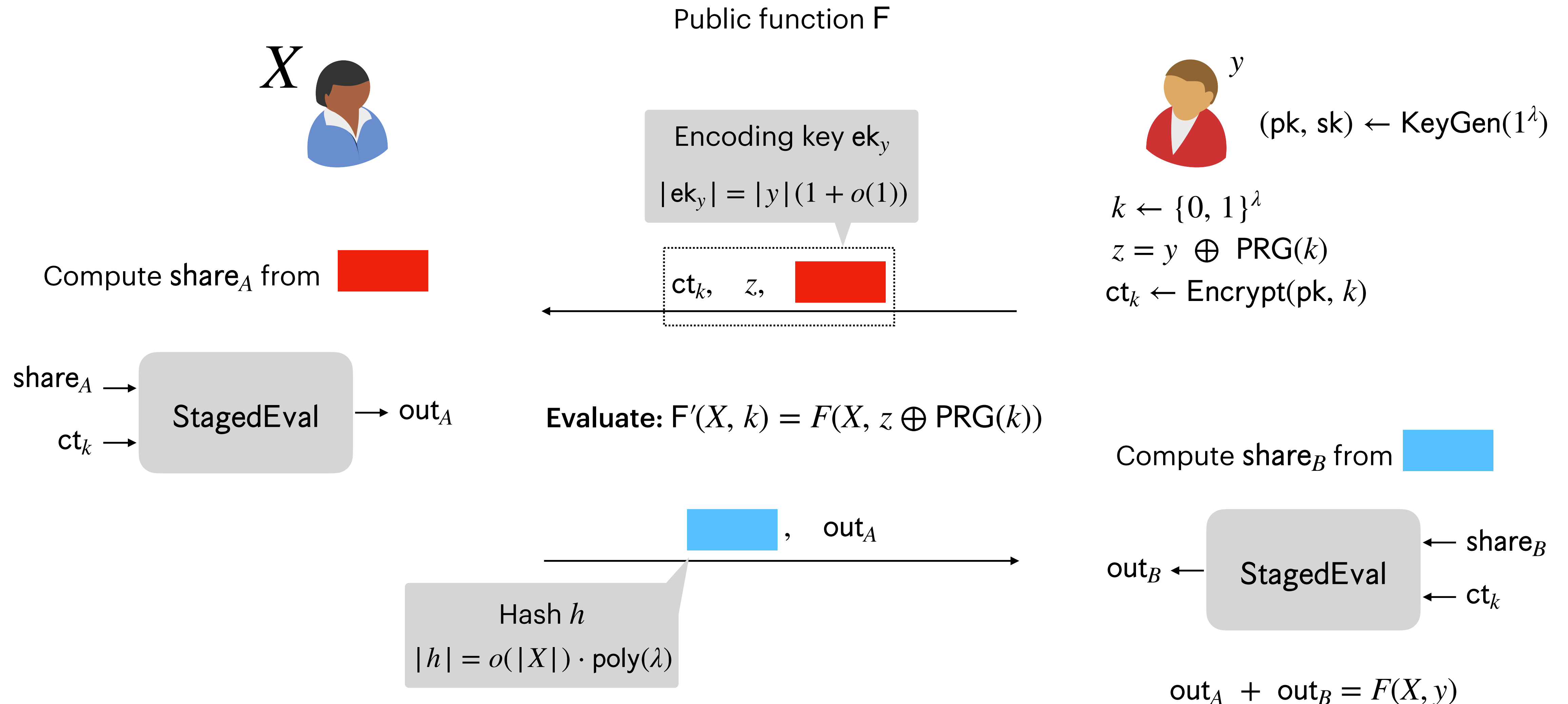
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



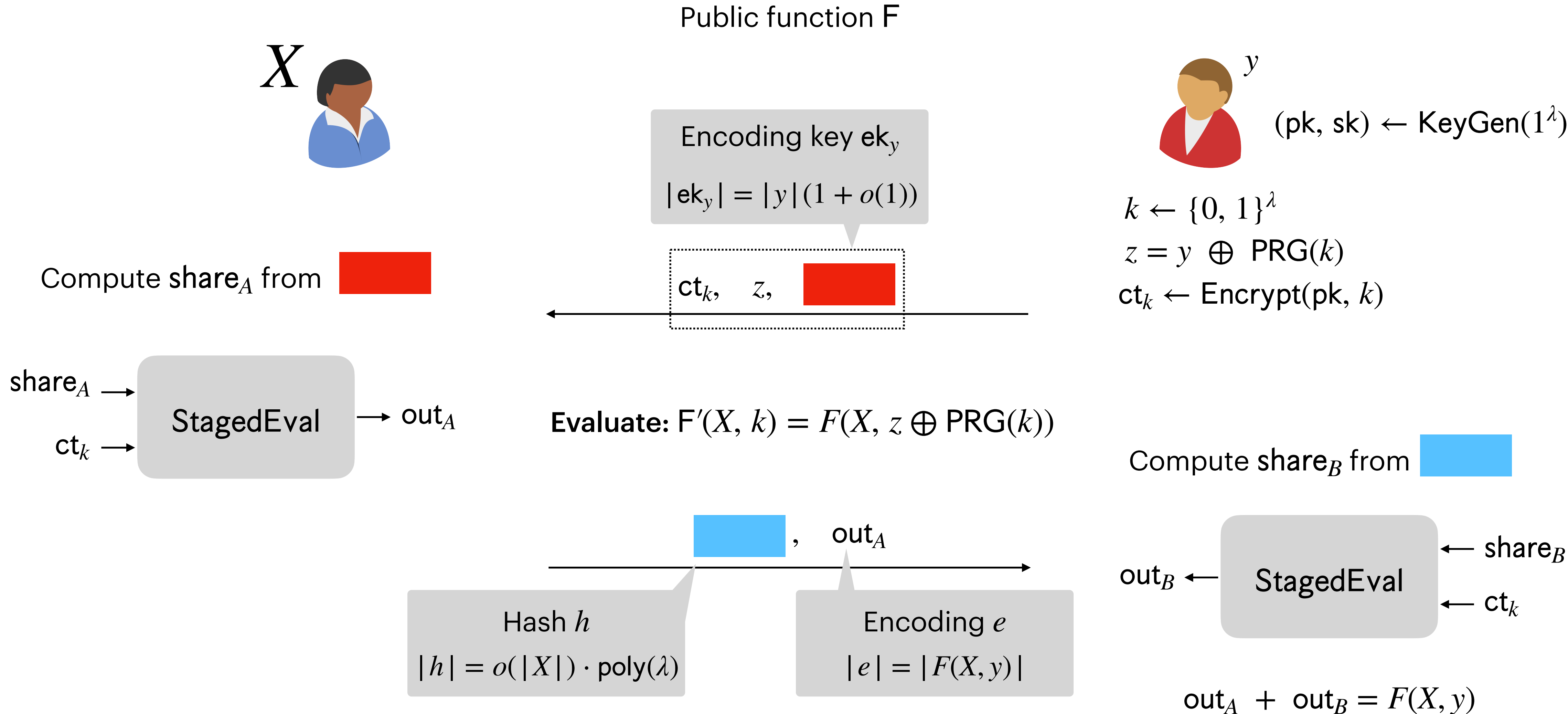
Enhanced TDH = Staged HSS + Succinct Distribution of Shares



Enhanced TDH = Staged HSS + Succinct Distribution of Shares



Enhanced TDH = Staged HSS + Succinct Distribution of Shares



Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$$X = (x_1, \dots, x_n)$$



Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$$X = (x_1, \dots, x_n)$$



$$\mathbb{G}, g$$

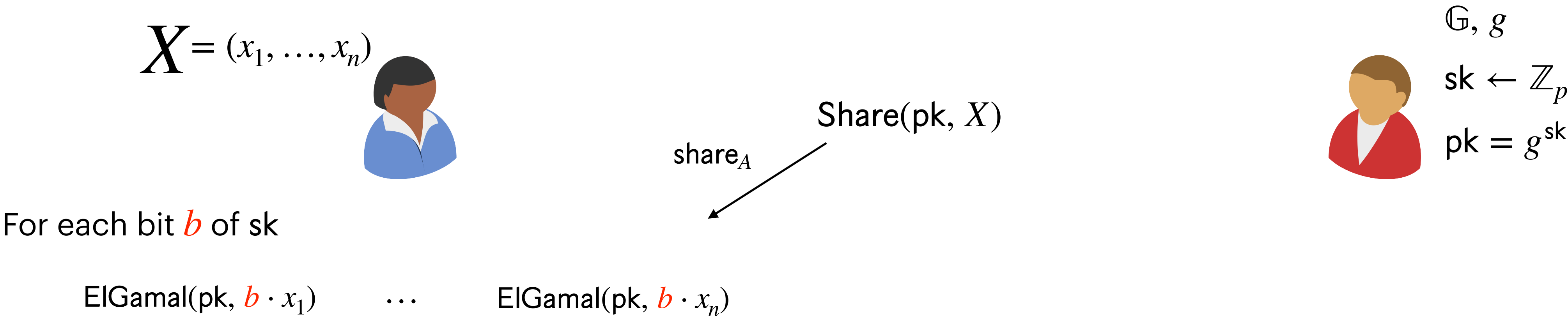
$$\text{sk} \leftarrow \mathbb{Z}_p$$

$$\text{pk} = g^{\text{sk}}$$



Succinct Distribution of Staged Input Shares

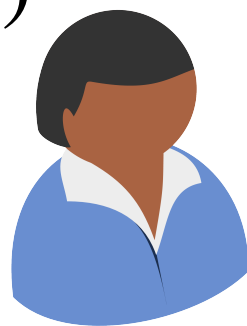
Structure of Staged Input Shares



Succinct Distribution of Staged Input Shares

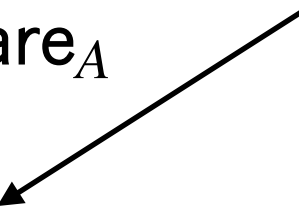
Structure of Staged Input Shares

$$X = (x_1, \dots, x_n)$$



Share(pk, X)

share_A



$$\mathbb{G}, g$$

$$sk \leftarrow \mathbb{Z}_p$$

$$pk = g^{sk}$$



For each bit *b* of sk

$$\text{ElGamal}(pk, b \cdot x_1) \quad \dots \quad \text{ElGamal}(pk, b \cdot x_n)$$

=

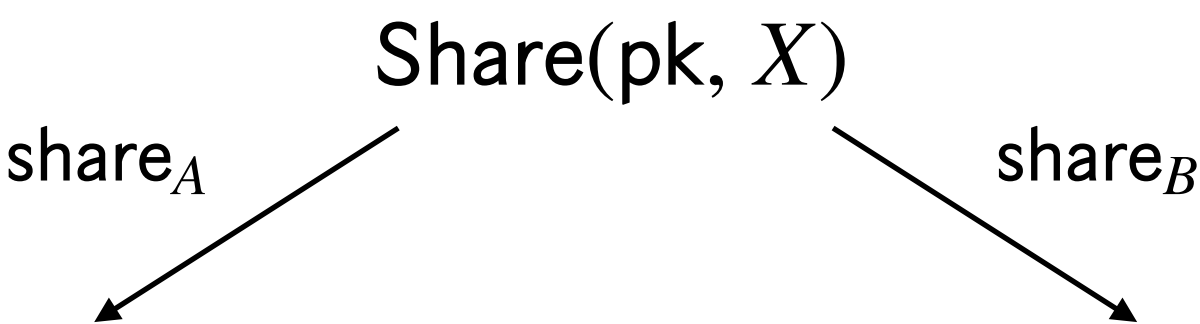
$$(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_1}) \quad \dots \quad (g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_n})$$

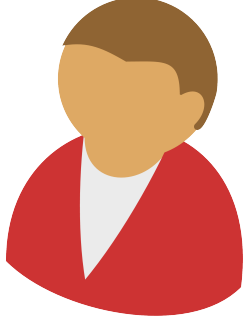
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$X = (x_1, \dots, x_n)$







$\mathbb{G}, g$
 $sk \leftarrow \mathbb{Z}_p$
 $pk = g^{sk}$

For each bit b of sk

ElGamal(pk, $b \cdot x_1$) ... ElGamal(pk, $b \cdot x_n$)

=

$(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_1})$... $(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_n})$

$g^{r_1} \dots g^{r_n}$

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

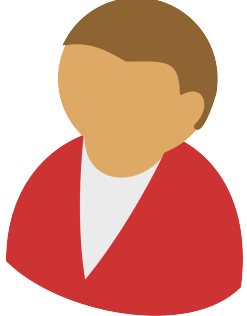
$X = (x_1, \dots, x_n)$



Share(pk, X)

share_A

share_B



$\mathbb{G}, g$
 $sk \leftarrow \mathbb{Z}_p$
 $pk = g^{sk}$

For each bit b of sk

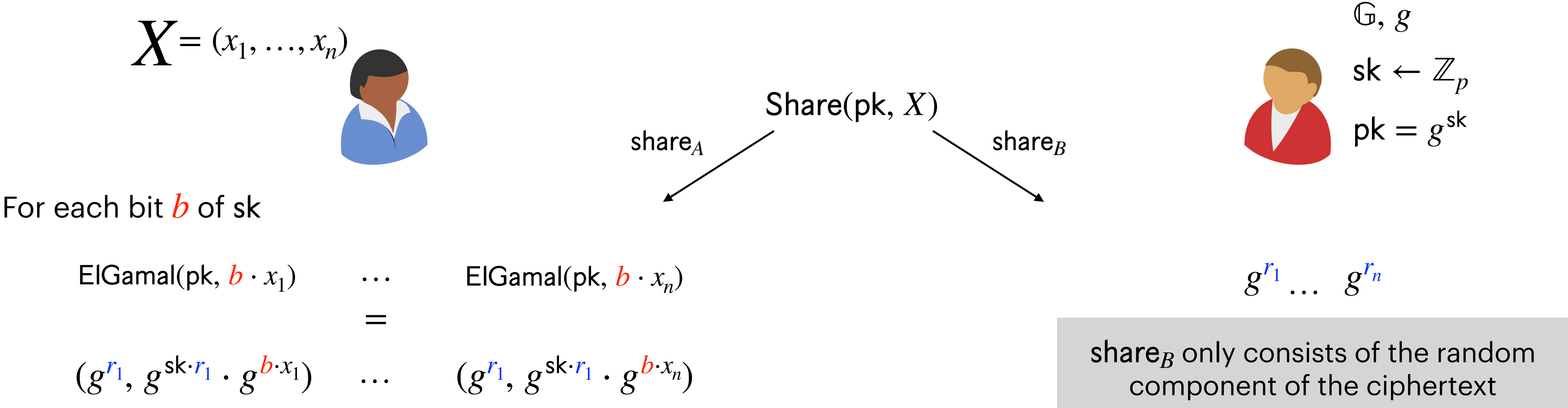
ElGamal(pk, $b \cdot x_1$) ... ElGamal(pk, $b \cdot x_n$)
=
 $(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_1})$... $(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_n})$

$g^{r_1} \dots g^{r_n}$

share_B only consists of the random component of the ciphertext

Succinct Distribution of Staged Input Shares

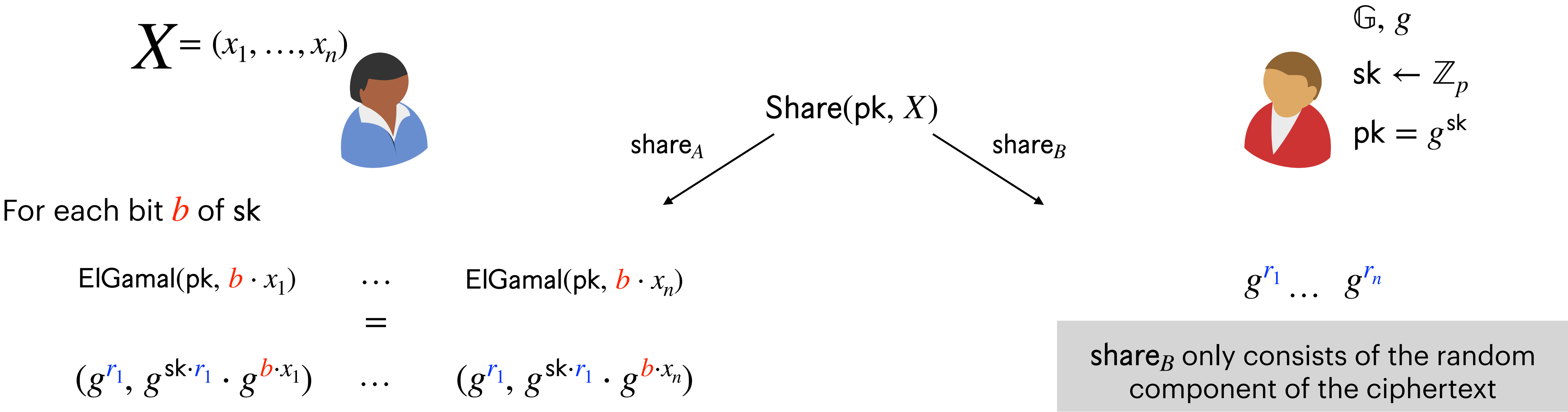
Structure of Staged Input Shares



Attempt:

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

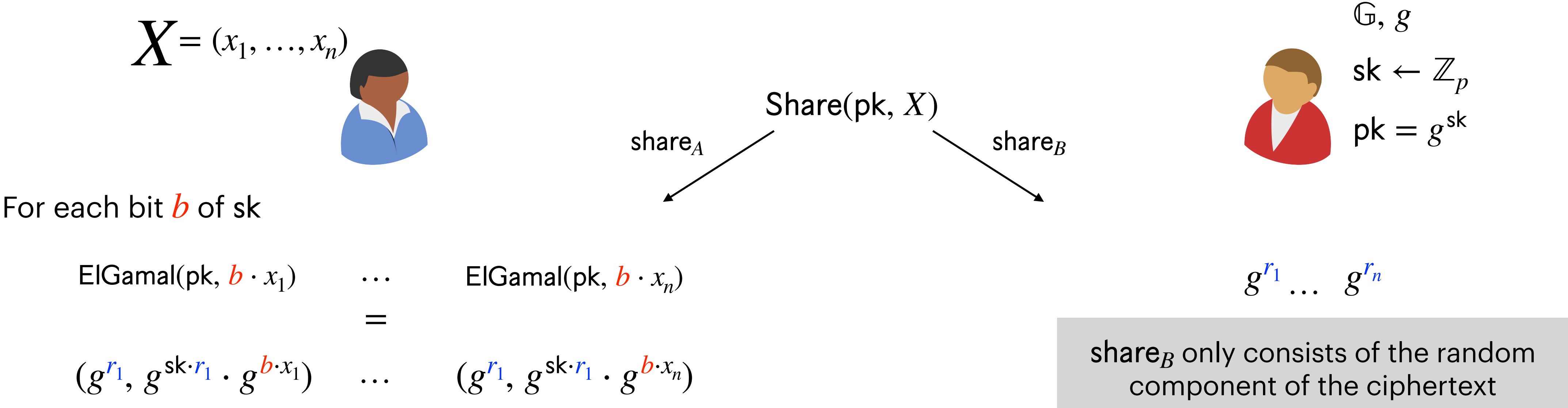


Attempt:

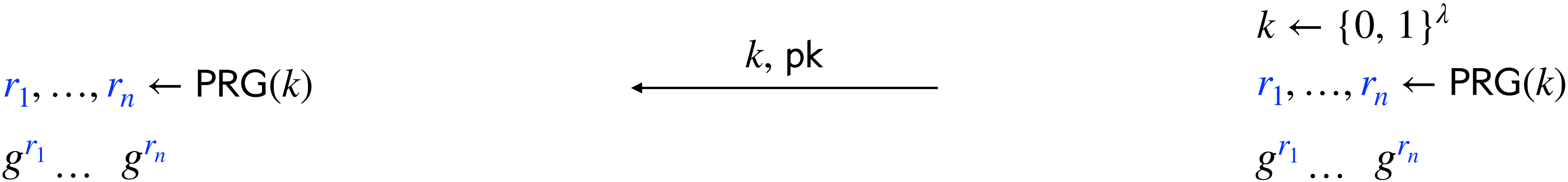
$k \leftarrow \{0, 1\}^\lambda$
 $r_1, \dots, r_n \leftarrow \text{PRG}(k)$
 $g^{r_1} \dots g^{r_n}$

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

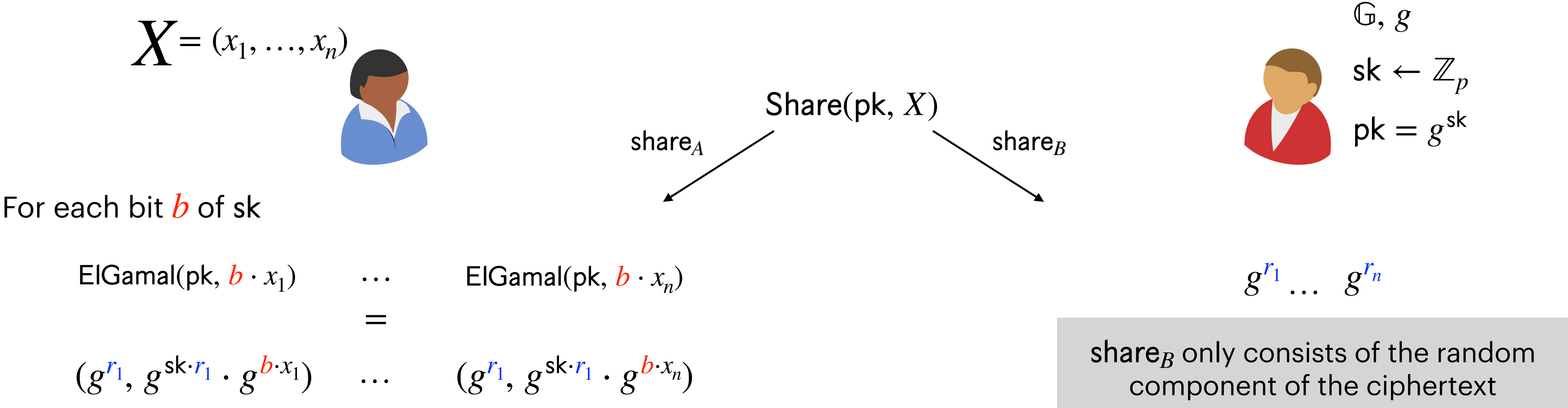


Attempt:

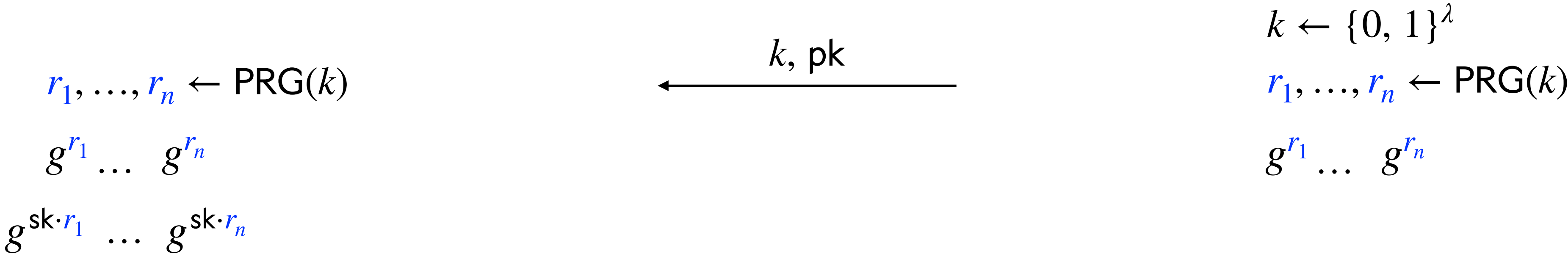


Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



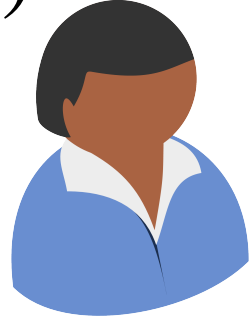
Attempt:

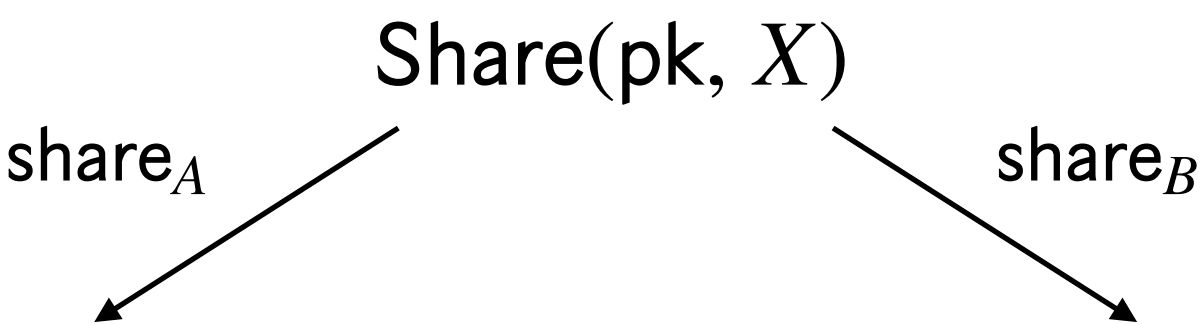


Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$X = (x_1, \dots, x_n)$







$\mathbb{G}, g$
 $\text{sk} \leftarrow \mathbb{Z}_p$
 $\text{pk} = g^{\text{sk}}$

For each bit b of sk

$\text{ElGamal}(\text{pk}, b \cdot x_1) \quad \dots \quad \text{ElGamal}(\text{pk}, b \cdot x_n)$

$=$

$(g^{r_1}, g^{\text{sk} \cdot r_1} \cdot g^{b \cdot x_1}) \quad \dots \quad (g^{r_1}, g^{\text{sk} \cdot r_1} \cdot g^{b \cdot x_n})$

$g^{r_1} \dots g^{r_n}$

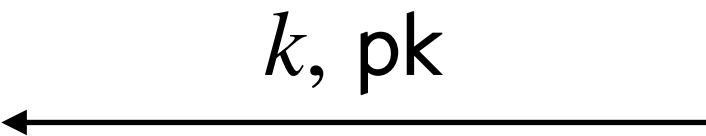
share_B only consists of the random component of the ciphertext

Attempt:

$r_1, \dots, r_n \leftarrow \text{PRG}(k)$

$g^{r_1} \dots g^{r_n}$

$g^{\text{sk} \cdot r_1} \dots g^{\text{sk} \cdot r_n}$



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

$r_1, \dots, r_n \leftarrow \text{PRG}(k)$

$g^{r_1} \dots g^{r_n}$

Alice needs g^b to compute $g^{\text{sk} \cdot r_i} \cdot g^{b \cdot x_i}$ but sending this would be **insecure**

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$$X = (x_1, \dots, x_n)$$



Share(pk, X)

share_A

share_B



$\mathbb{G}, g$

$sk \leftarrow \mathbb{Z}_p$

$pk = g^{sk}$

For each bit b of sk

ElGamal(pk, $b \cdot x_1$) ... ElGamal(pk, $b \cdot x_n$)

=

$(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_1})$... $(g^{r_1}, g^{sk \cdot r_1} \cdot g^{b \cdot x_n})$

$g^{r_1} \dots g^{r_n}$

share_B only consists of the random component of the ciphertext

Observation: Can be computed succinctly using [techniques from Trapdoor Hashing](#) [Döttling-Garg-Ishai-Malavolta-Mour-Ostrovsky'19]

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

$$X = (x_1, \dots, x_n)$$



$$\mathbb{G}, g$$

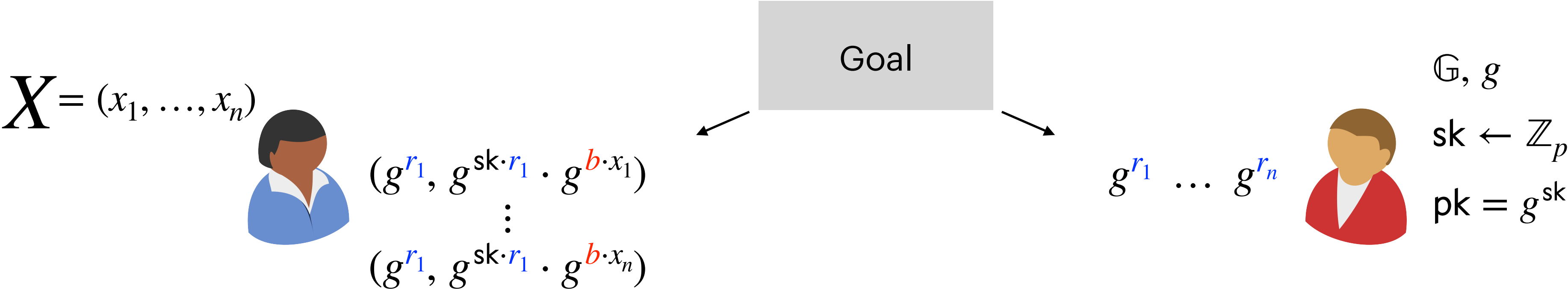
$$\text{sk} \leftarrow \mathbb{Z}_p$$

$$\text{pk} = g^{\text{sk}}$$



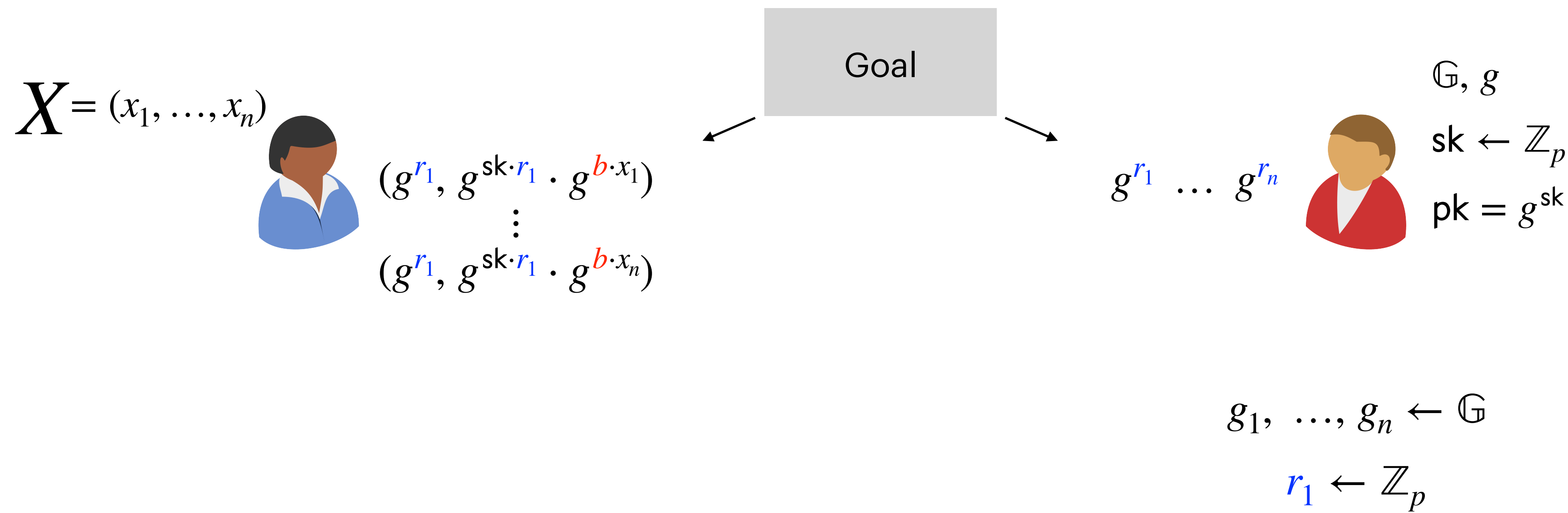
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



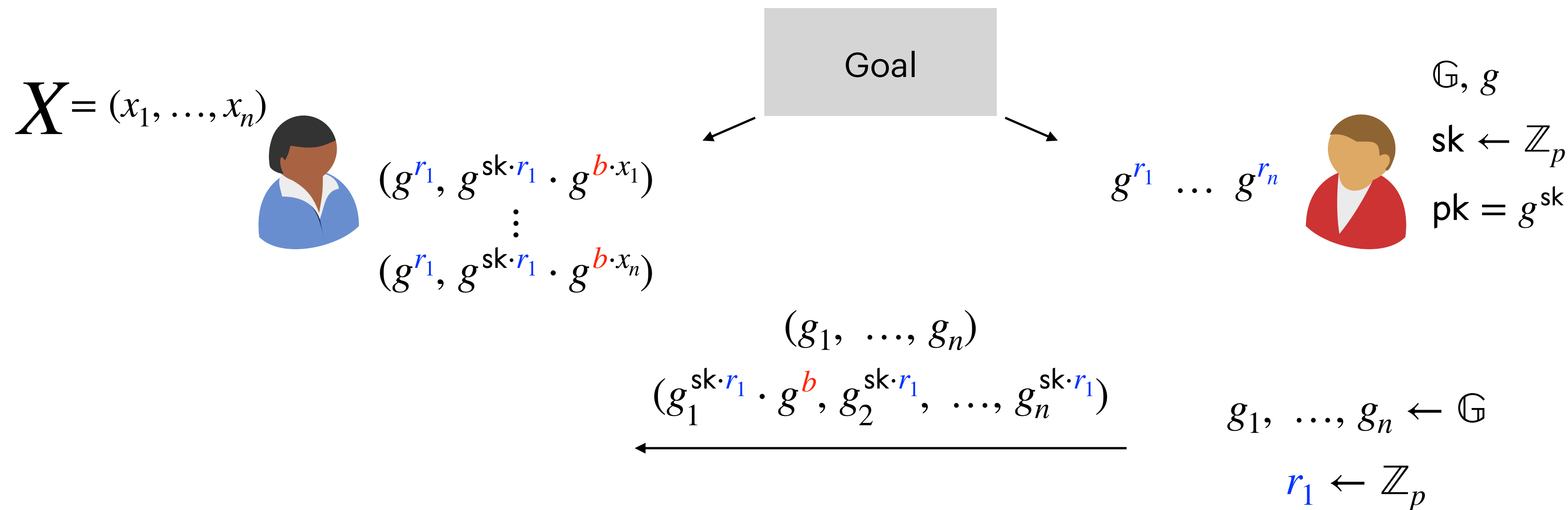
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



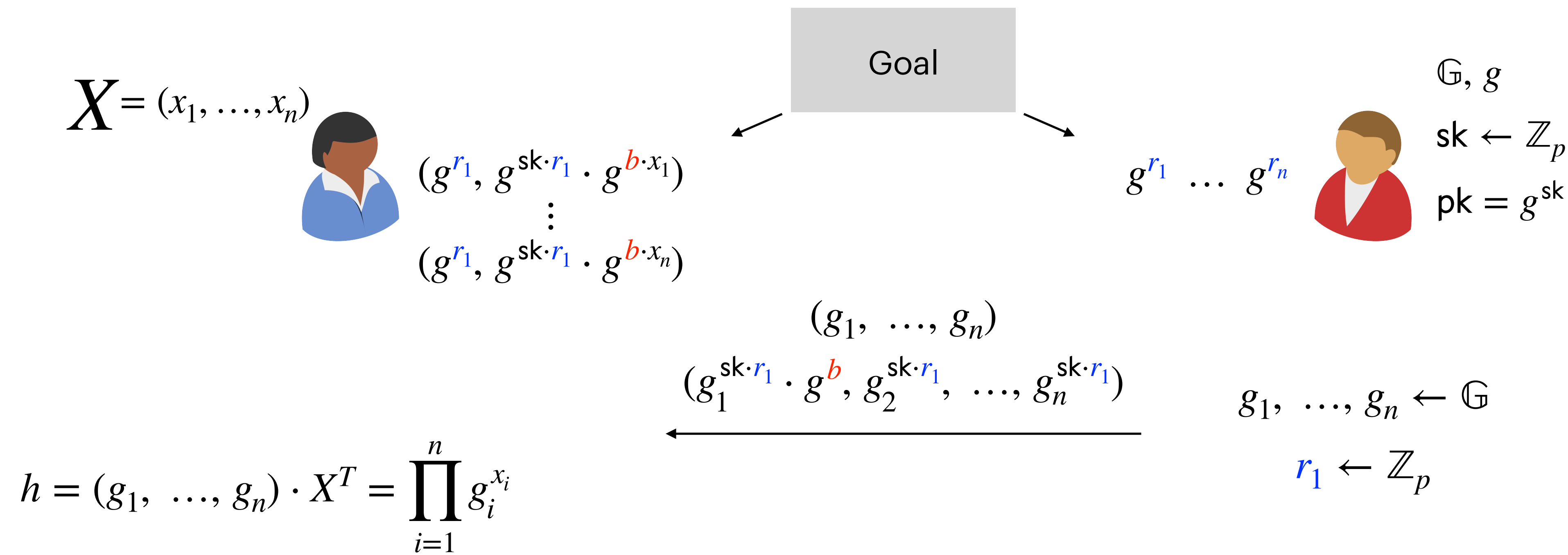
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



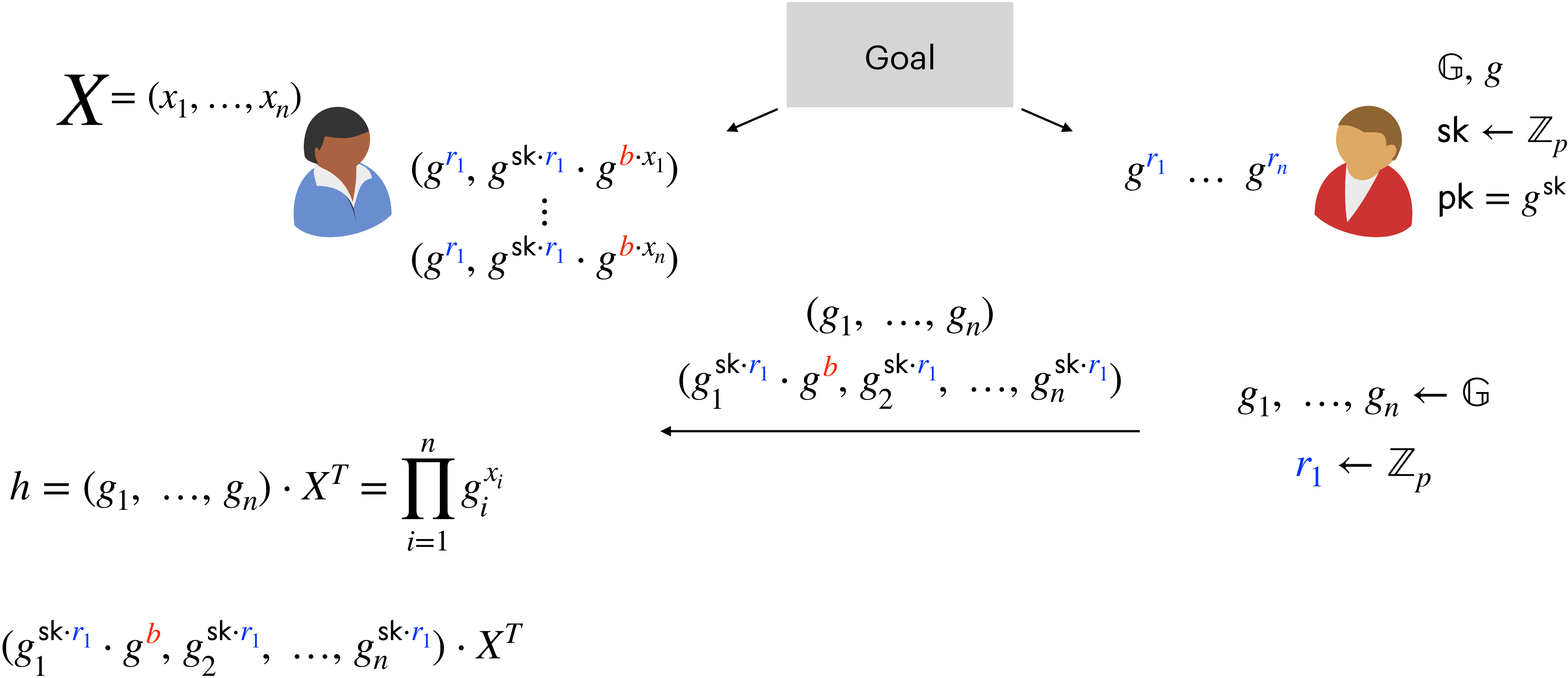
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



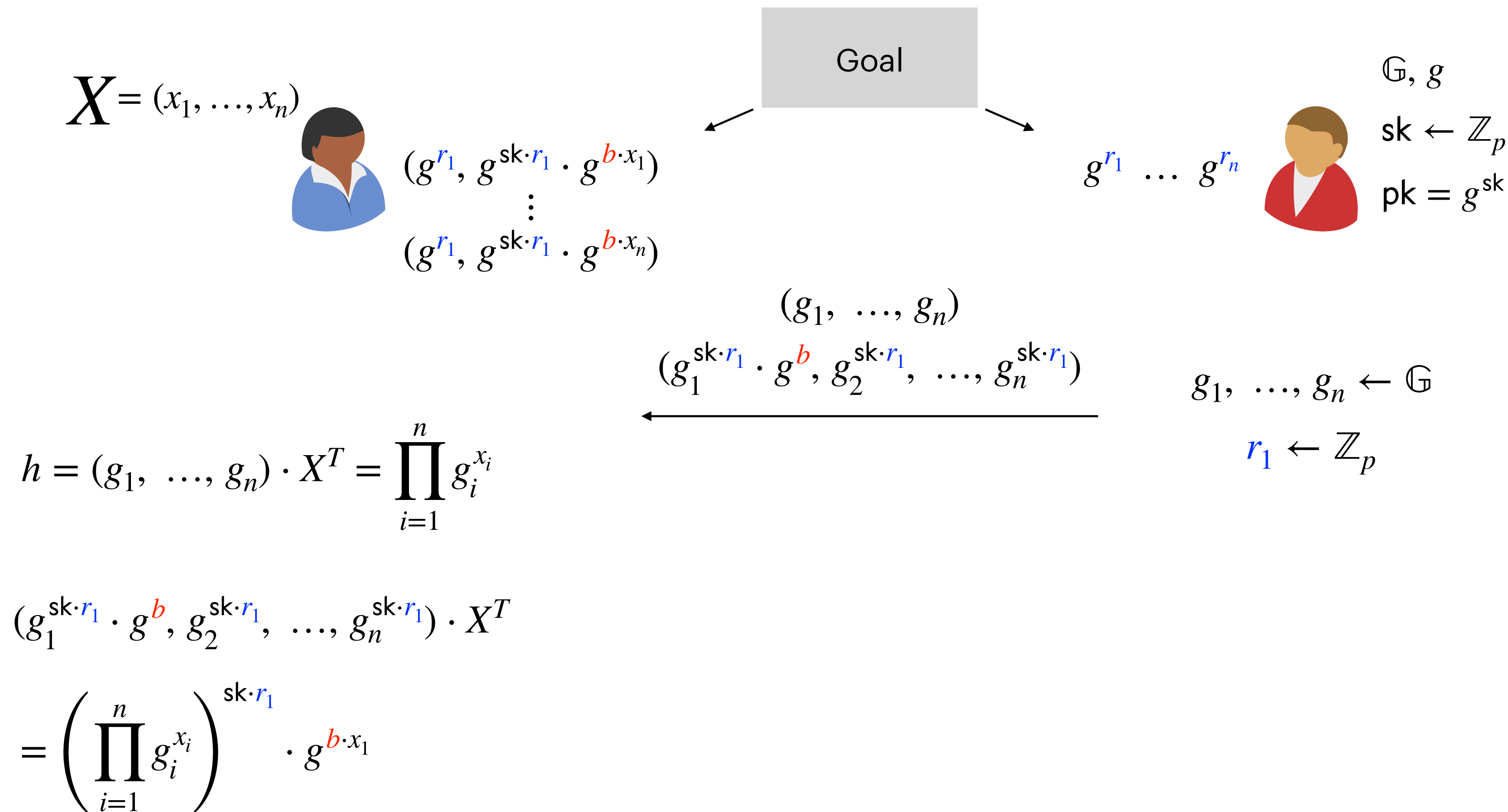
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



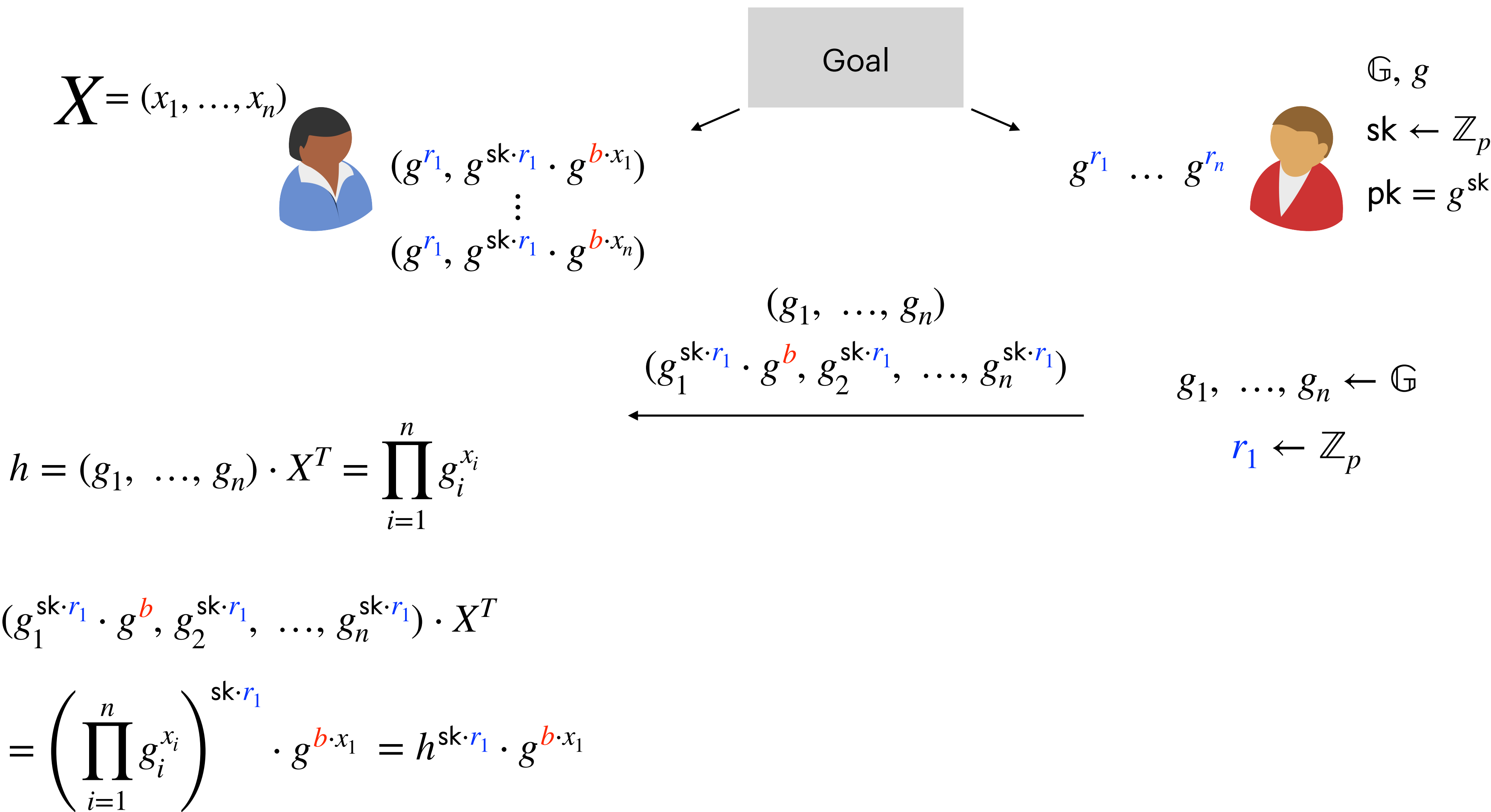
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



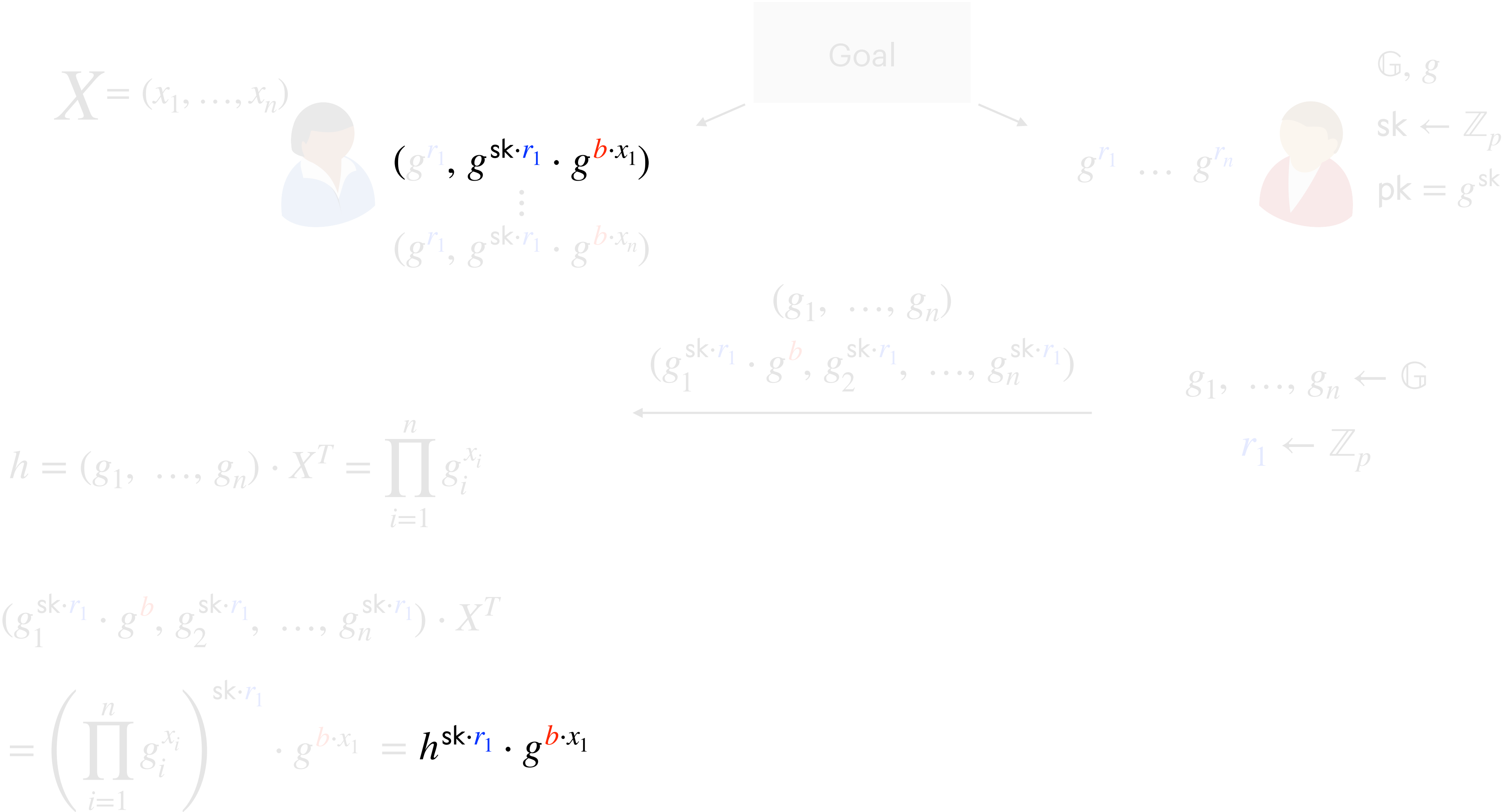
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



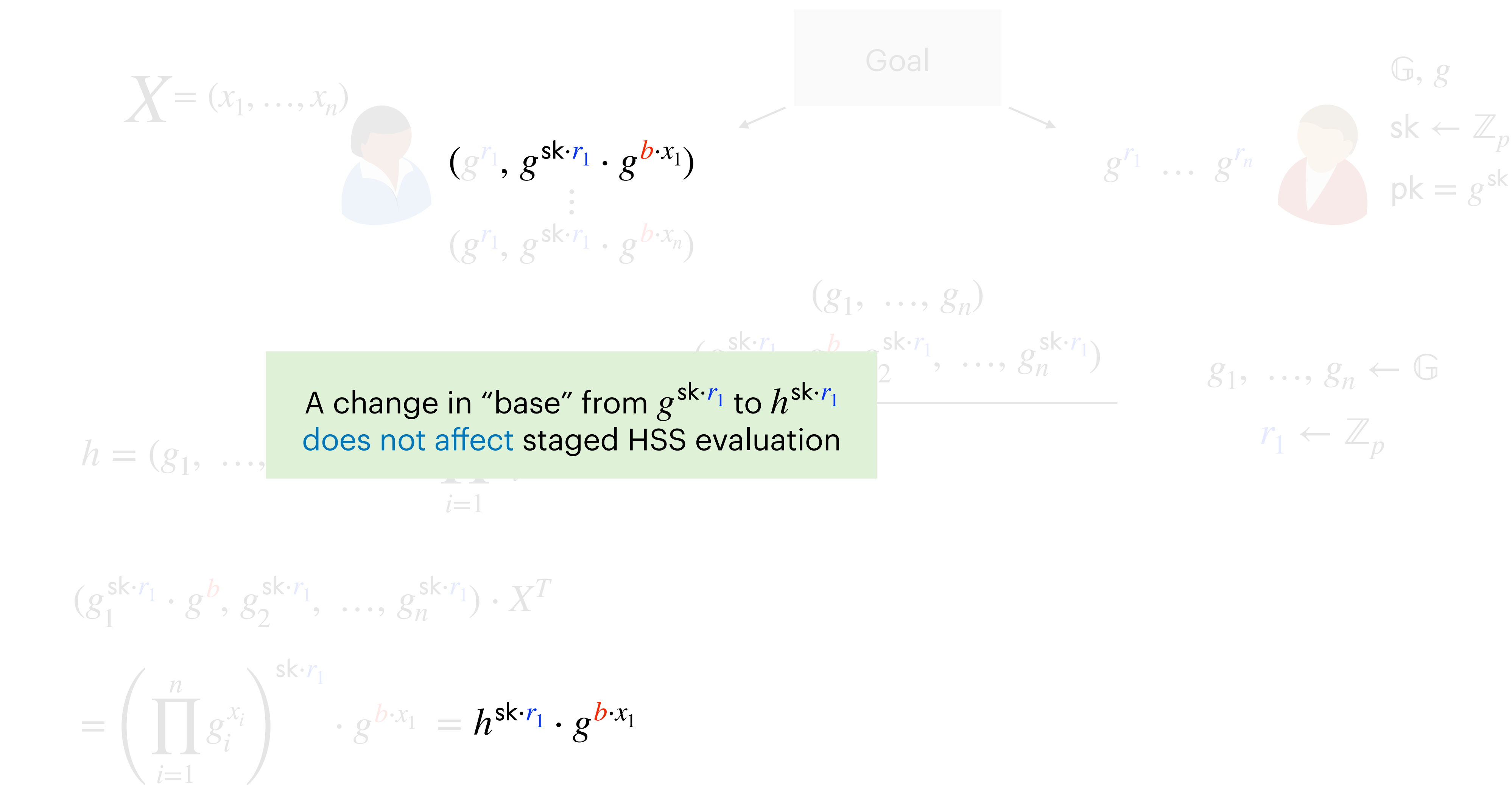
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



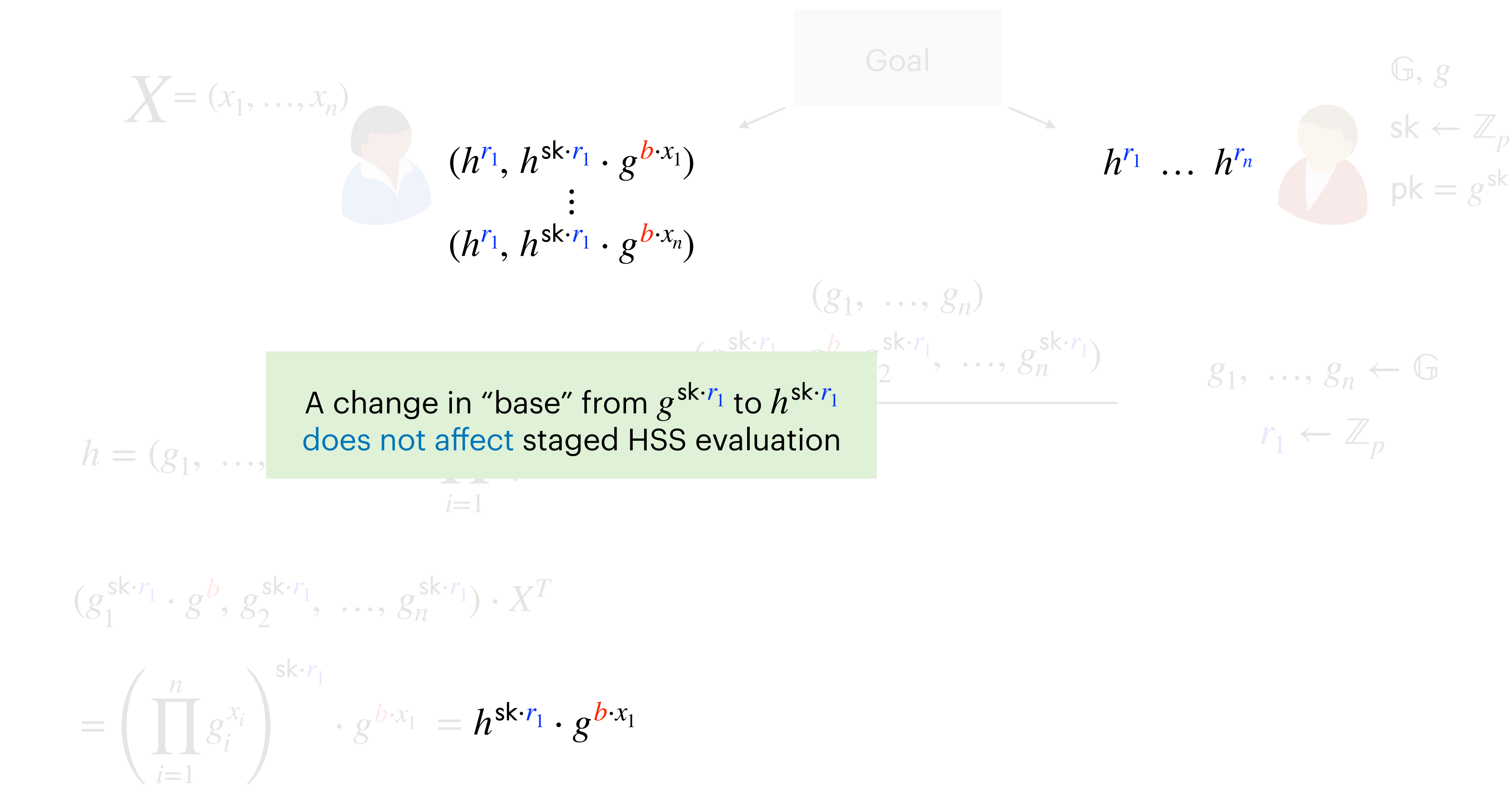
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



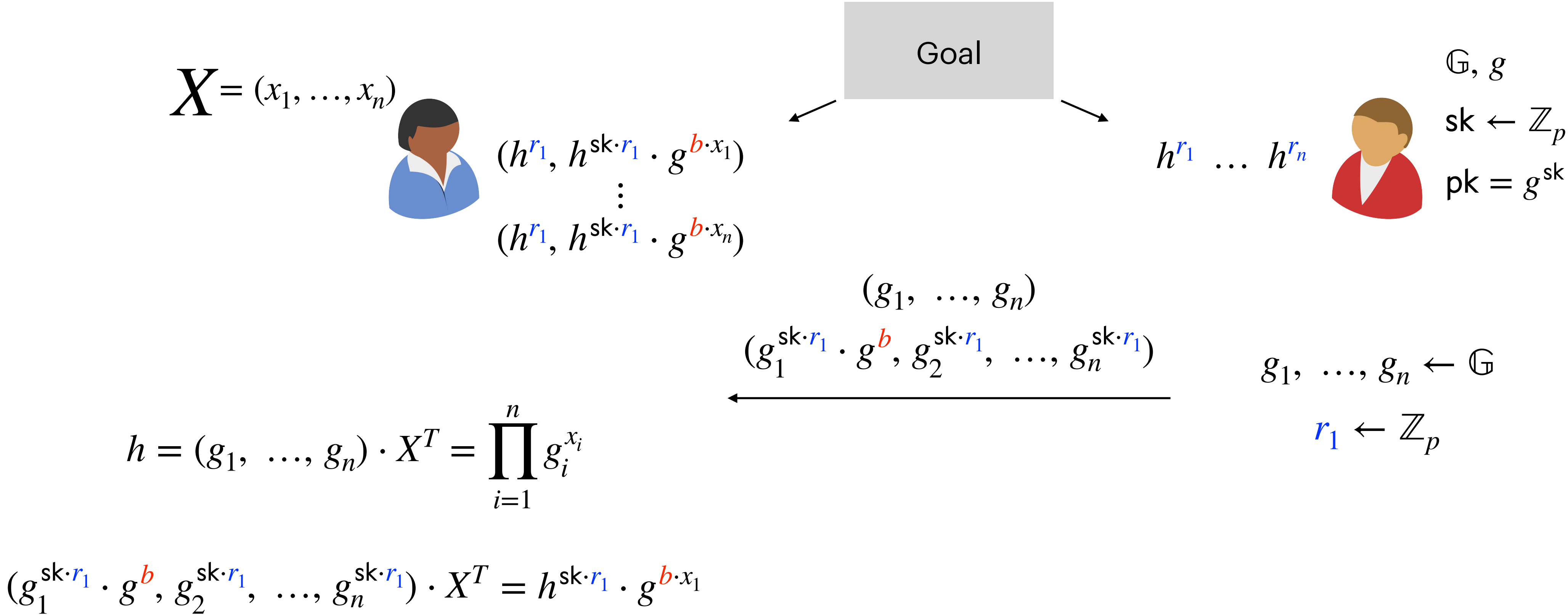
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



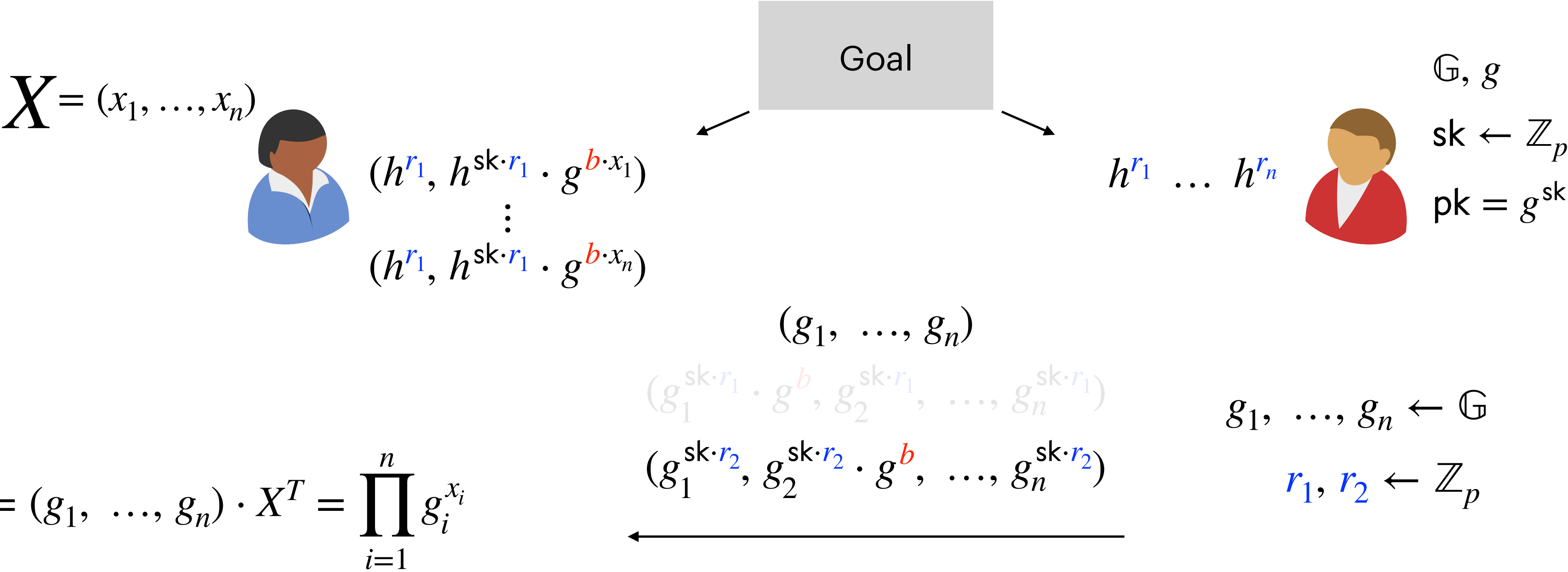
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

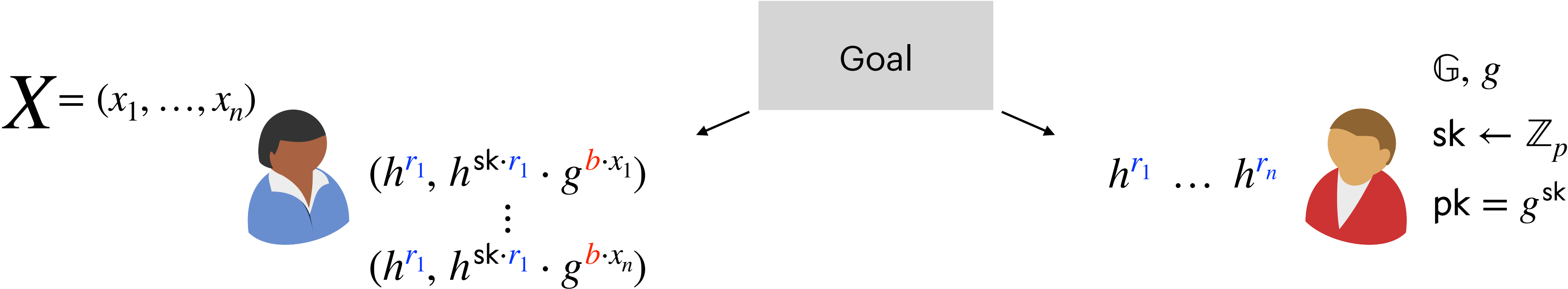


$$(g_1^{sk \cdot r_1} \cdot g^b, g_2^{sk \cdot r_1}, \dots, g_n^{sk \cdot r_1}) \cdot X^T = h^{sk \cdot r_1} \cdot g^{b \cdot x_1}$$

$$(g_1^{sk \cdot r_2}, g_2^{sk \cdot r_2} \cdot g^b, \dots, g_n^{sk \cdot r_2}) \cdot X^T = h^{sk \cdot r_2} \cdot g^{b \cdot x_2}$$

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



$$h = (g_1, \dots, g_n) \cdot X^T = \prod_{i=1}^n g_i^{x_i}$$

$(g_1, \dots, g_n)$
 $(g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1})$
 $\vdots$
 $(g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n}, \dots, g_n^{\text{sk} \cdot r_n} \cdot g^b)$

$g_1, \dots, g_n \leftarrow \mathbb{G}$
 $r_1, \dots, r_n \leftarrow \mathbb{Z}_p$

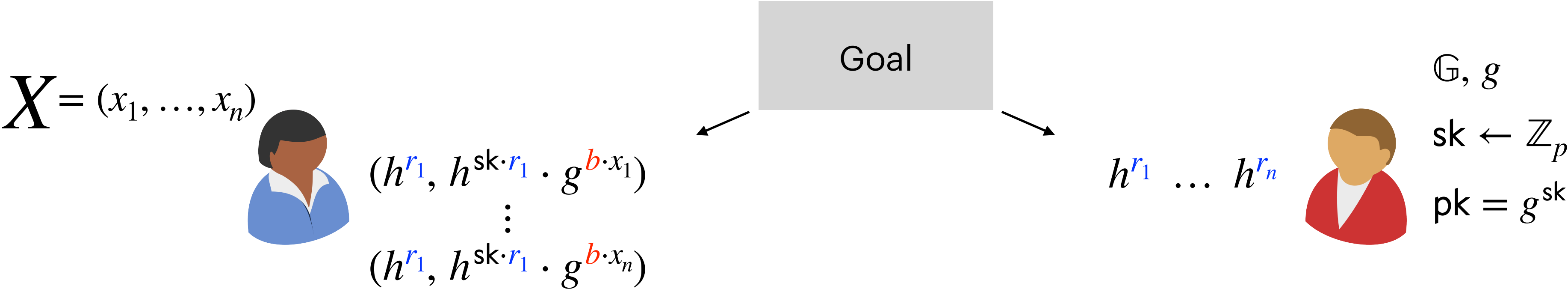
$$(g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1}) \cdot X^T = h^{\text{sk} \cdot r_1} \cdot g^{b \cdot x_1}$$

$\vdots$

$$(g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n} \cdot g^b, \dots, g_n^{\text{sk} \cdot r_n}) \cdot X^T = h^{\text{sk} \cdot r_n} \cdot g^{b \cdot x_n}$$

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



$$h = (g_1, \dots, g_n) \cdot X^T = \prod_{i=1}^n g_i^{x_i}$$

$$\begin{aligned} & (g_1, \dots, g_n) \\ & (g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1}) \\ & \vdots \\ & (g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n}, \dots, g_n^{\text{sk} \cdot r_n} \cdot g^b) \end{aligned}$$

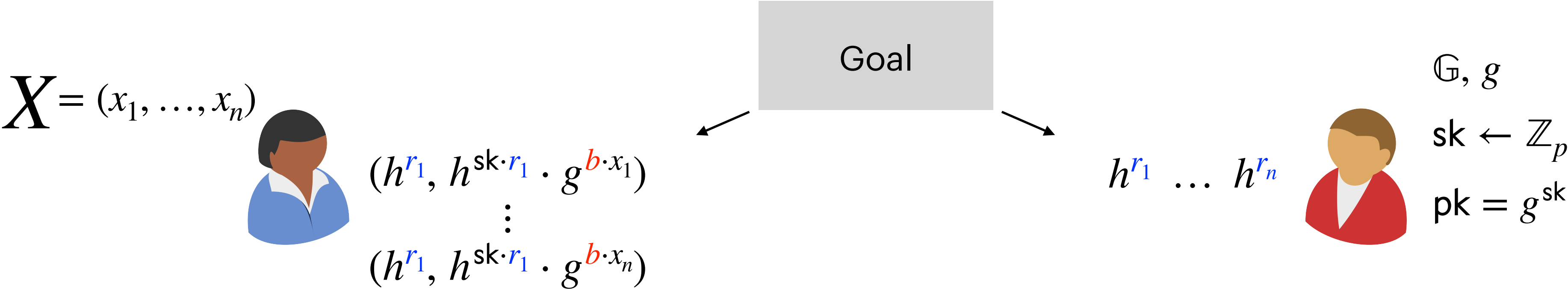
$$\begin{aligned} g_1, \dots, g_n &\leftarrow \mathbb{G} \\ r_1, \dots, r_n &\leftarrow \mathbb{Z}_p \end{aligned}$$

$$\begin{aligned} (g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1}) \cdot X^T &= h^{\text{sk} \cdot r_1} \cdot g^{b \cdot x_1} \\ &\vdots \\ (g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n} \cdot g^b, \dots, g_n^{\text{sk} \cdot r_n}) \cdot X^T &= h^{\text{sk} \cdot r_n} \cdot g^{b \cdot x_n} \end{aligned}$$

$h^{r_1}, \dots, h^{r_n}$ can be computed similarly

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



$$h = (g_1, \dots, g_n) \cdot X^T = \prod_{i=1}^n g_i^{x_i}$$

$$\begin{aligned} & (g_1, \dots, g_n) \\ & (g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1}) \\ & \vdots \\ & (g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n}, \dots, g_n^{\text{sk} \cdot r_n} \cdot g^b) \end{aligned}$$

$$\begin{aligned} g_1, \dots, g_n &\leftarrow \mathbb{G} \\ r_1, \dots, r_n &\leftarrow \mathbb{Z}_p \end{aligned}$$

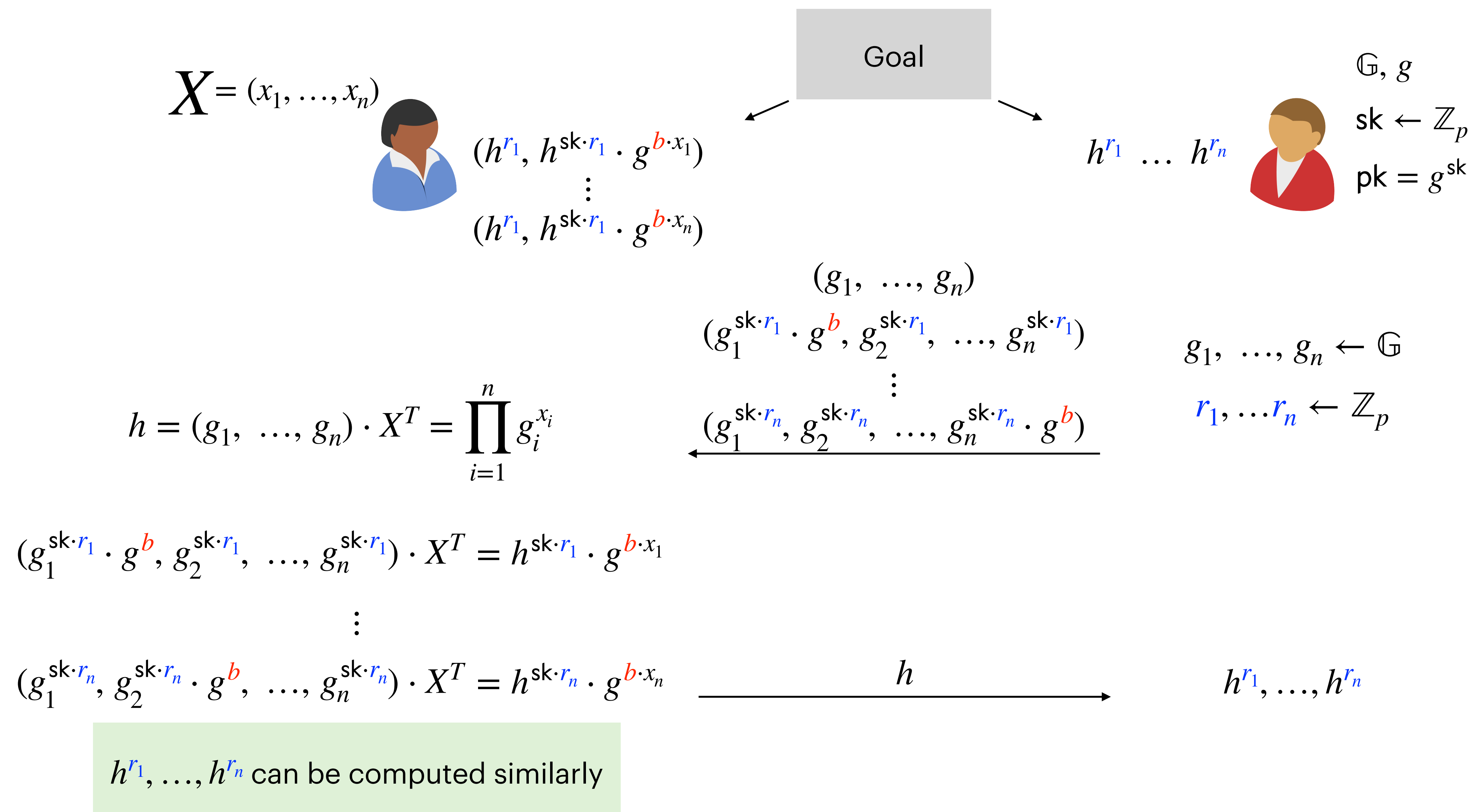
$$\begin{aligned} & (g_1^{\text{sk} \cdot r_1} \cdot g^b, g_2^{\text{sk} \cdot r_1}, \dots, g_n^{\text{sk} \cdot r_1}) \cdot X^T = h^{\text{sk} \cdot r_1} \cdot g^{b \cdot x_1} \\ & \vdots \\ & (g_1^{\text{sk} \cdot r_n}, g_2^{\text{sk} \cdot r_n} \cdot g^b, \dots, g_n^{\text{sk} \cdot r_n}) \cdot X^T = h^{\text{sk} \cdot r_n} \cdot g^{b \cdot x_n} \end{aligned}$$

_____ h _____>

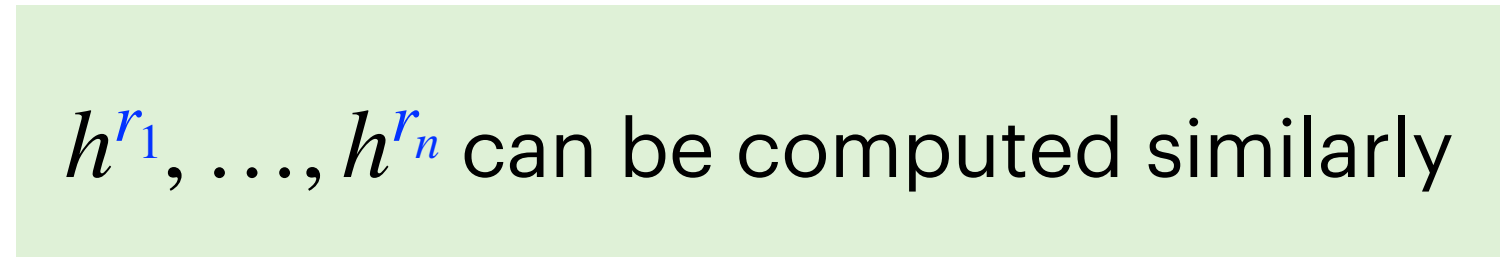
$h^{r_1}, \dots, h^{r_n}$ can be computed similarly

Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares

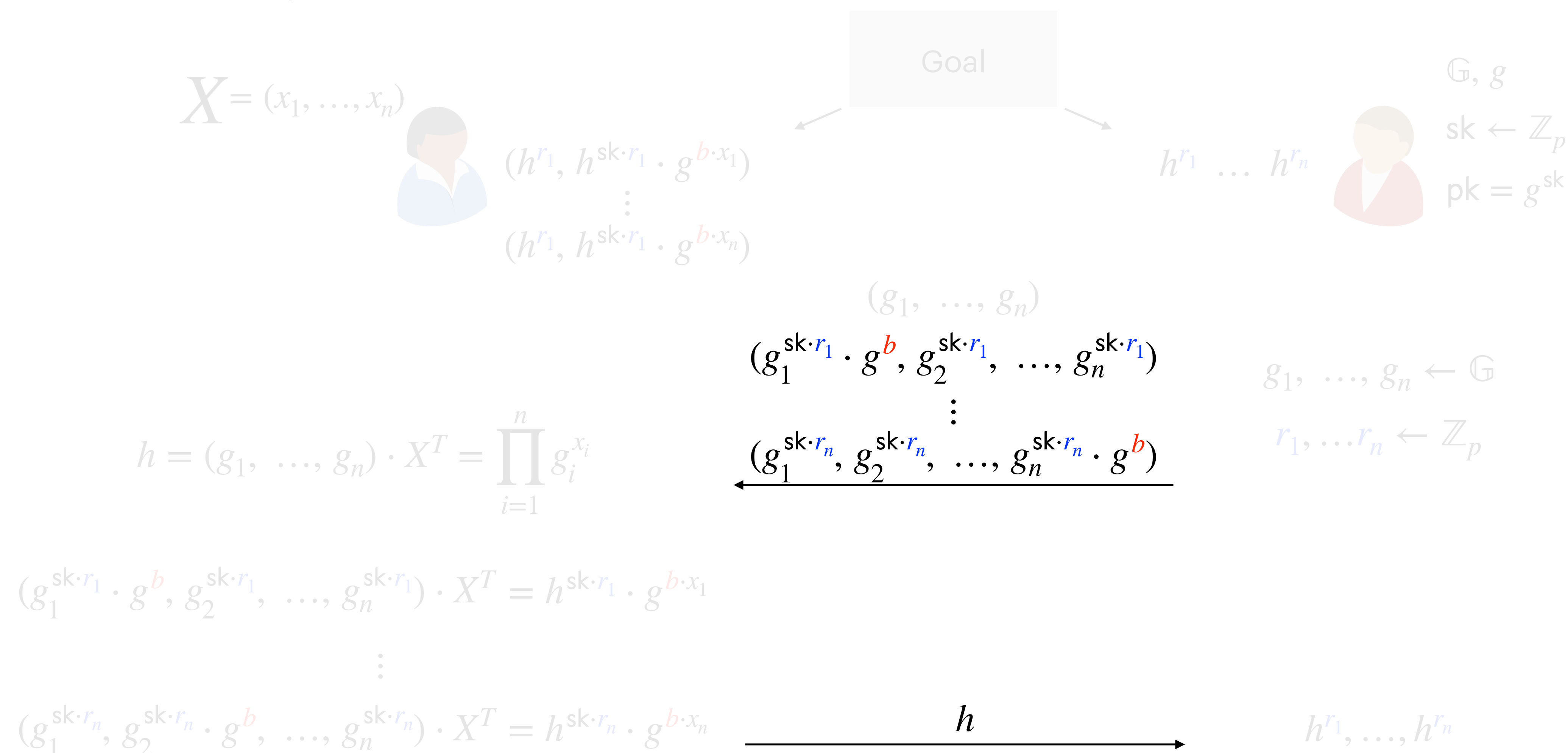


Structure of Staged Input Shares



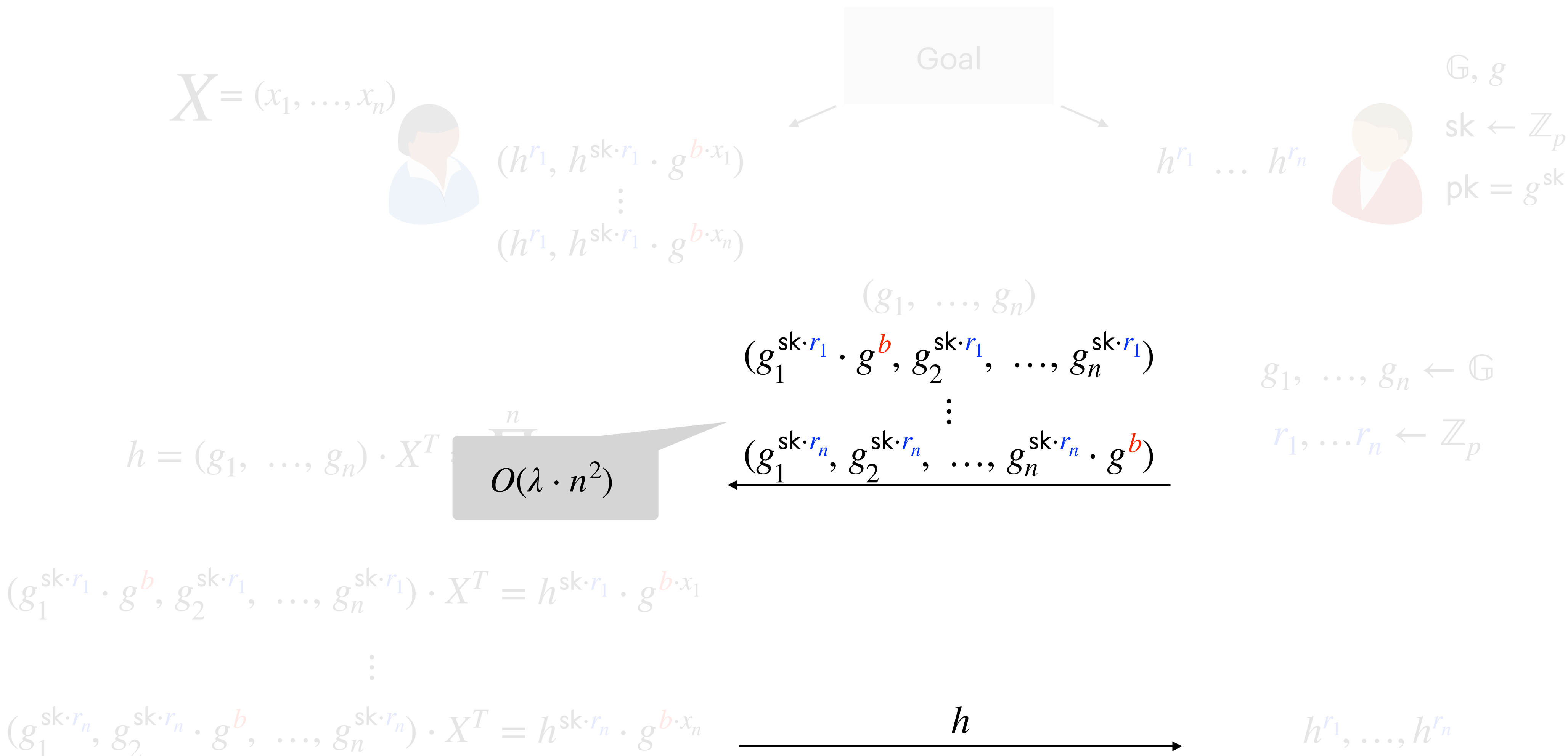
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



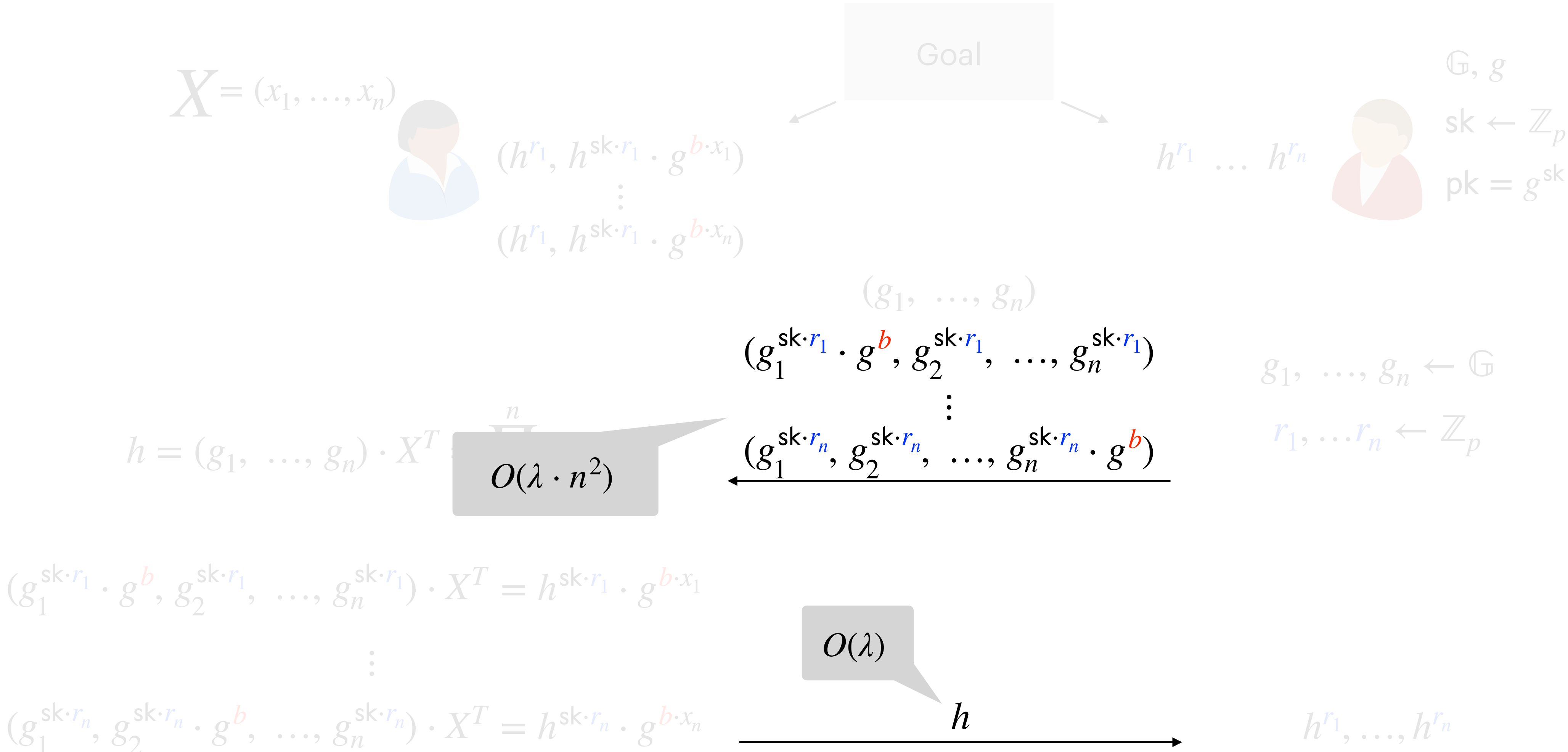
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



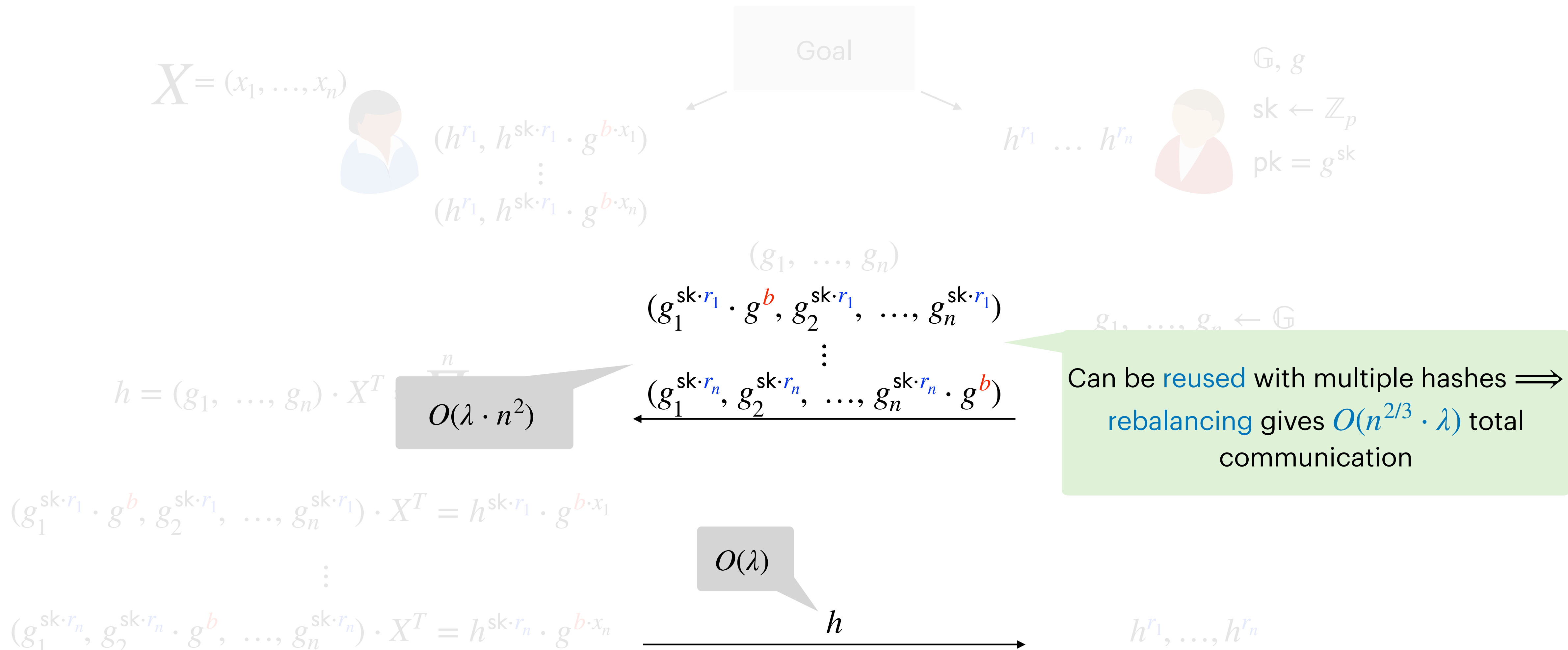
Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



Succinct Distribution of Staged Input Shares

Structure of Staged Input Shares



Conclusion

- Discussed approach assumes circular security of ElGamal. Constructing from plain DDH and DCR requires extending to circular-secure variants [Boneh-Halevi-Hamburg-Ostrovsky'08] [Brakerski-Goldwasser'10]
- DDH-based Staged HSS evaluation has noticeable error probability which affects privacy. Requires developing new techniques to build trapdoor hash functions

Conclusion

- Discussed approach assumes circular security of ElGamal. Constructing from plain **DDH and DCR** requires **extending to circular-secure variants** [Boneh-Halevi-Hamburg-Ostrovsky'08] [Brakerski-Goldwasser'10]
- DDH-based **Staged HSS evaluation** has **noticeable error probability** which affects **privacy**. Requires developing new techniques to build trapdoor hash functions

Thank You