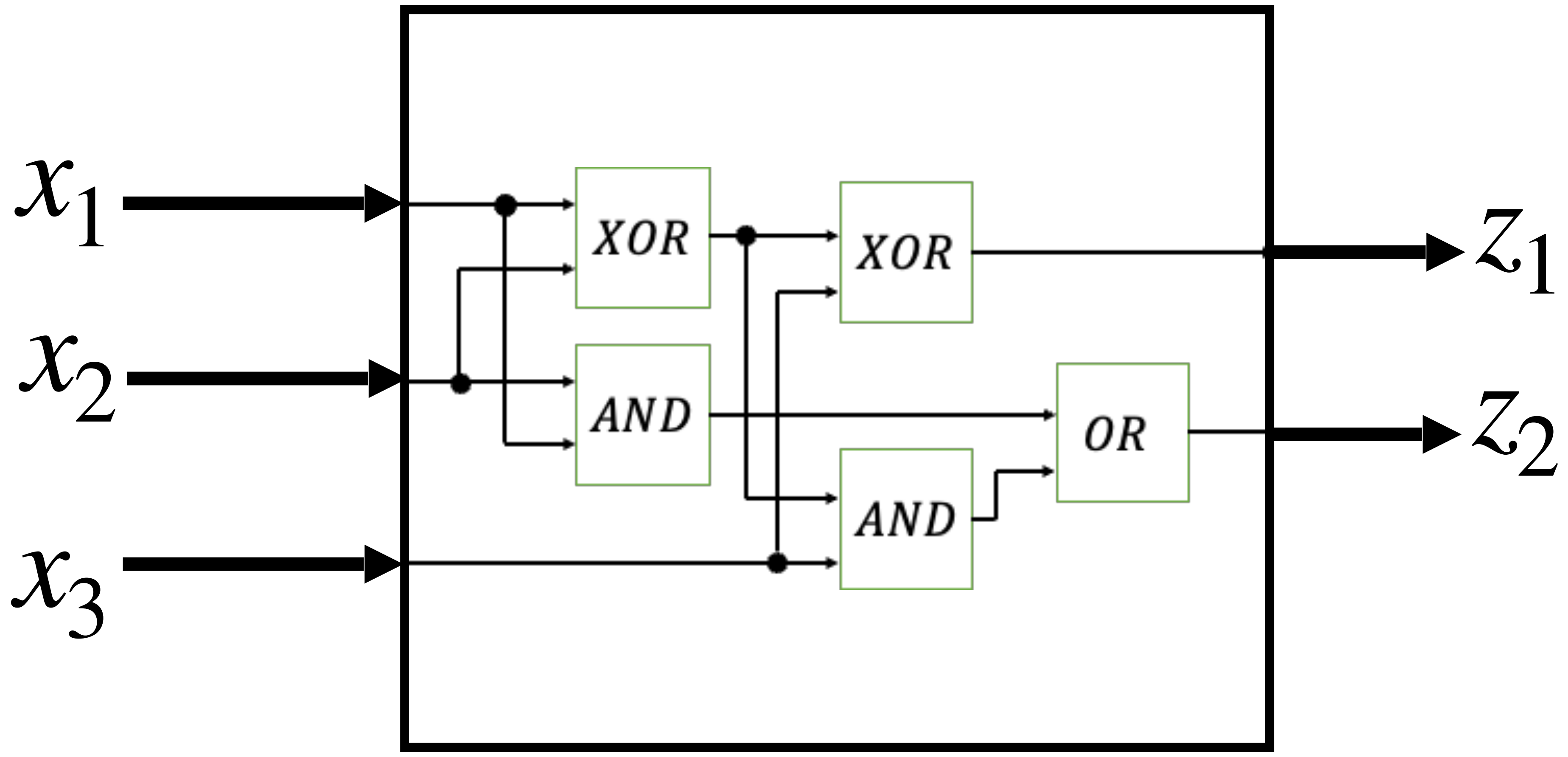
TinyLabels: How to Compress Garbled Circuit Input Labels, Efficiently

Marian Dietz Hanjun Li Huijia (Rachel) Lin

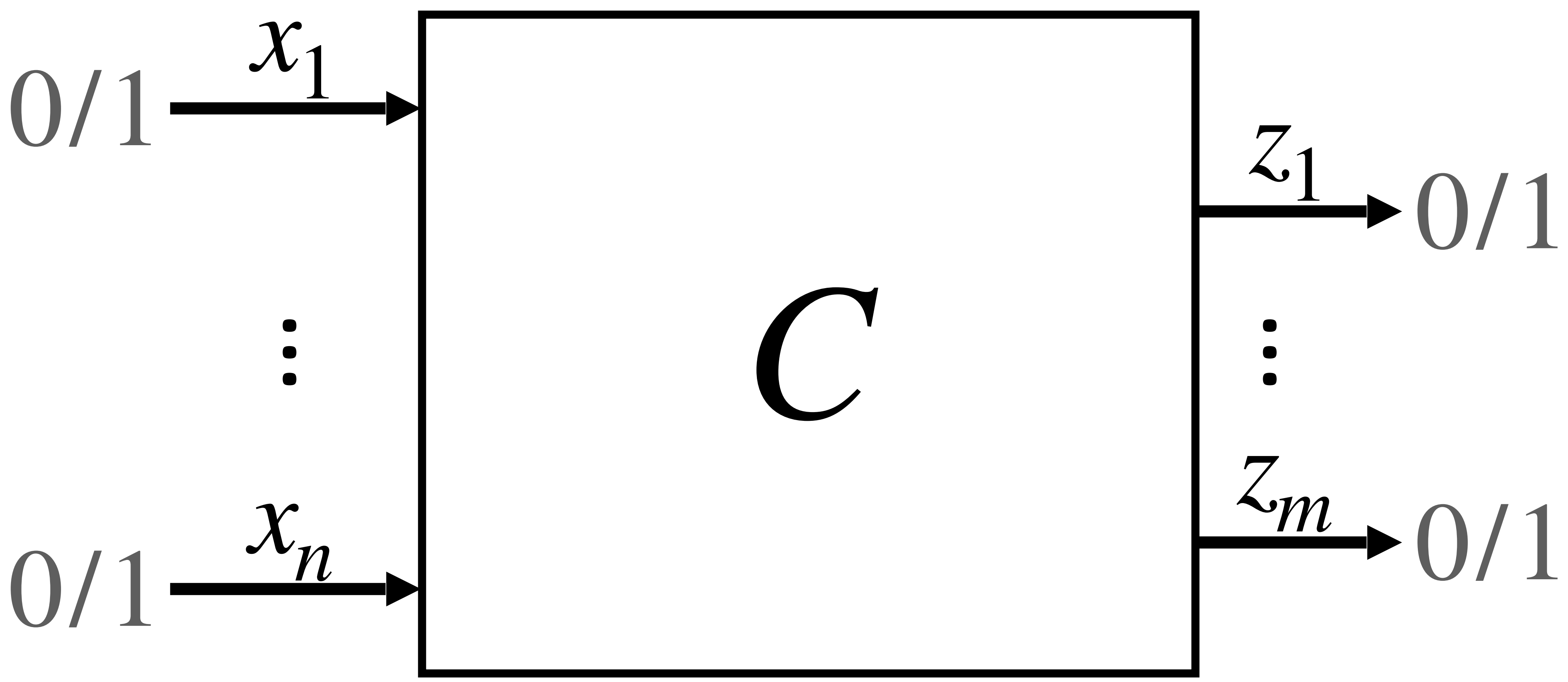
ETH Zurich University of Washington

Slides prepared together with Hanjun Li

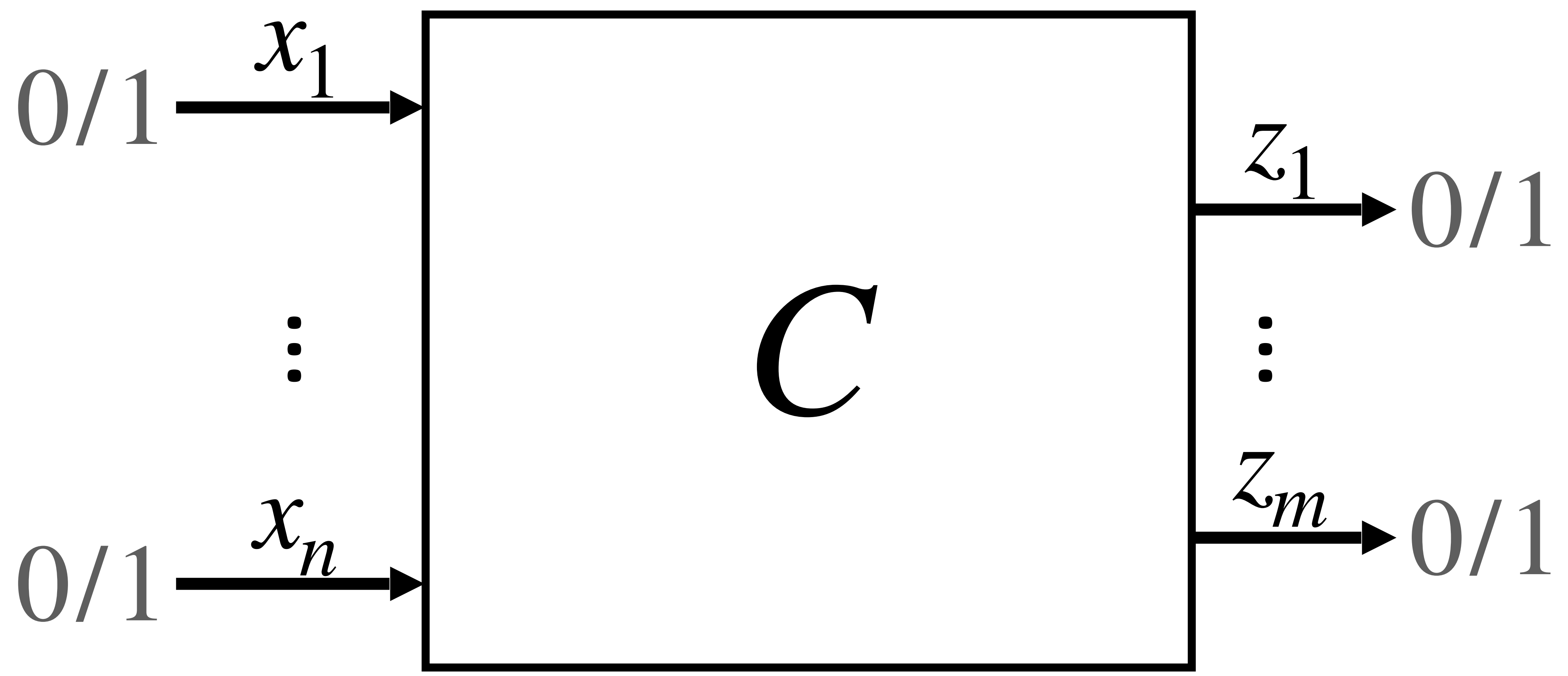
Garbling Scheme



Garbling Scheme

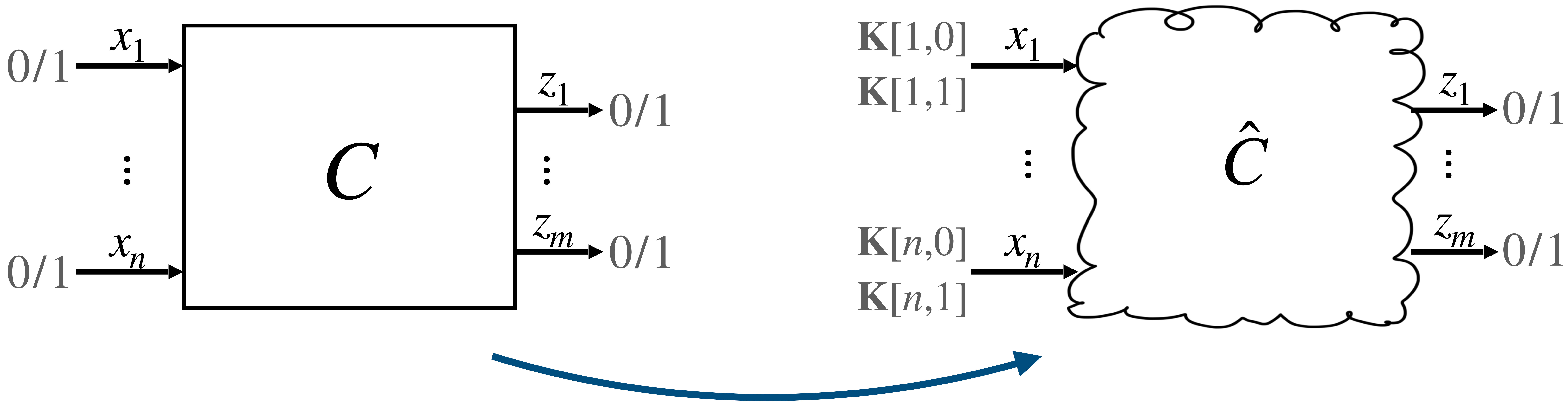


Garbling Scheme



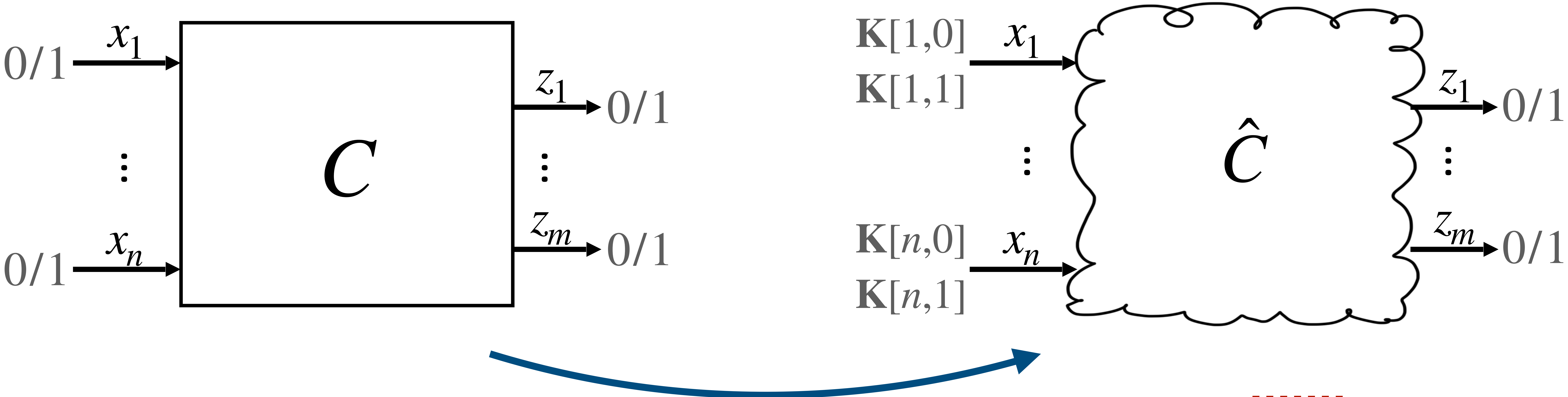
$\textsf{Garble}(C) \rightarrow \hat{C}, \mathbf{K}$

Garbling Scheme



Garble$(C) \to \hat{C}, \mathbf{K}$

Garbling Scheme

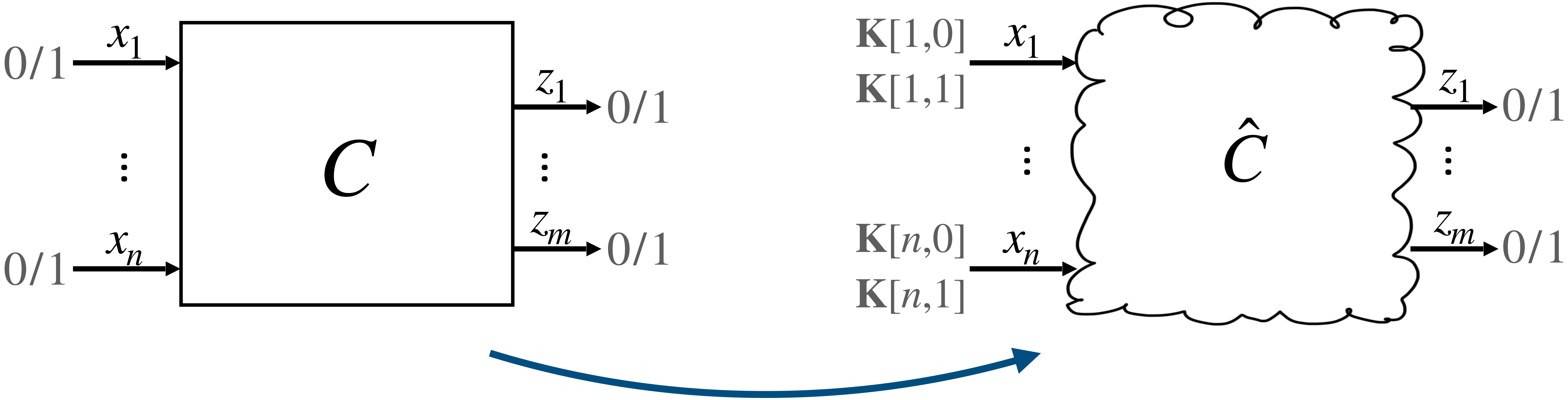


Garble$(C) \to \hat{C}, \mathbf{K}$

Eval$(\hat{C}, \boxed{\mathbf{K}[\mathbf{x}]}) \to C(\mathbf{x})$

Input Labels: One key $\mathbf{K}[i, x_i]$ per input bit

Garbling Scheme

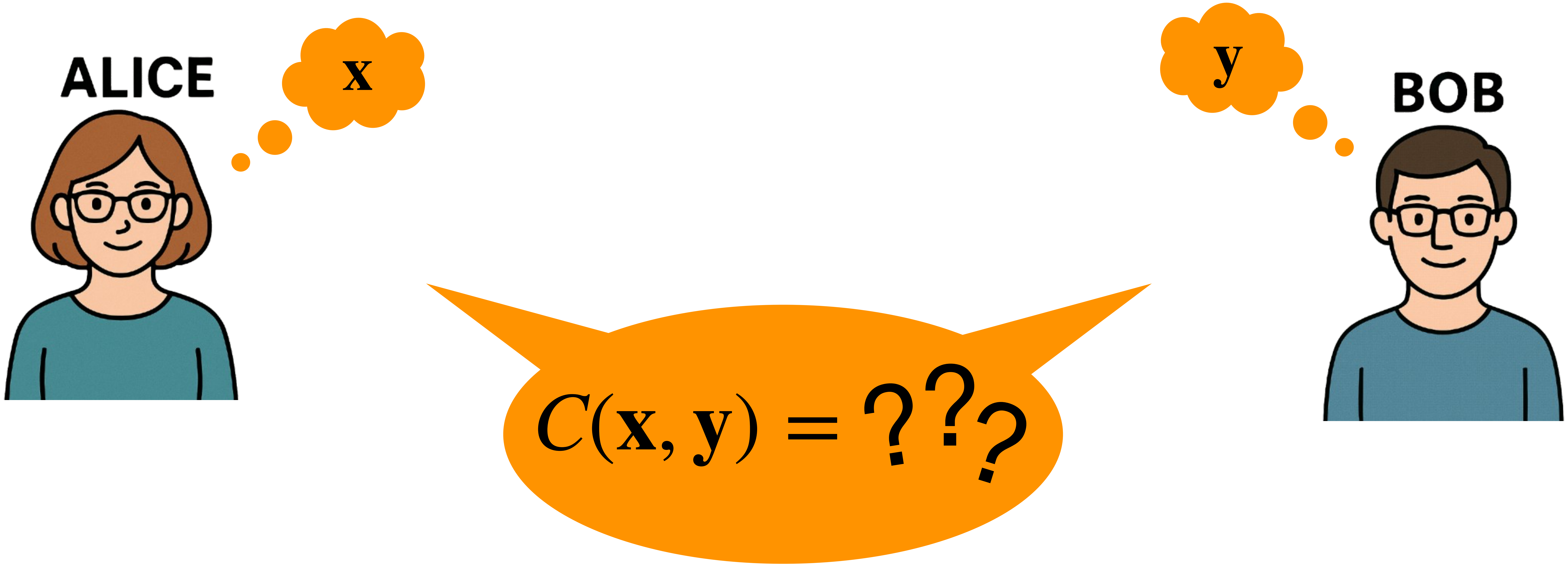


Garble$(C) \rightarrow \hat{C}, \mathbf{K}$ Eval$(\hat{C}, \mathbf{K}[\mathbf{x}]) \rightarrow C(\mathbf{x})$

Simulation Security: Sim$(C(\mathbf{x})) \rightarrow \tilde{C}, L \approx \hat{C}, \mathbf{K}[\mathbf{x}]$

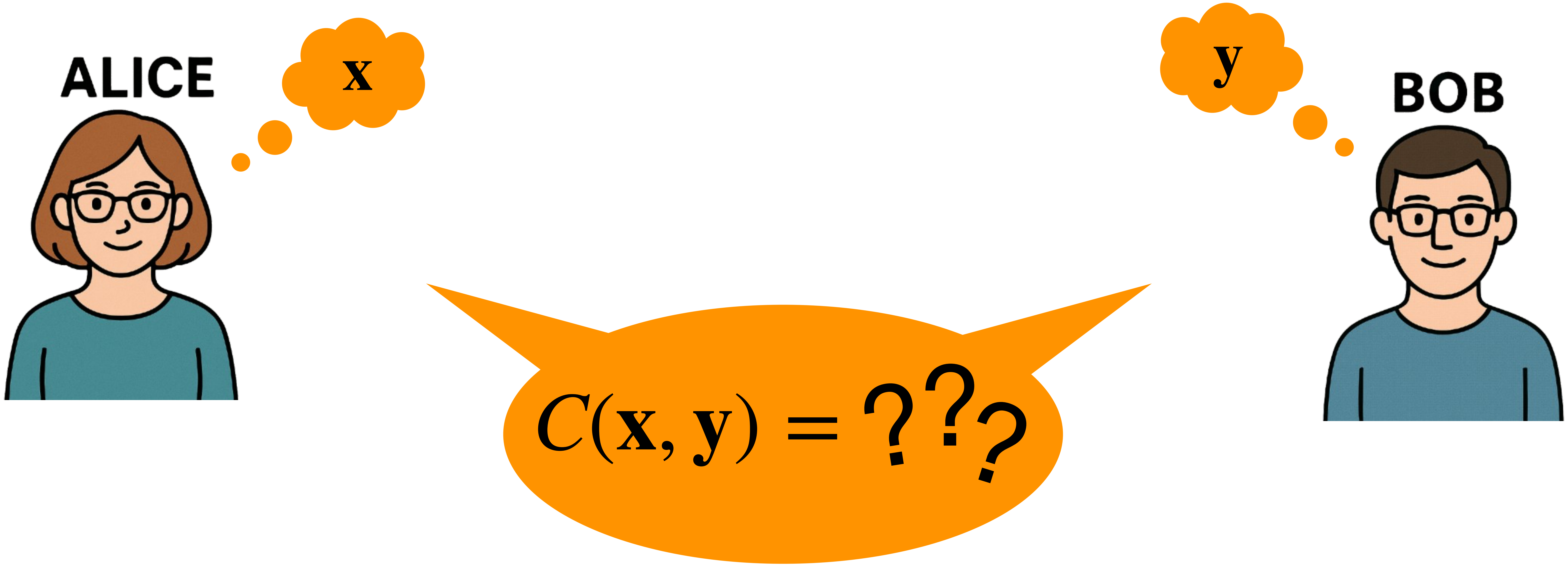
Garbling Schemes: Applications

- **Two-party computation**



Garbling Schemes: Applications

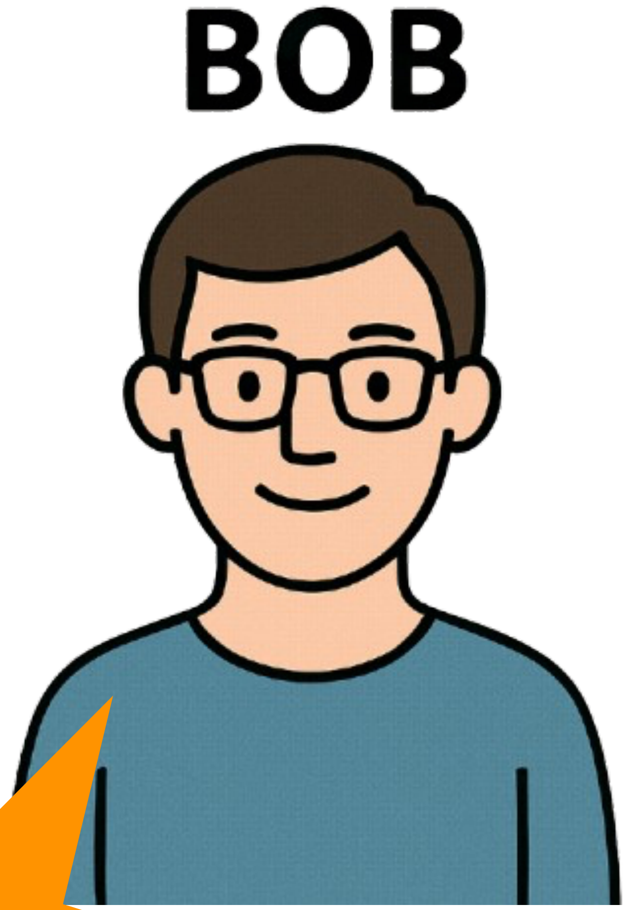
- **Two-party computation**



- **Outsourcing computation** on sensitive data

Application: Outsourcing computation

[AIKW13]



$\mathbf{x}$

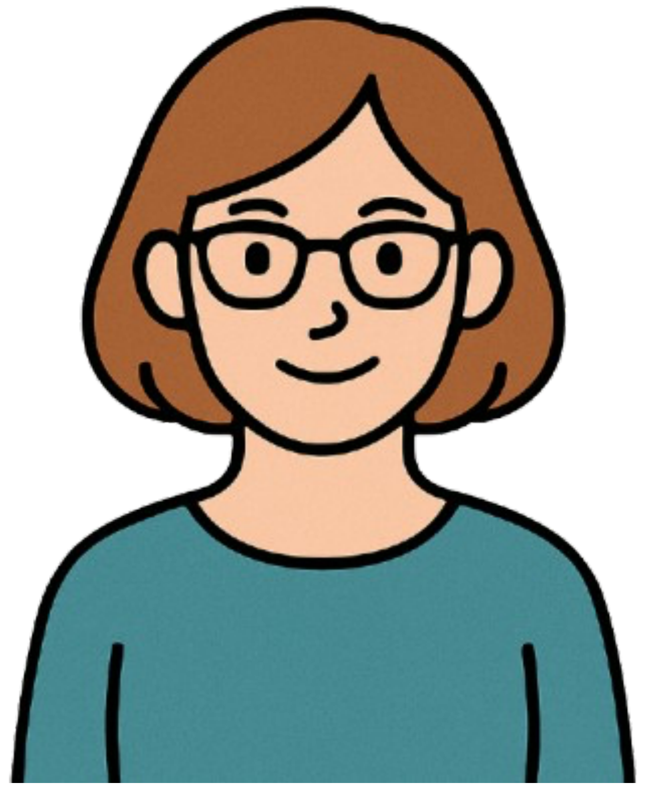
Alice has collected sensitive data $\mathbf{x}$
in a place with *limited resources*

$C(\mathbf{x}) = ???$

Application: Outsourcing computation

[AIKW13]

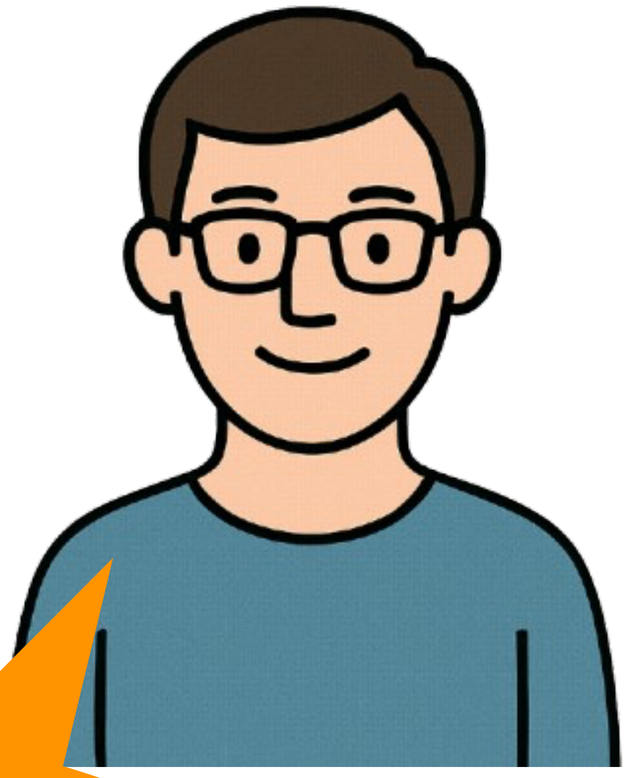
ALICE



Offline (known C)

Alice has large resources

BOB



Online

$\mathbf{x}$

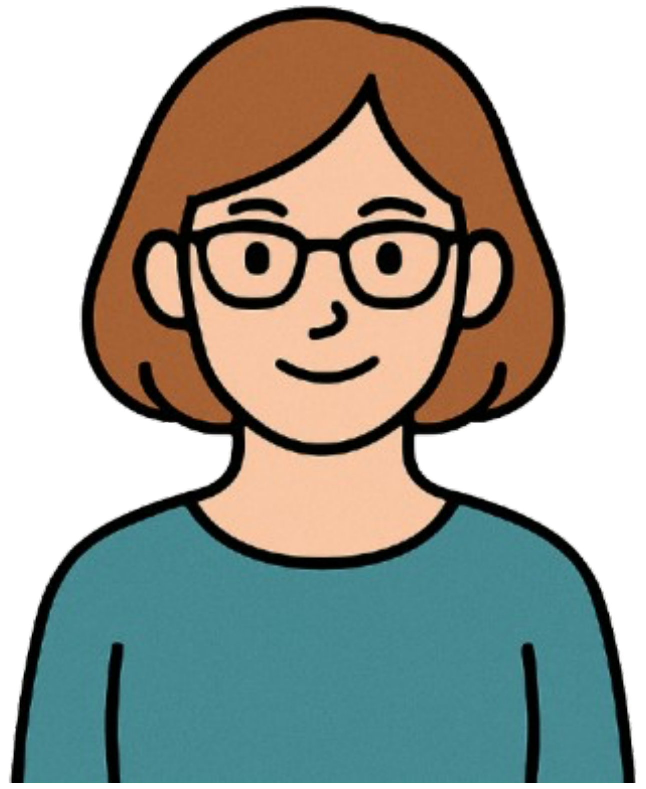
Alice has collected sensitive data $\mathbf{x}$
in a place with *limited resources*

$C(\mathbf{x}) = ???$

Application: Outsourcing computation

[AIKW13]

ALICE

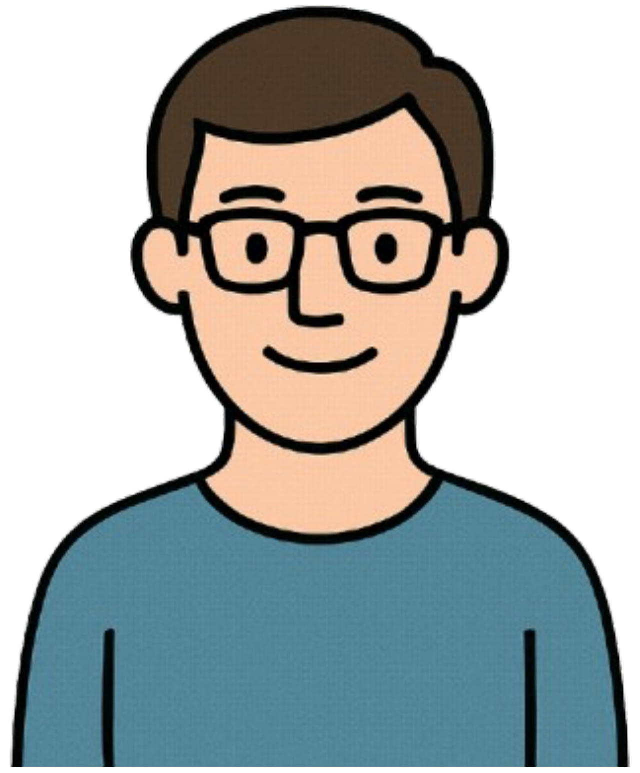


Offline (known C)

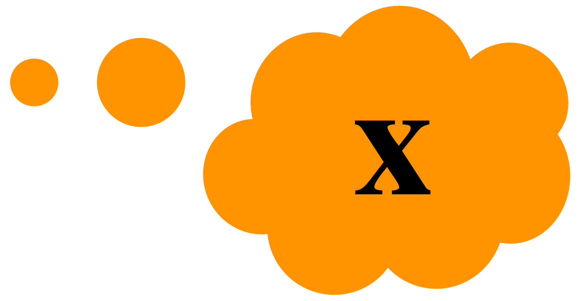
$\mathsf{Garble}(C) \to \hat{C}, \mathbf{K}$

$$\xrightarrow{\hat{C}}$$

BOB



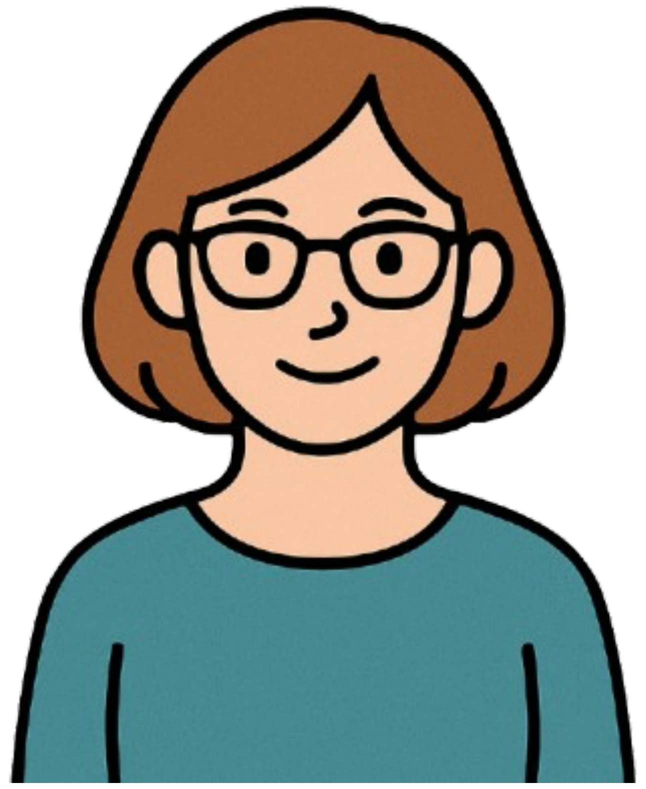
Online



Application: Outsourcing computation

[AIKW13]

ALICE

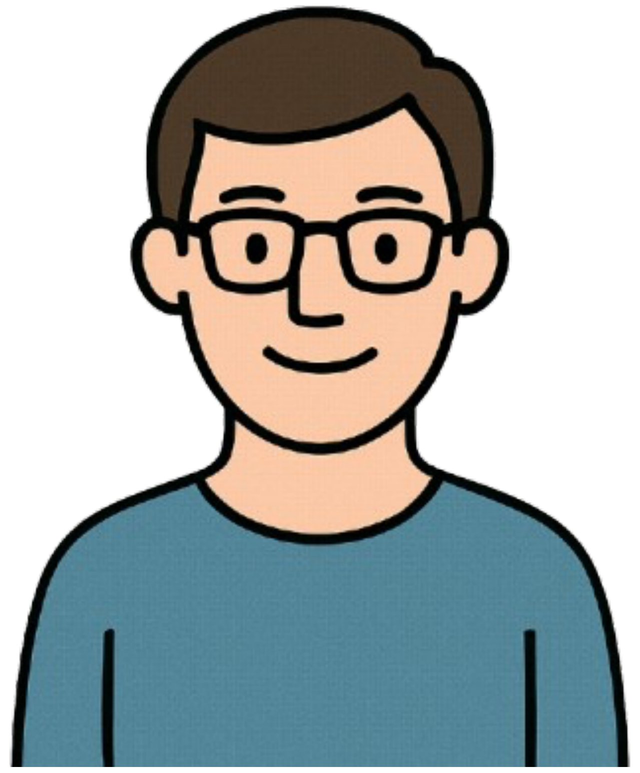


Offline (known C)

Garble(C) $\rightarrow \hat{C}, \mathbf{K}$

$\xrightarrow{\hat{C}}$

BOB



Online

$\mathbf{x}$

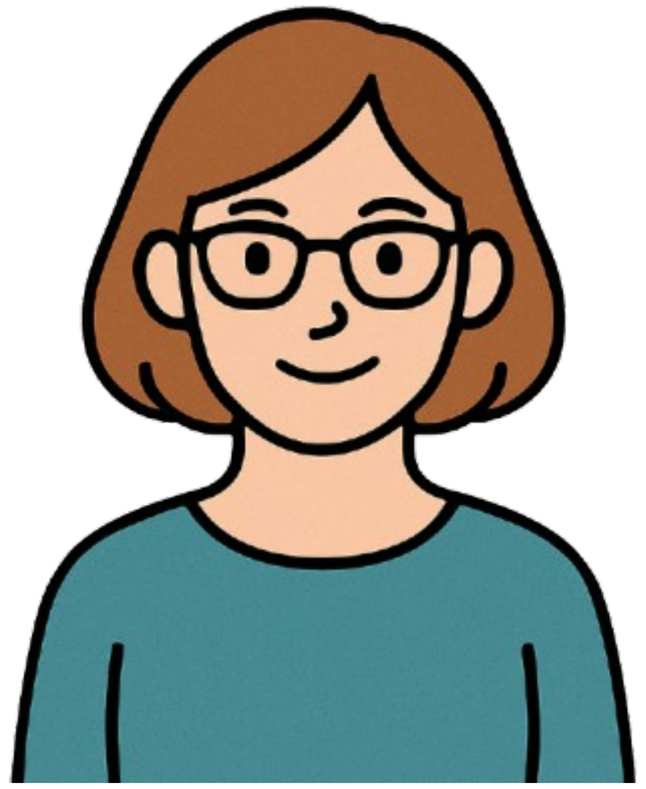
$\xrightarrow{\mathbf{L} = \mathbf{K}[\mathbf{x}]}$

Eval$(\hat{C}, \mathbf{L})$ $\rightarrow C(\mathbf{x})$

Application: Outsourcing computation

[AIKW13]

ALICE

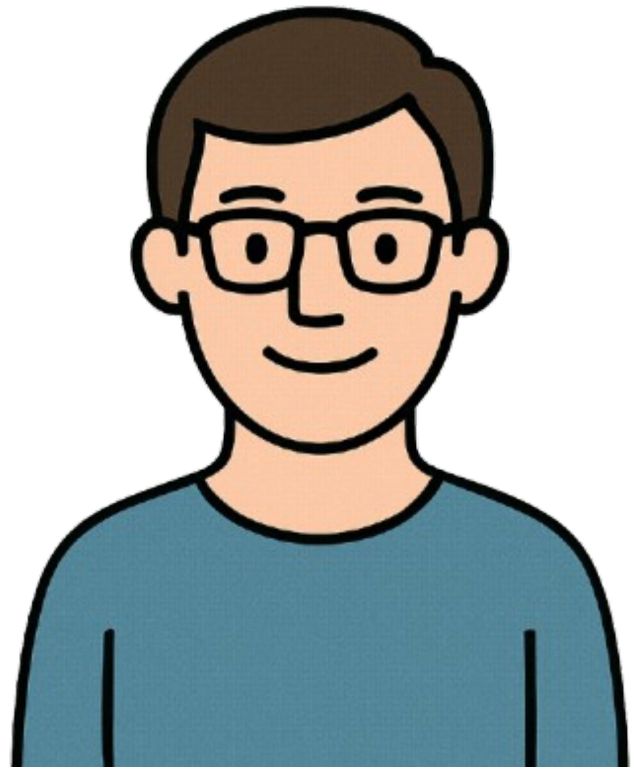


Offline (known C)

$\mathsf{Garble}(C) \to \hat{C}, \mathbf{K}$

$\xrightarrow{\hat{C}}$

BOB



Online

$\mathbf{x}$

$\xrightarrow{\mathbf{L} = \mathbf{K}[\mathbf{x}]}$

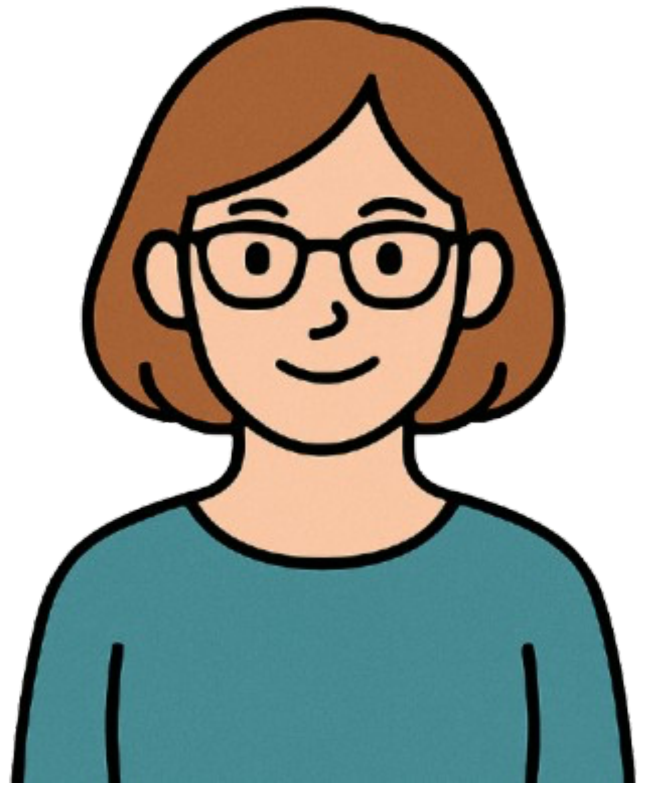
Size: $|\mathbf{x}| \cdot \lambda$

$\mathsf{Eval}(\hat{C}, \mathbf{L}) \to C(\mathbf{x})$

Application: Outsourcing computation

[AIKW13]

ALICE



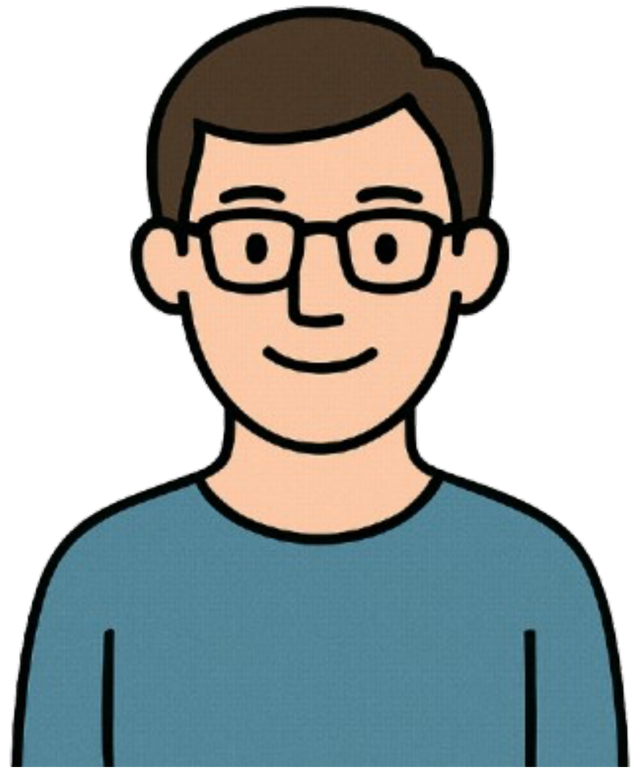
Offline (known C)

Garble(C)

$\rightarrow \hat{C}, \mathbf{K}$

$\xrightarrow{\hat{C}}$

BOB



Online

Compress$(\mathbf{K}[\mathbf{x}]), \bar{\mathbf{x}}$ $\longrightarrow$

Decompress $\rightarrow \mathbf{L}$

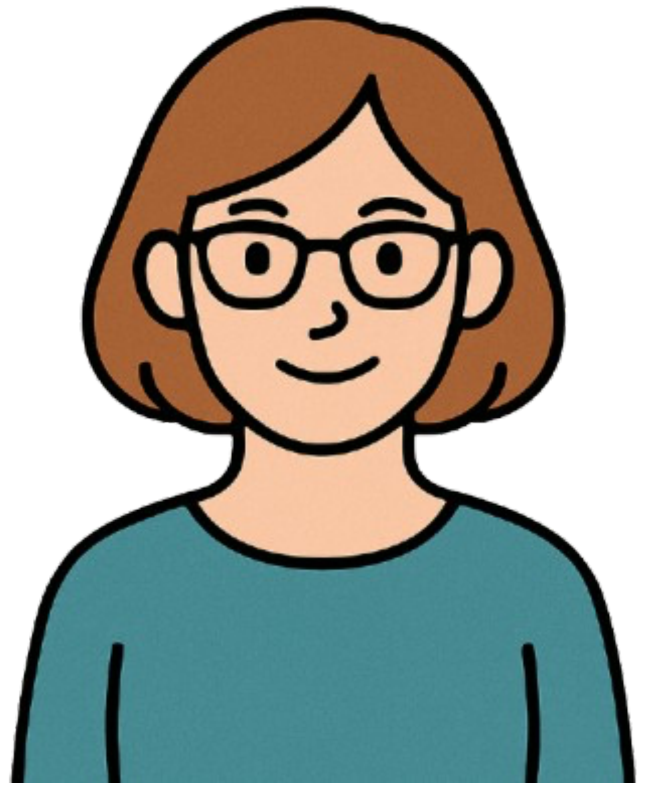
Eval$(\hat{C}, \mathbf{L})$

$\rightarrow C(\mathbf{x})$

Application: Outsourcing computation

[AIKW13]

ALICE



Offline (known C)

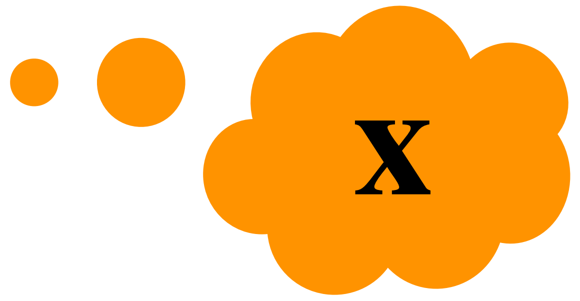
Garble(C)

$\rightarrow \hat{C}, \mathbf{K}$

$\hat{C}$

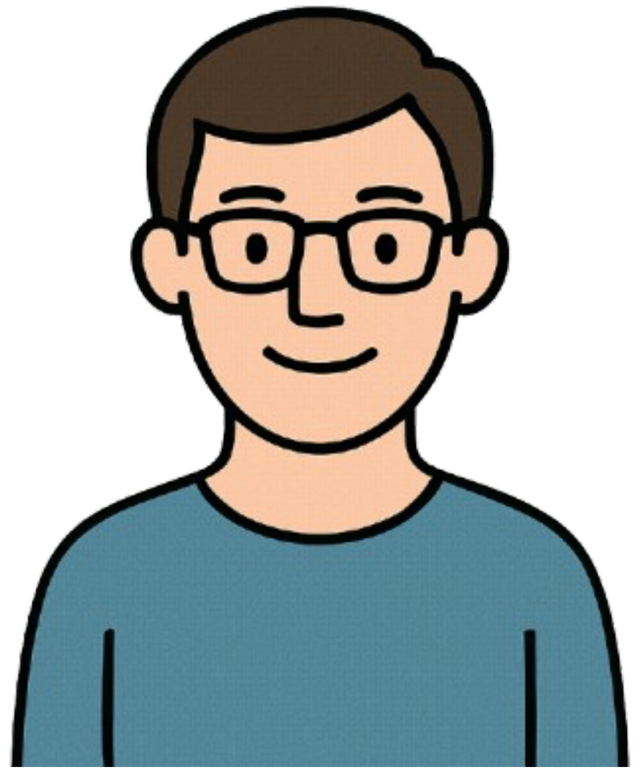
$CT(\mathbf{K})$

Online



Compress$(\mathbf{K}[\mathbf{x}]), \bar{\mathbf{x}}$

BOB



Decompress $\rightarrow \mathbf{L}$

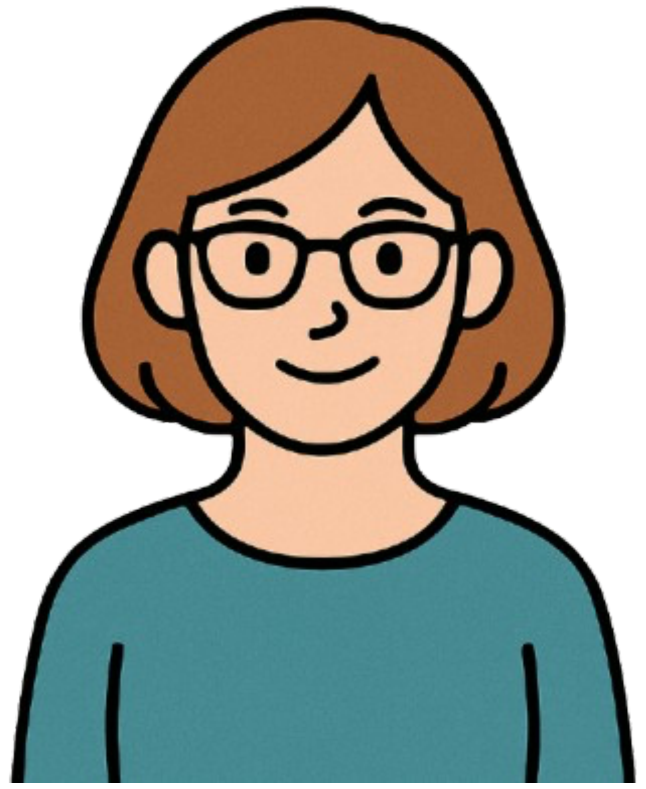
Eval$(\hat{C}, \mathbf{L})$

$\rightarrow C(\mathbf{x})$

Application: Outsourcing computation

[AIKW13]

ALICE



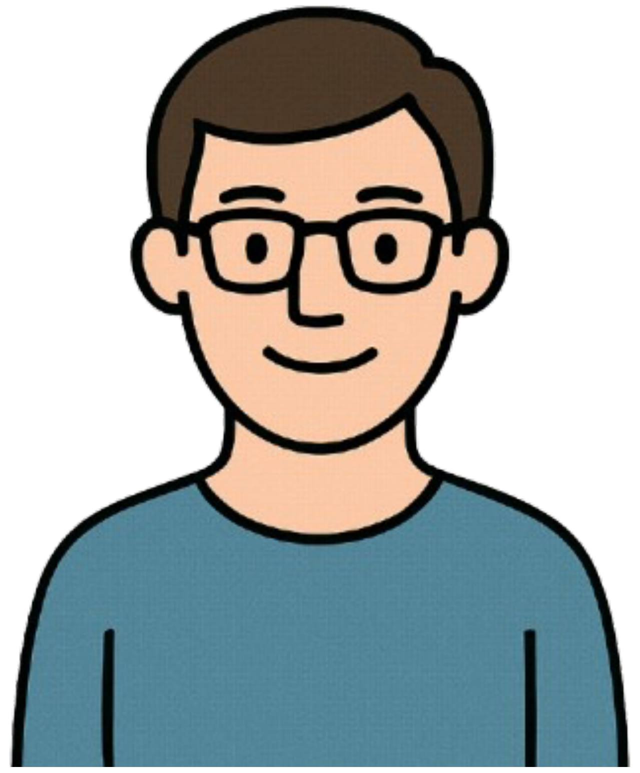
Offline (known C)

Garble(C)
$\rightarrow \hat{C}, \mathbf{K}$

$\hat{C}$

$CT(\mathbf{K})$

BOB



Online

Compress$(\mathbf{K}[\mathbf{x}]), \bar{\mathbf{x}}$

Size: $\mathrm{poly}(\lambda) + |\mathbf{x}|$

Decompress $\rightarrow \mathbf{L}$

Eval$(\hat{C}, \mathbf{L})$
$\rightarrow C(\mathbf{x})$

Existing Techniques

	Assumption	Communication Offline	Online
Naive		0	$\lvert \mathbf{x} \rvert \cdot \lambda$
[AIKW13]	RSA	$\lvert \mathbf{x} \rvert \cdot \mathrm{poly}(\lambda)$	$\lvert \mathbf{x} \rvert + \log N$
	DDH/LWE	"	$\lvert \mathbf{x} \rvert + \lvert \mathbf{x} \rvert / \lambda$

N = RSA modulus

Existing Techniques

	Assumption	Communication Offline	Communication Online
Naive		0	$\lvert\mathbf{x}\rvert \cdot \lambda$
[AIKW13]	RSA	$\lvert\mathbf{x}\rvert \cdot \mathrm{poly}(\lambda)$	$\lvert\mathbf{x}\rvert + \log N$
	DDH/LWE	"	$\lvert\mathbf{x}\rvert + \lvert\mathbf{x}\rvert/\lambda$

[GS18,GOS18]: **weaker assumptions** (factoring / CDH), but **non-black-box** cryptography

N = RSA modulus

Existing Techniques

	Assumption	Communication Offline	Communication Online
Naive		0	$\lvert \mathbf{x} \rvert \cdot \lambda$
[AIKW13]	RSA	$\lvert \mathbf{x} \rvert \cdot \mathrm{poly}(\lambda)$	$\lvert \mathbf{x} \rvert + \log N$
	DDH/LWE	"	$\lvert \mathbf{x} \rvert + \lvert \mathbf{x} \rvert / \lambda$
[ABIKLV23]	RSA/iO+SSBH	0	$\lvert \mathbf{x} \rvert + \log N$
	Bilinear DDH	0*	$\lvert \mathbf{x} \rvert + \lvert \mathbf{x} \rvert / \lambda$

N = RSA modulus

Existing Techniques

	Assumption	Communication Offline	Communication Online	Computation (Decompress)
Naive		0	$\lvert \mathbf{x} \rvert \cdot \lambda$	0
[AIKW13]	RSA DDH/LWE	$\lvert \mathbf{x} \rvert \cdot \mathrm{poly}(\lambda)$	$\lvert \mathbf{x} \rvert + \log N$	$\lvert \mathbf{x} \rvert \log \lvert \mathbf{x} \rvert$ RSA exponentiations
[ABIKLV23]	RSA/iO+SSBH Bilinear DDH	0	$\lvert \mathbf{x} \rvert + \log N$	$\lvert \mathbf{x} \rvert \log \lvert \mathbf{x} \rvert$ RSA exponentiations

N = RSA modulus

Existing Techniques

	Assumption	Communication Offline	Communication Online	Computation (Decompress)
Naive		0	$\lvert\mathbf{x}\rvert \cdot \lambda$	0
[AIKW13]	RSA DDH/LWE	$\lvert\mathbf{x}\rvert \cdot \mathrm{poly}(\lambda)$	$\lvert\mathbf{x}\rvert + \log N$	$\lvert\mathbf{x}\rvert \log \lvert\mathbf{x}\rvert$ RSA exponentiations 30 min
[ABIKLV23]	RSA/iO+SSBH Bilinear DDH	0	$\lvert\mathbf{x}\rvert + \log N$	$\lvert\mathbf{x}\rvert \log \lvert\mathbf{x}\rvert$ RSA exponentiations

(for $\lvert\mathbf{x}\rvert = 10^5$)

N = RSA modulus

Existing Techniques

* assuming 1Mbps

	Assumption	Communication		Computation
		Offline	Online	(Decompress)
Naive		0	13 sec* $\vert \mathbf{x} \vert$	0
[AIKW13]	RSA DDH/LWE	$\vert \mathbf{x} \vert \cdot \mathrm{poly}(\lambda)$	$\vert \mathbf{x} \vert + \log N$	$\vert \mathbf{x} \vert \log \vert \mathbf{x} \vert$ RSA exponentiations 30 min
[ABIKLV23]	RSA/iO+SSBH Bilinear DDH	0	$\vert \mathbf{x} \vert + \log N$	$\vert \mathbf{x} \vert \log \vert \mathbf{x} \vert$ RSA exponentiations

(for $|\mathbf{x}| = 10^5$)

N = RSA modulus

Our work — TinyLabels

	Assumption	Communication Offline	Communication Online	Computation (Decompress)
Naive		0	$\|\mathbf{x}\| \cdot \lambda$	0
[AIKW13]	RSA DDH/LWE	$\|\mathbf{x}\| \cdot \mathrm{poly}(\lambda)$	$\|\mathbf{x}\| + \log N$	$\|\mathbf{x}\| \log \|\mathbf{x}\|$ RSA exponentiations
[ABIKLV23]	RSA/iO+SSBH Bilinear DDH	0	$\|\mathbf{x}\| + \log N$	$\|\mathbf{x}\| \log \|\mathbf{x}\|$ RSA exponentiations
TinyLabels	RingLWE+RO	0*	$\|\mathbf{x}\| + \|\mathscr{R}_q\|$	$\|\mathbf{x}\| \log \|\mathbf{x}\| / n$ $\mathscr{R}_q$ multiplications

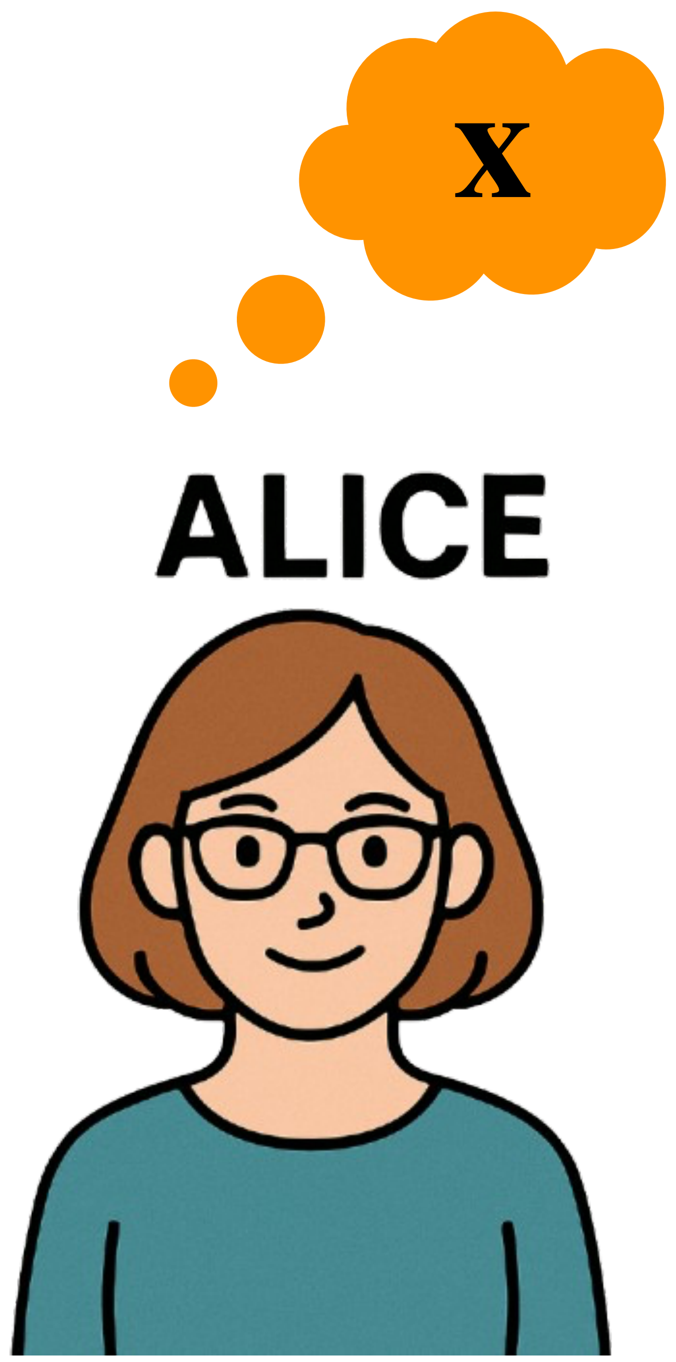
N = RSA modulus $\quad\quad$ $\mathscr{R}_q$ = ring with degree n and modulus q

Our work — TinyLabels

	Assumption	Communication: Offline	Communication: Online	Computation (Decompress)
Naive		0	$\lvert \mathbf{x} \rvert$ 13 sec*	0
[AIKW13]	RSA DDH/LWE	$\lvert \mathbf{x} \rvert \cdot \mathrm{poly}(\lambda)$	$\lvert \mathbf{x} \rvert + \log N$	$\lvert \mathbf{x} \rvert \log \lvert \mathbf{x} \rvert$ RSA exponentiations 30 min
[ABIKLV23]	RSA/iO+SSBH Bilinear DDH	0	$\lvert \mathbf{x} \rvert + \log N$	$\lvert \mathbf{x} \rvert \log \lvert \mathbf{x} \rvert$ RSA exponentiations
TinyLabels	RingLWE+RO	0*	$\lvert \mathbf{x} \rvert + \lvert \mathscr{R}_q \rvert$	$\lvert \mathbf{x} \rvert \log \lvert \mathbf{x} \rvert / n$ $\mathscr{R}_q$ multiplications 0.2 sec

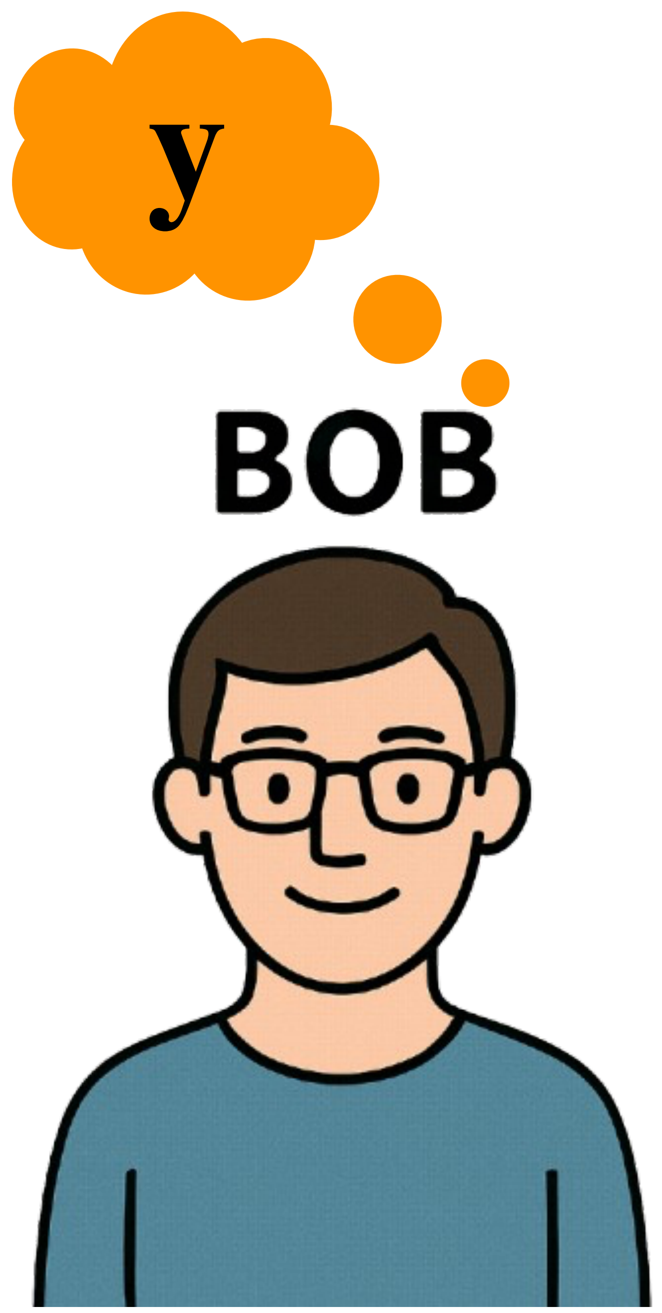
N = RSA modulus $\mathscr{R}_q$ = ring with degree n and modulus q

Application: Garbling with preprocessing

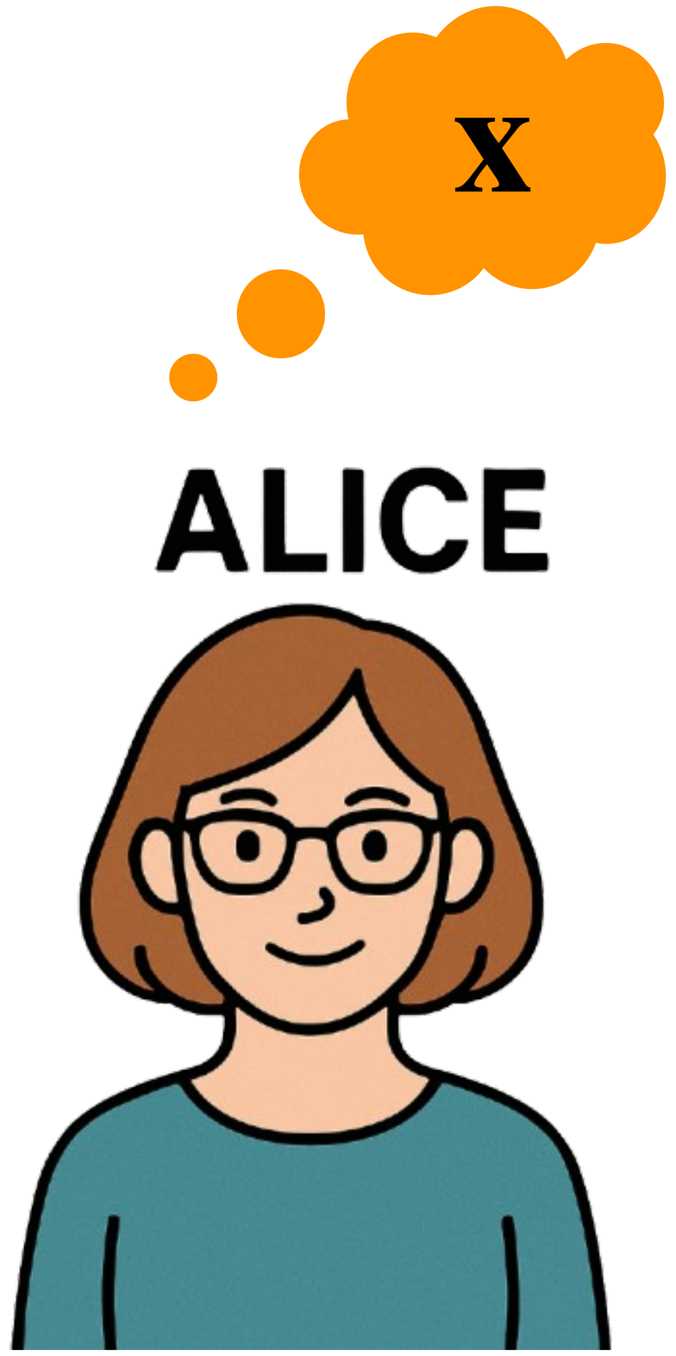


Let’s analyze our joint data securely!

Okay, but using which circuit C?

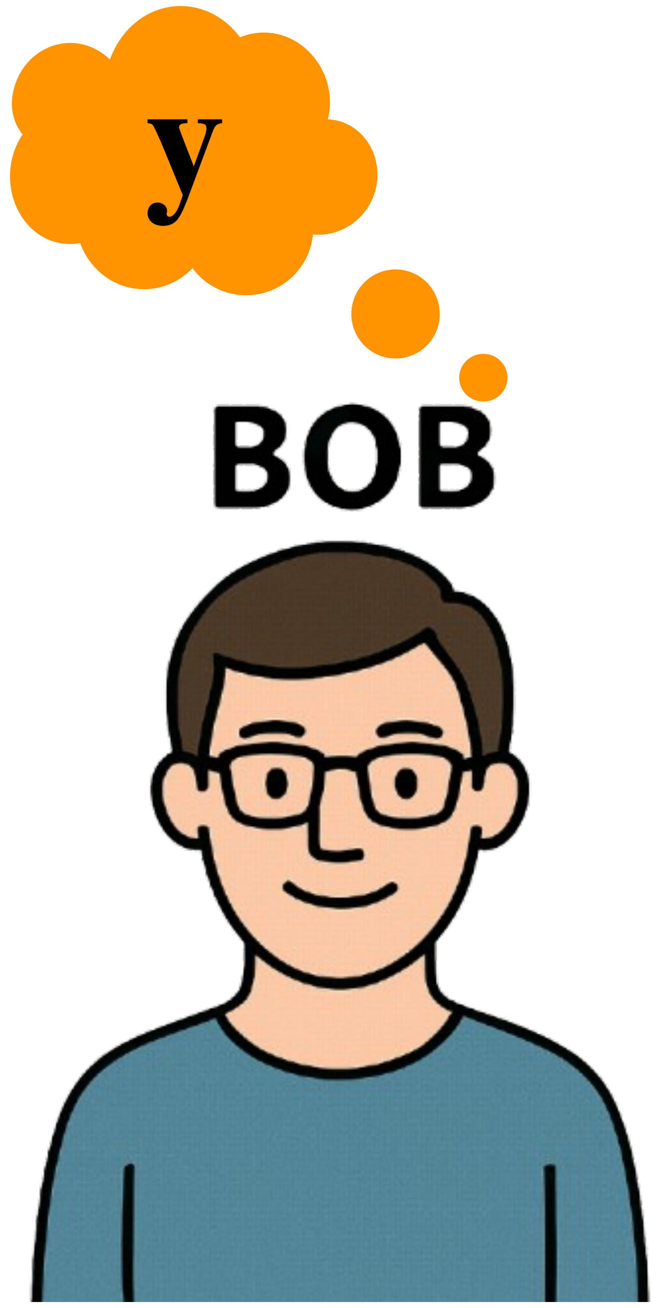


Application: Garbling with preprocessing



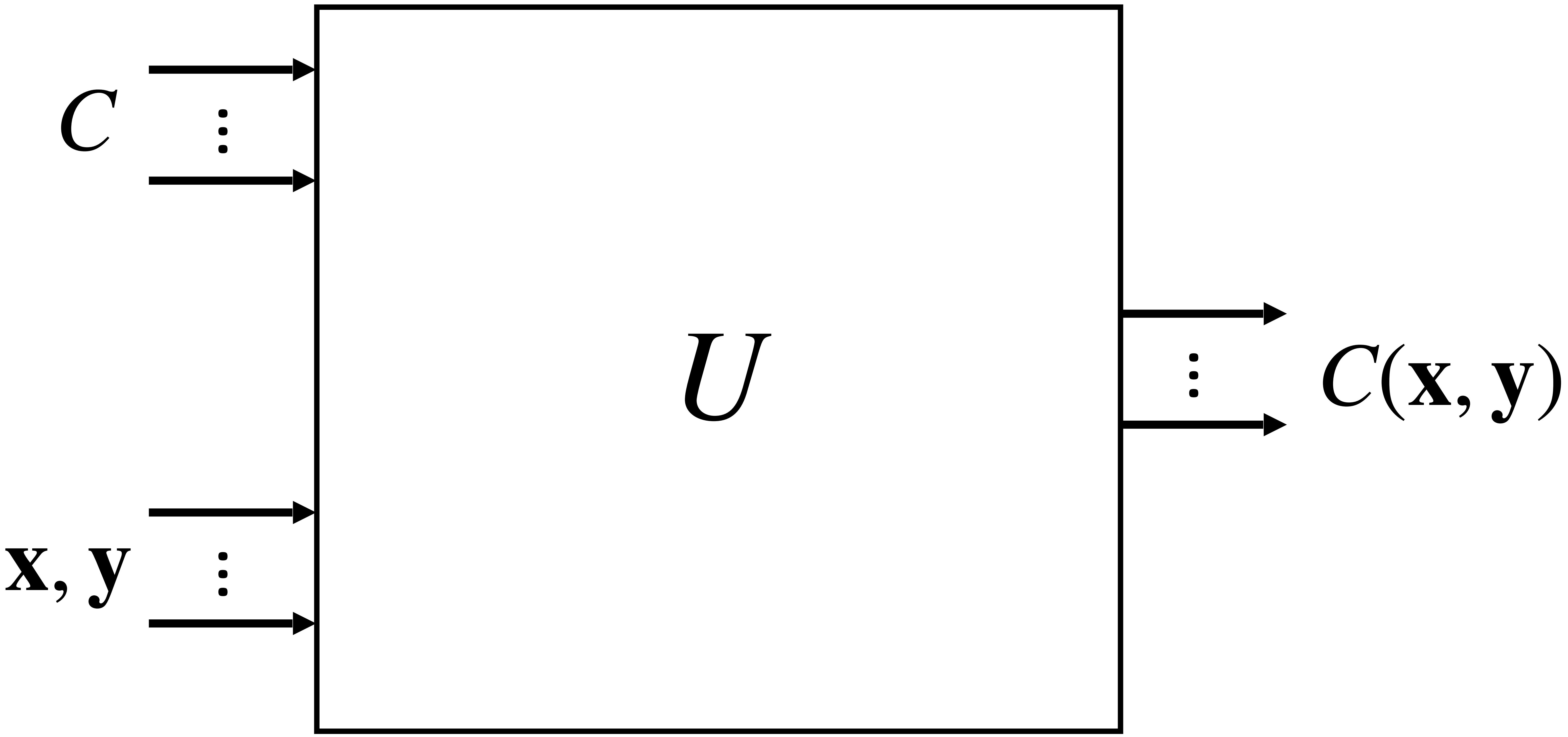
Let's analyze our joint data securely!

Okay, but using which circuit C?



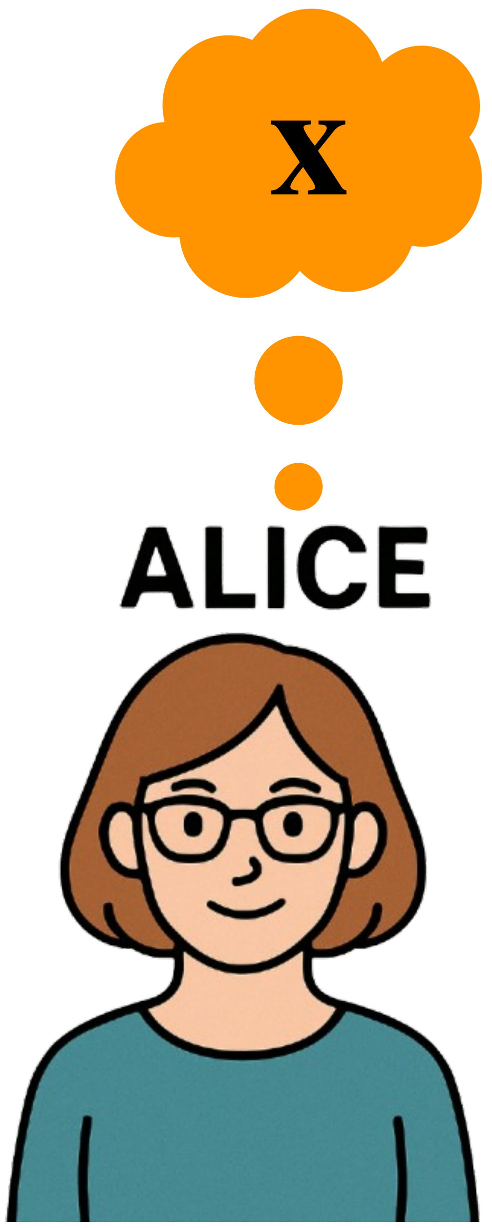
Let's discuss. In the meantime, we should preprocess with TinyLabels.

Universal Circuits



[ZYZL19]: $|U| \approx 4.5\,|C| \log |C|$ AND gates

Application: Garbling with preprocessing

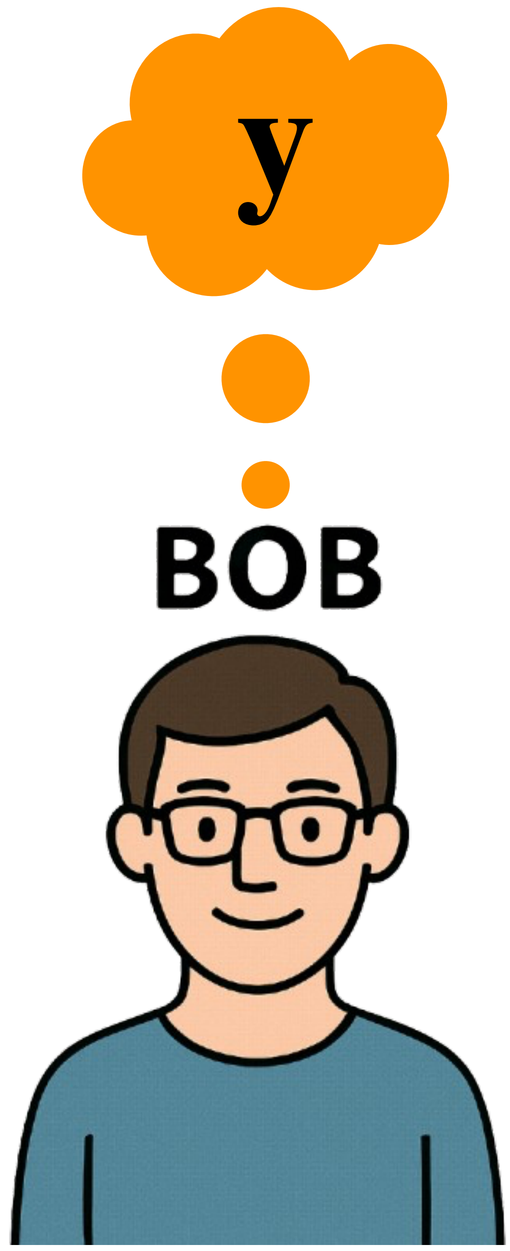


Garble(C)

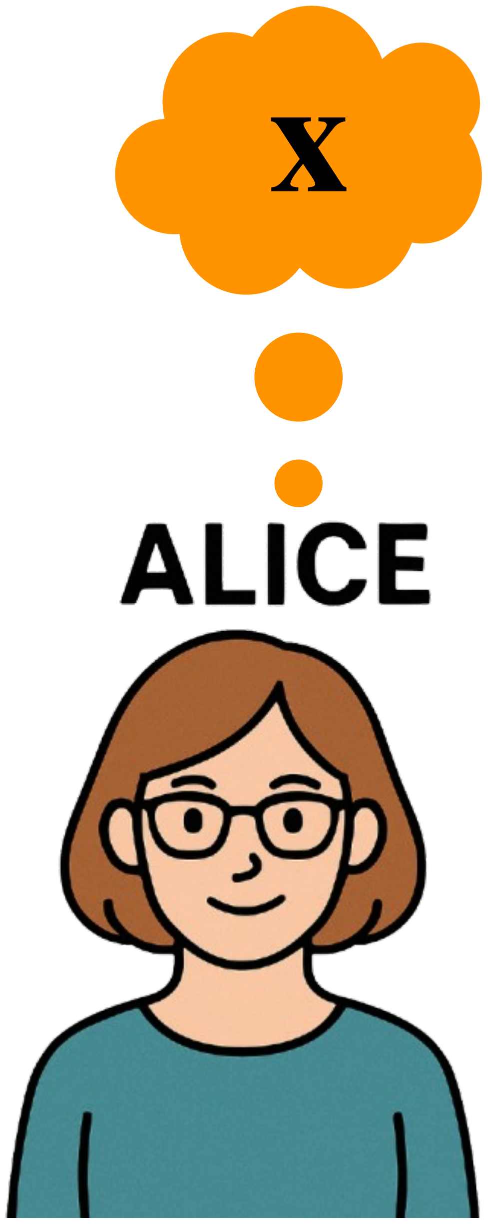
$\rightarrow \hat{C}, \mathbf{K}$

$\xrightarrow{\hat{C}}$

Size: $O(|C| \cdot \lambda)$



Application: Garbling with preprocessing

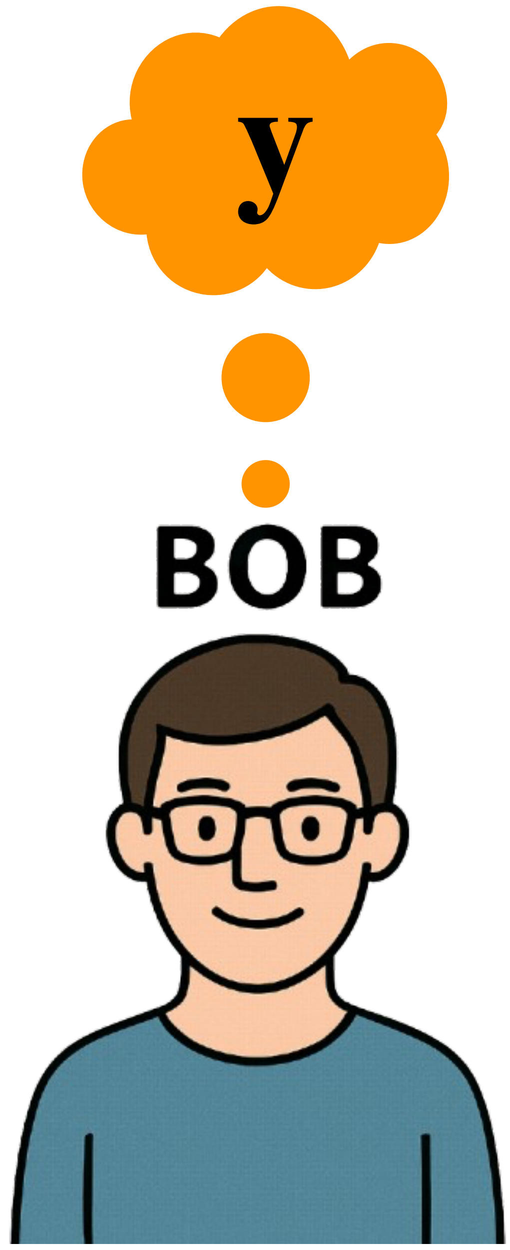


$\mathsf{Garble}(C)$

$\rightarrow \hat{C}, \mathbf{K}$

$\xrightarrow{\hat{C}}$

Size: $O(|C| \cdot \lambda)$



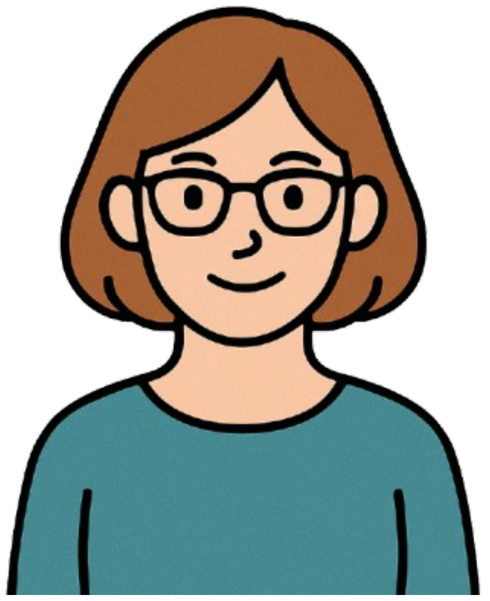
Transferring $\mathbf{K}^x[\mathbf{x}], \mathbf{K}^y[\mathbf{y}]$

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

Application: Garbling with preprocessing

$\mathbf{x}$

ALICE



Garble(C)

$\rightarrow \hat{C}, \mathbf{K}$

$\hat{C}$

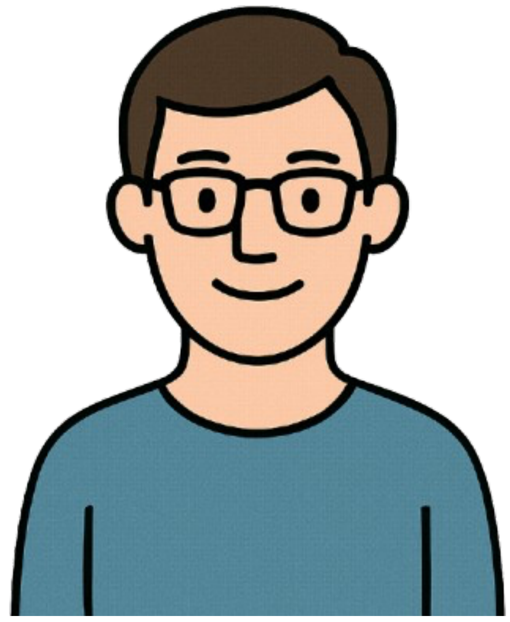
Size: $O(|C| \cdot \lambda)$

Transferring $\mathbf{K}^x[\mathbf{x}], \mathbf{K}^y[\mathbf{y}]$

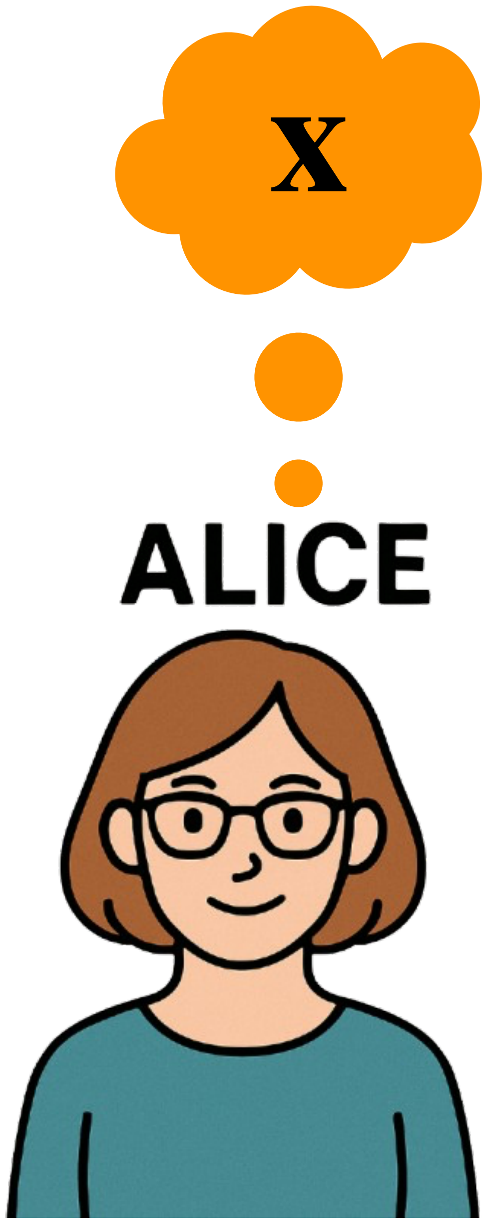
Size: $|\mathbf{x}| + |\mathbf{y}| \cdot \mathrm{poly}(\lambda)$

$\mathbf{y}$

BOB



Application: Garbling with preprocessing



Offline

Garble(U)

$\rightarrow \hat{U}, (\mathbf{K}^C, \mathbf{K}^x, \mathbf{K}^y)$

$\xrightarrow{\hat{U}}$

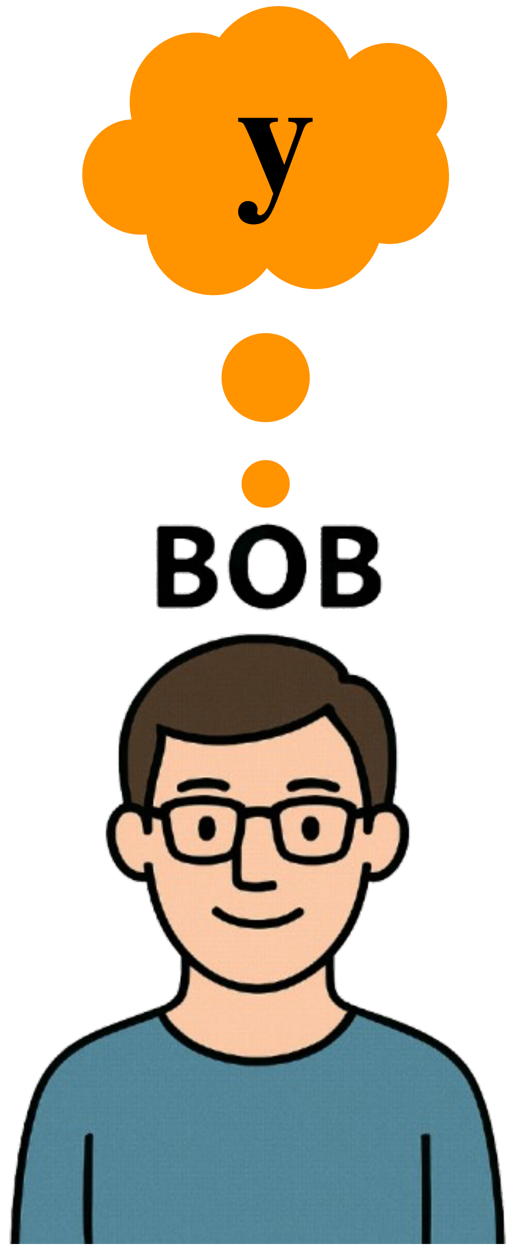
Online (C known)

$\xrightarrow{\text{Compress}(\mathbf{K}^C[C])}$

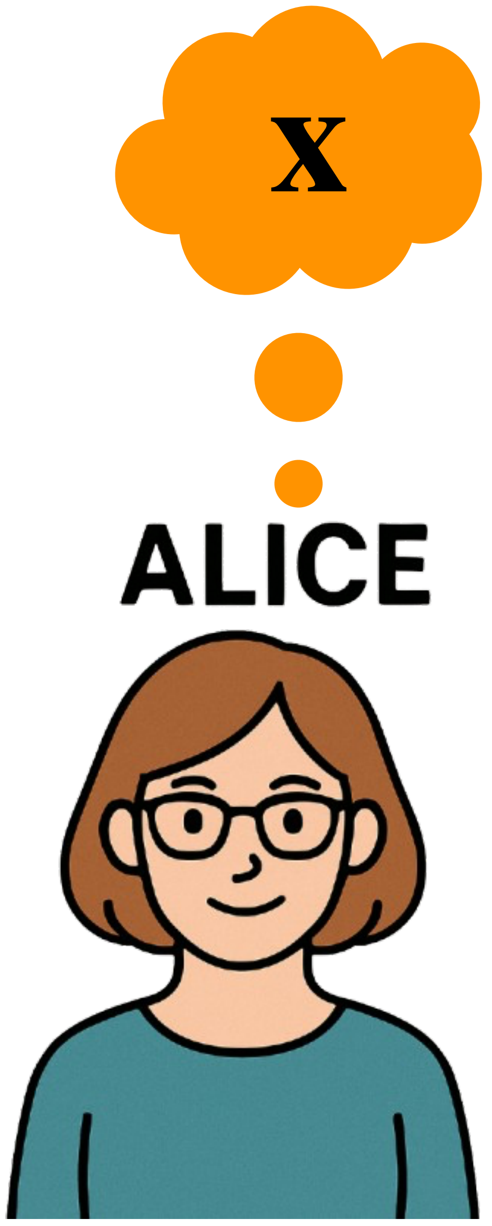
Decompress $\rightarrow \mathbf{L}^C$

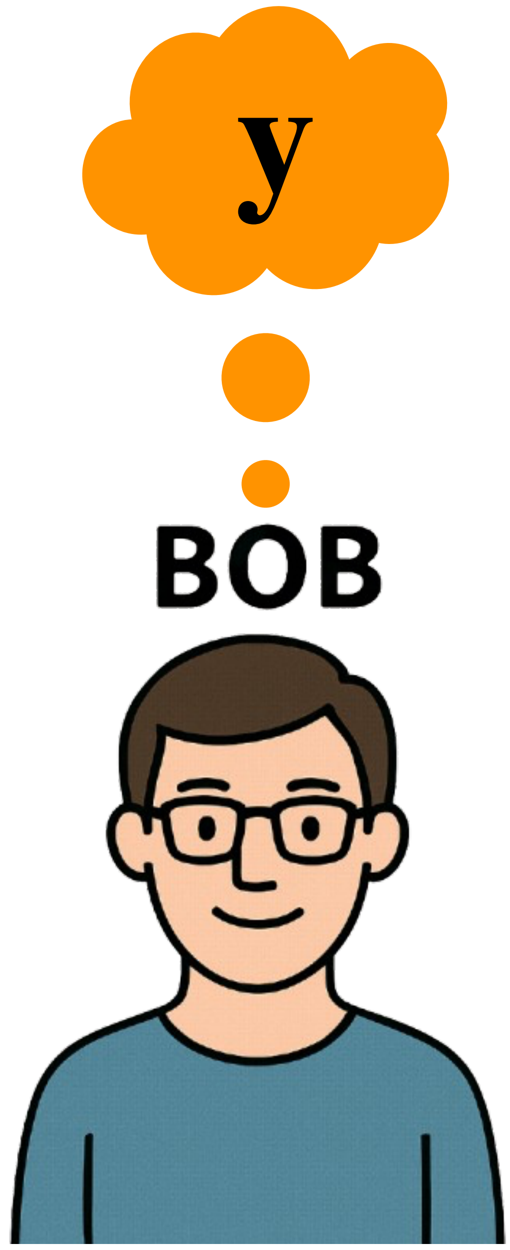
Transferring $\mathbf{K}^x[\mathbf{x}], \mathbf{K}^y[\mathbf{y}]$

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



Application: Garbling with preprocessing





Offline

Garble(U)

$\to \hat{U}, (\mathbf{K}^C, \mathbf{K}^x, \mathbf{K}^y)$

$\xrightarrow{\hat{U}}$

Online (C known)

Size: $\mathrm{poly}(\lambda)$

Compress$(\mathbf{K}^C[C])$ $\longrightarrow$

Decompress $\to \mathbf{L}^C$

Transferring $\mathbf{K}^x[\mathbf{x}], \mathbf{K}^y[\mathbf{y}]$

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

Main Tool: Batch-Select

<u>Sender</u>

Message vectors

$\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{m}_1) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{m}_2) \to \mathrm{st}_2, \mathrm{ct}_2$

Main Tool: Batch-Select

Sender

Message vectors

$\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}_p^w$

$$\mathrm{Enc}_1(\mathbf{m}_1) \to \mathrm{st}_1, \mathrm{ct}_1$$

$$\mathrm{Enc}_2(\mathbf{m}_2) \to \mathrm{st}_2, \mathrm{ct}_2$$

"Selection" vector

$\mathbf{x} \in \mathbb{Z}_p^w$

$$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$$

Main Tool: Batch-Select

Message vectors

$\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}_p^w$

"Selection" vector

$\mathbf{x} \in \mathbb{Z}_p^w$

<u>Sender</u>

$\mathrm{Enc}_1(\mathbf{m}_1) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{m}_2) \to \mathrm{st}_2, \mathrm{ct}_2$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

<u>Receiver</u>

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\to \mathbf{m}_1 \odot \mathbf{x} \ + \ \mathbf{m}_2$

$$= \begin{pmatrix} \mathbf{m}_1[1] \cdot x_1 + \mathbf{m}_2[1] \\ \vdots \\ \mathbf{m}_w[w] \cdot x_w + \mathbf{m}_2[w] \end{pmatrix}$$

Main Tool: Batch-Select

Message vectors

$\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}_p^w$

"Selection" vector

$\mathbf{x} \in \mathbb{Z}_p^w$

<u>Sender</u>

$\mathrm{Enc}_1(\mathbf{m}_1) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{m}_2) \to \mathrm{st}_2, \mathrm{ct}_2$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

<u>Receiver</u>

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$$\to \mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2 = \begin{pmatrix} \mathbf{m}_1[1] \cdot x_1 + \mathbf{m}_2[1] \\ \vdots \\ \mathbf{m}_w[w] \cdot x_w + \mathbf{m}_2[w] \end{pmatrix}$$

Succinctness: $|\mathrm{sk}_{\mathbf{x}}| = \mathrm{poly}(\lambda)$

Main Tool: Batch-Select

Sender | **Receiver**

Message vectors

$\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{m}_1) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{m}_2) \to \mathrm{st}_2, \mathrm{ct}_2$

$$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$$

$$\to \mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2$$

$$= \begin{pmatrix} \mathbf{m}_1[1] \cdot x_1 + \mathbf{m}_2[1] \\ \vdots \\ \mathbf{m}_w[w] \cdot x_w + \mathbf{m}_2[w] \end{pmatrix}$$

"Selection" vector

$\mathbf{x} \in \mathbb{Z}_p^w$

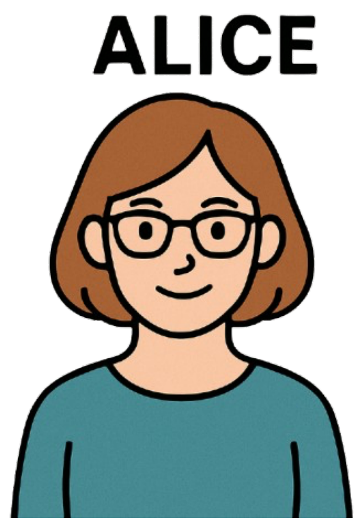
$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

Simulation Security:

For all $\mathbf{m}_1, \mathbf{m}_2, \mathbf{x}$:

$$(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2) \approx \mathrm{Sim}(\mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2, \mathbf{x})$$

Compressing Labels using Batch-Select



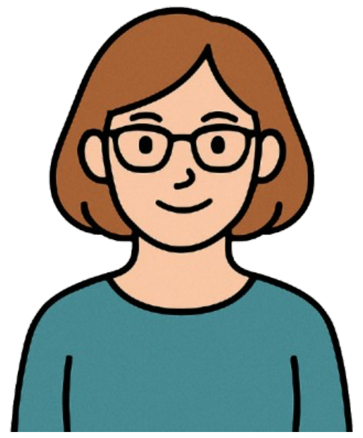
GC input keys

$\mathbf{K[0]}, \mathbf{K[1]} \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K[1]} - \mathbf{K[0]}) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K[0]}) \rightarrow \mathrm{st}_2, \mathrm{ct}_2$

Compressing Labels using Batch-Select



GC input keys

$\mathbf{K}[\mathbf{0}], \mathbf{K}[\mathbf{1}] \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K}[\mathbf{1}] - \mathbf{K}[\mathbf{0}]) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[\mathbf{0}]) \rightarrow \mathrm{st}_2, \mathrm{ct}_2$

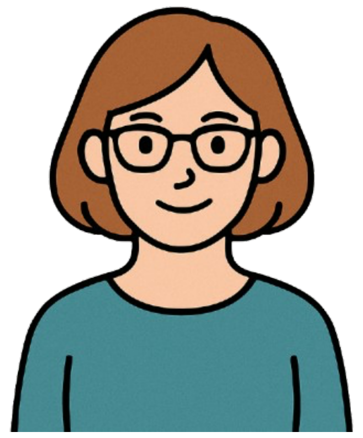
Input

$\mathbf{x} \in \{0,1\}^w$

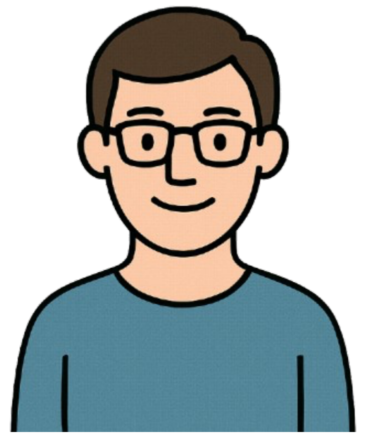
$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \rightarrow \mathrm{sk}_{\mathbf{x}}$

Compressing Labels using Batch-Select

ALICE



BOB



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

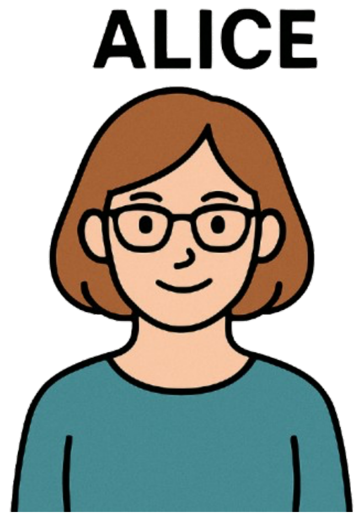
$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$$\to (\mathbf{K}[1] - \mathbf{K}[0]) \odot \mathbf{x} + \mathbf{K}[0]$$

$$= \begin{pmatrix} \mathbf{K}[x_1] \\ \vdots \\ \mathbf{K}[x_w] \end{pmatrix} = \mathbf{K}[\mathbf{x}]$$

Compressing Labels using Batch-Select



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

$\mathsf{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \rightarrow \mathsf{st}_1, \mathsf{ct}_1$

$\mathsf{Enc}_2(\mathbf{K}[0]) \rightarrow \mathsf{st}_2, \mathsf{ct}_2$

offline communication

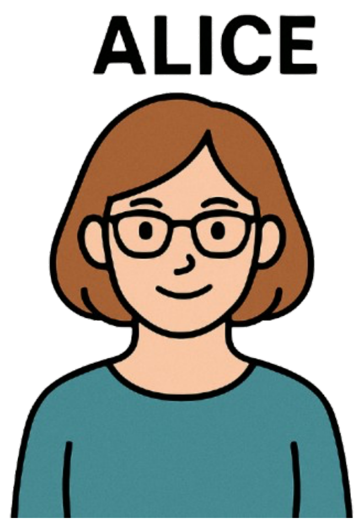
Input

$\mathbf{x} \in \{0,1\}^w$

$\mathsf{KeyGen}(\mathsf{st}_1, \mathsf{st}_2, \mathbf{x}) \rightarrow \mathsf{sk}_{\mathbf{x}}$

online communication

Compressing Labels using Batch-Select



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \rightarrow \mathrm{st}_2, \mathrm{ct}_2$

offline communication

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \rightarrow \mathrm{sk}_{\mathbf{x}}$

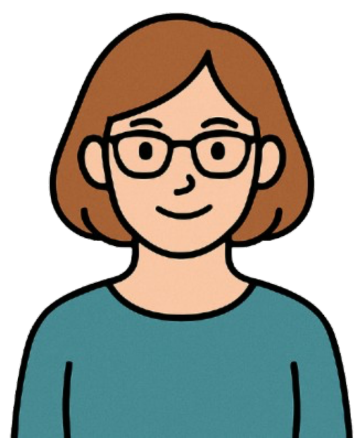
$|\mathcal{R}_q|$

online communication

(Assuming our construction from RingLWE)

Compressing Labels using Batch-Select

ALICE



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

offline communication

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

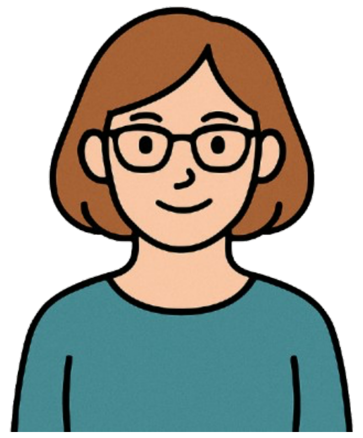
$|\mathscr{R}_q|$

($+|\mathbf{x}|$ if input is secret)

online communication

(Assuming our construction from RingLWE)

Compressing Labels using Batch-Select



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

$O(w \log w \cdot |\mathbb{Z}_p|)$

offline communication

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

$|\mathcal{R}_q|$

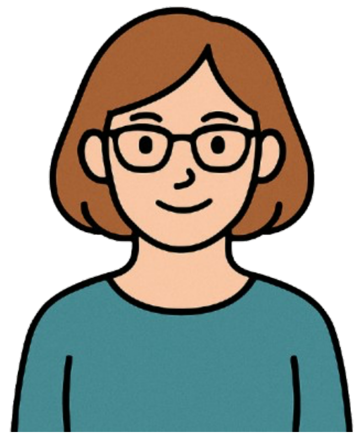
(+ $|\mathbf{x}|$ if input is secret)

online communication

(Assuming our construction from RingLWE)

Compressing Labels using Batch-Select

ALICE



GC input keys

$\mathbf{K}[0], \mathbf{K}[1] \in \mathbb{Z}_p^w$

$\mathrm{Enc}_1(\mathbf{K}[1] - \mathbf{K}[0]) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

Can be optimized to be essentially free!

offline communication

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

$|\mathscr{R}_q|$

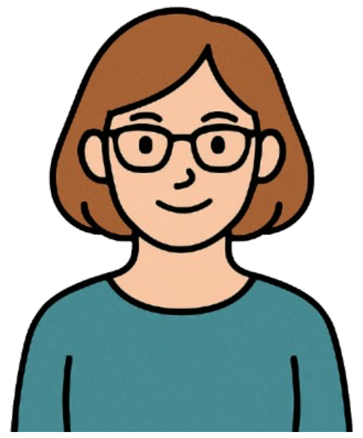
(+ $|\mathbf{x}|$ if input is secret)

online communication

(Assuming our construction from RingLWE)

Optimization 1: Free-XOR keys

ALICE



GC input keys

$\mathbf{K}[0]$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

$\mathrm{Enc}_1(\mathbf{K}[\mathbf{1}] - \mathbf{K}[\mathbf{0}]) \to \mathrm{st}_1, \mathrm{ct}_1$

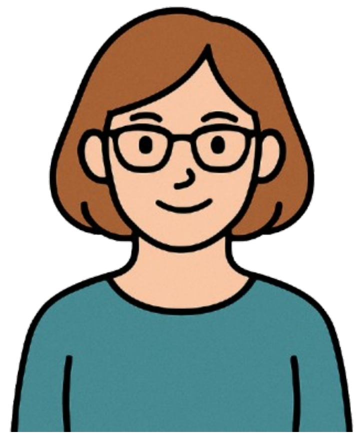
$\mathrm{Enc}_2(\mathbf{K}[\mathbf{0}]) \to \mathrm{st}_2, \mathrm{ct}_2$

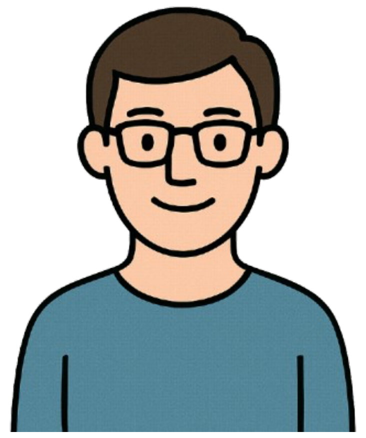
Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

Optimization 1: Free-XOR keys





GC input keys

$\mathbf{K}[0]$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

$\mathrm{Enc}_1(\Delta \cdot \mathbf{1}_w) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\to (\Delta \cdot \mathbf{1}_w) \odot \mathbf{x} \; + \; \mathbf{K}[0]$

$= \mathbf{K}[\mathbf{x}]$

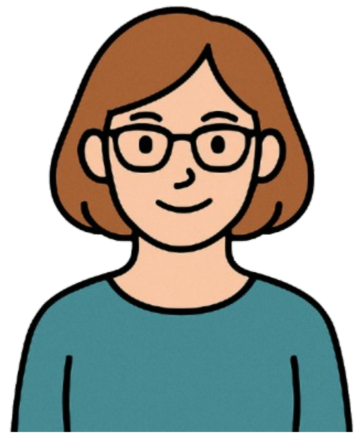
Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

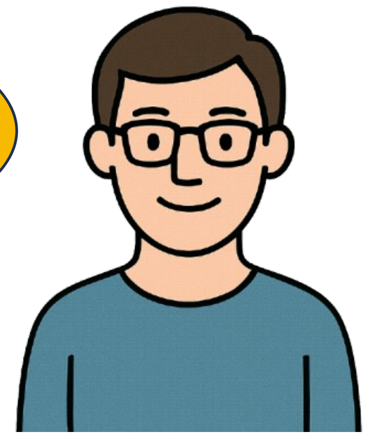
Optimization 1: Free-XOR keys

ALICE



Reusable!

BOB



GC input keys

$\mathbf{K}[0]$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

$\mathrm{Enc}_1(\Delta \cdot \mathbf{1}_w) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \rightarrow \mathrm{st}_2, \mathrm{ct}_2$

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\rightarrow (\Delta \cdot \mathbf{1}_w) \odot \mathbf{x} \; + \; \mathbf{K}[0]$

$= \mathbf{K}[\mathbf{x}]$

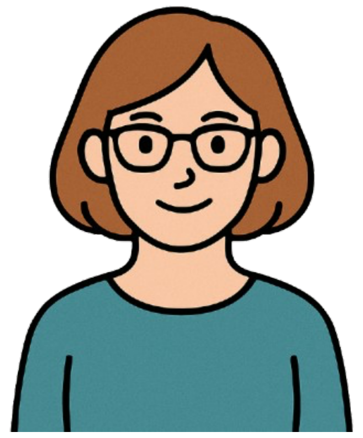
Input

$\mathbf{x} \in \{0,1\}^w$

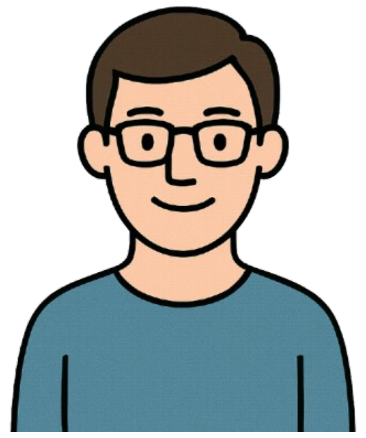
$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \rightarrow \mathrm{sk}_{\mathbf{x}}$

Optimization 2: Free encryption of random keys

ALICE



BOB



GC input keys

$\mathbf{K}[0] \leftarrow \$$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

$\mathrm{Enc}_1(\Delta \cdot \mathbf{1}_w) \to \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{Enc}_2(\mathbf{K}[0]) \to \mathrm{st}_2, \mathrm{ct}_2$

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\to (\Delta \cdot \mathbf{1}_w) \odot \mathbf{x} \; + \; \mathbf{K}[0]$

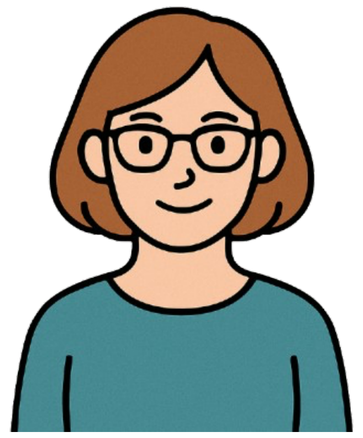
$= \mathbf{K}[\mathbf{x}]$

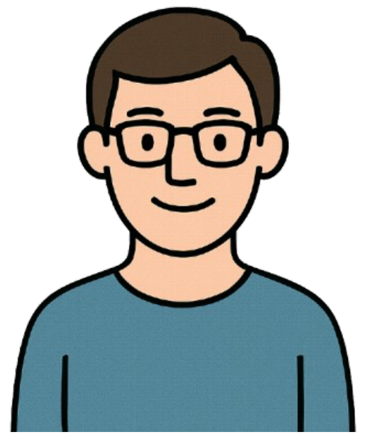
Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \to \mathrm{sk}_{\mathbf{x}}$

Optimization 2: Free encryption of random keys





GC input keys

$\mathbf{K}[0] \leftarrow \$$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

$\mathrm{Enc}_1(\Delta \cdot \mathbf{1}_w) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{ct}_2 \leftarrow \mathrm{RO}(\mathrm{seed})$

$\mathbf{K}[0], \mathrm{st}_2 \leftarrow \mathrm{Dec}(\mathrm{ct}_2)$

$\mathrm{Dec}(\mathrm{sk}_{\mathbf{x}}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\rightarrow (\Delta \cdot \mathbf{1}_w) \odot \mathbf{x} + \mathbf{K}[0]$

$= \mathbf{K}[\mathbf{x}]$

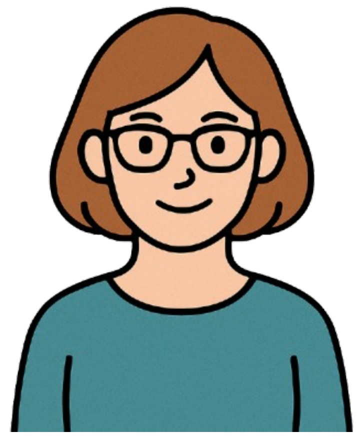
Input

$\mathbf{x} \in \{0,1\}^w$

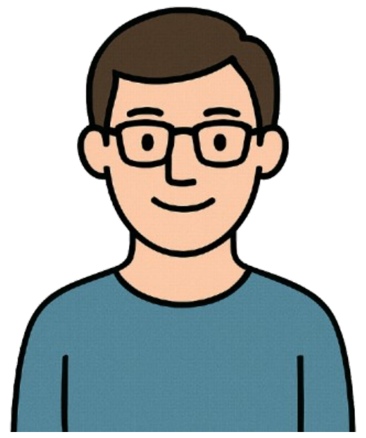
$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \rightarrow \mathrm{sk}_{\mathbf{x}}$

Optimization 2: Free encryption of random keys

ALICE



BOB



GC input keys

$\mathbf{K}[0] \leftarrow \$$

$\mathbf{K}[1] = \mathbf{K}[0] + \Delta \cdot \mathbf{1}_w$

Input

$\mathbf{x} \in \{0,1\}^w$

$\mathrm{Enc}_1(\Delta \cdot \mathbf{1}_w) \rightarrow \mathrm{st}_1, \mathrm{ct}_1$

$\mathrm{ct}_2 \leftarrow \mathrm{RO}(\mathrm{seed})$

$\mathbf{K}[0], \mathrm{st}_2 \leftarrow \mathrm{Dec}(\mathrm{ct}_2)$

$\mathrm{KeyGen}(\mathrm{st}_1, \mathrm{st}_2, \mathbf{x}) \rightarrow \mathrm{sk}_\mathbf{x}$

Compressible to $|\mathrm{seed}| = \lambda$ bits!

$\mathrm{Dec}(\mathrm{sk}_\mathbf{x}, \mathrm{ct}_1, \mathrm{ct}_2, \mathbf{x})$

$\rightarrow (\Delta \cdot \mathbf{1}_w) \odot \mathbf{x} + \mathbf{K}[0]$

$= \mathbf{K}[\mathbf{x}]$

Constructing Batch-Select from RingLWE

LHE

[AIK11,AIKW13,BLLL23,...]

$$\mathbf{m}_1 \cdot x + \mathbf{m}_2$$

public scalar

succinct

$\mathbf{x}$-dependent key

Batch-Select

$$\mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2$$

public vector

succinct

$\mathbf{x}$-dependent key

Constructing Batch-Select from RingLWE

LHE [AIK11,AIKW13,BLLL23,...]		LEnc [DKLLMR23]		Batch-Select
$\mathbf{m}_1 \cdot x + \mathbf{m}_2$	$+$	$\mathbf{m}_1 \odot \mathbf{x}$	$\Rightarrow$	$\mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2$
public scalar				**public** vector
succinct				**succinct**
$\mathbf{x}$-dependent key				$\mathbf{x}$-dependent key

Constructing Batch-Select from RingLWE

LHE

[AIK11,AIKW13,BLLL23,…]

$$\mathbf{m}_1 \cdot x + \mathbf{m}_2$$

public scalar

succinct

$\mathbf{x}$-dependent key

$+$

LEnc

[DKLLMR23]

$$\mathbf{m}_1 \odot \mathbf{x}$$

non-succinct

$\mathbf{x}$-dependent key

$\Rightarrow$

Batch-Select

$$\mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2$$

public vector

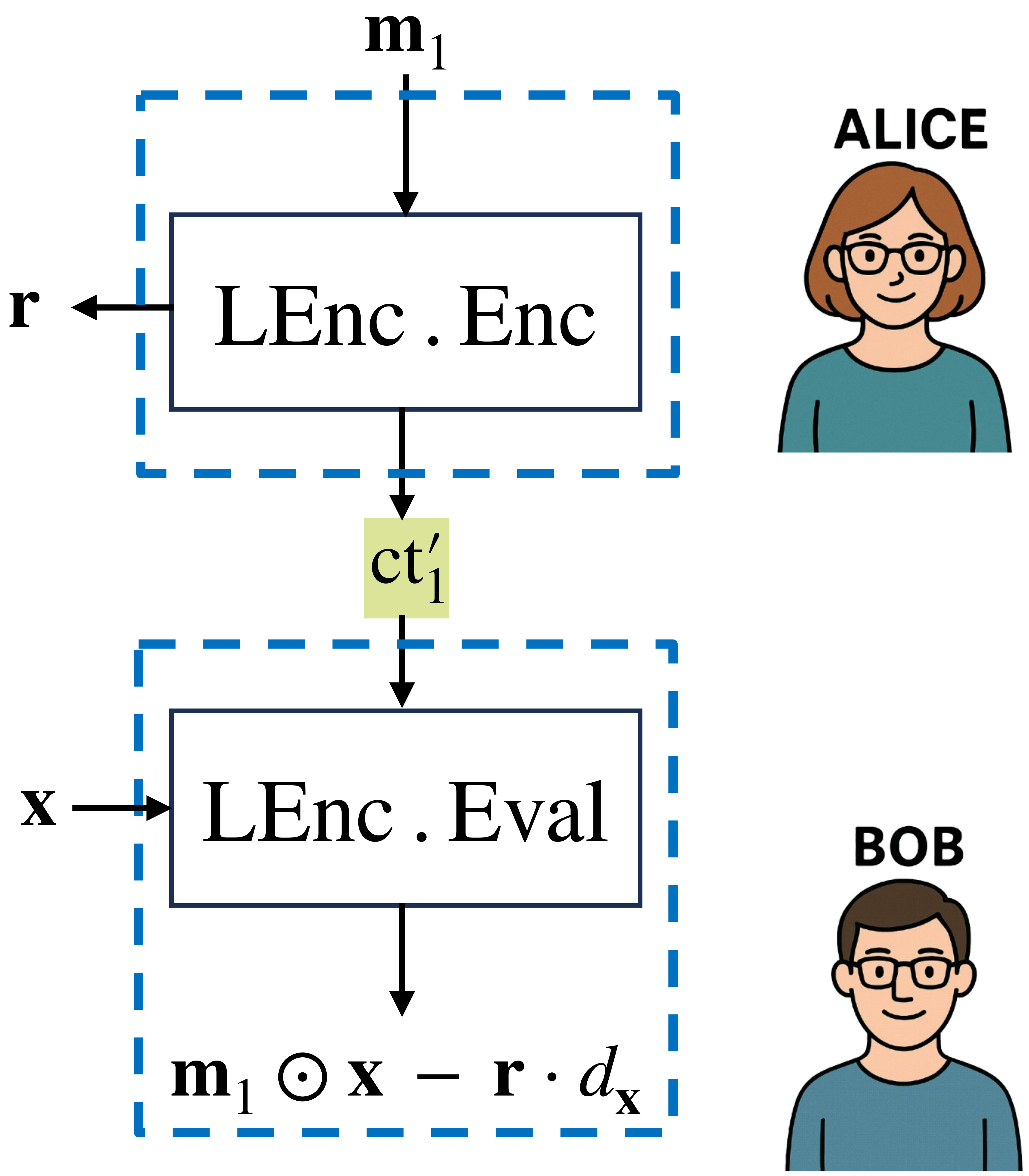
succinct

$\mathbf{x}$-dependent key

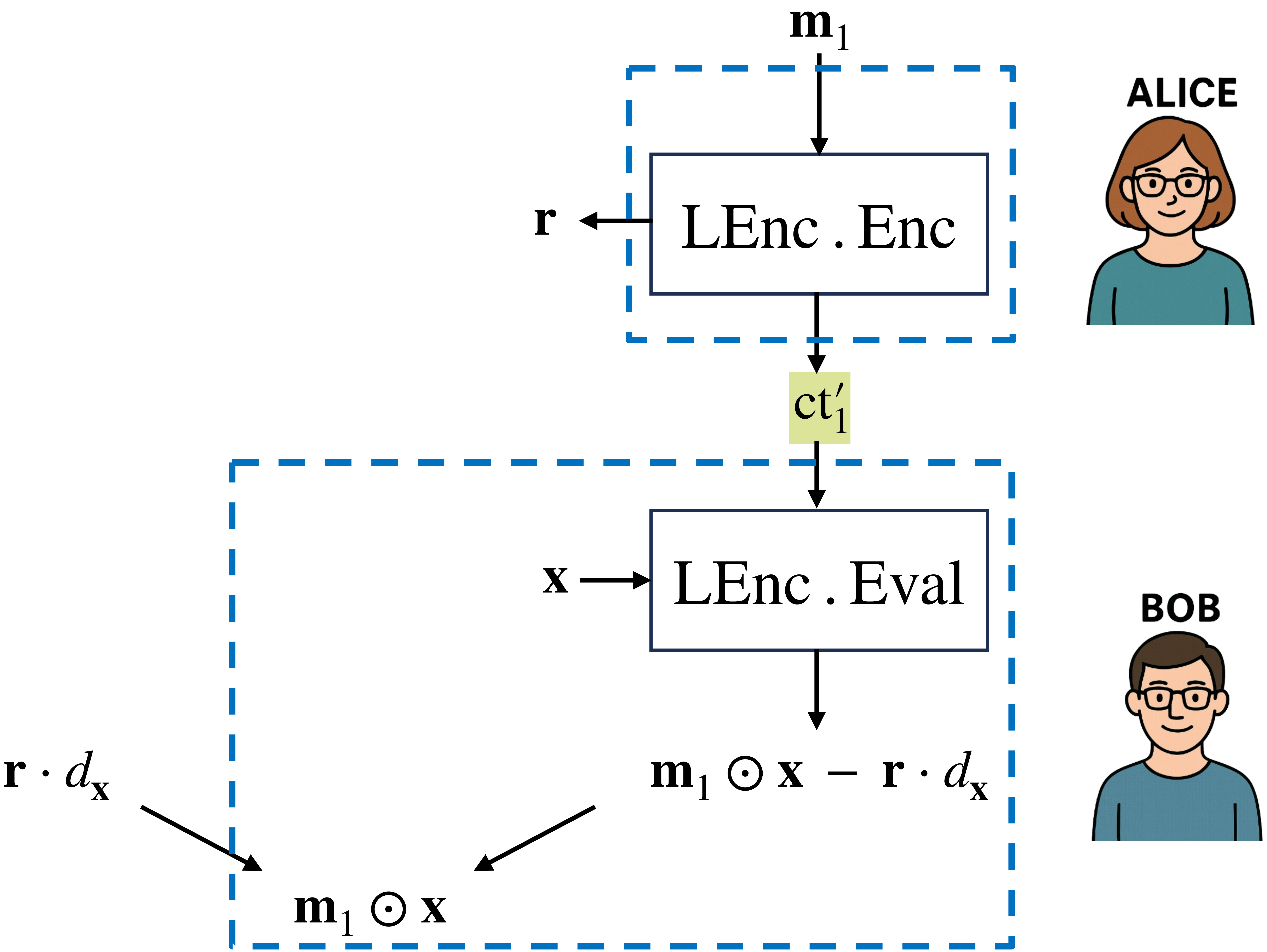
Constructing Batch-Select from RingLWE

LHE [AIK11,AIKW13,BLLL23,...]		LEnc [DKLLMR23]		Batch-Select
$\mathbf{m}_1 \cdot x + \mathbf{m}_2$	$+$	$\mathbf{m}_1 \odot \mathbf{x}$	$\Rightarrow$	$\mathbf{m}_1 \odot \mathbf{x} + \mathbf{m}_2$
public scalar				**public** vector
succinct		**non-succinct**		**succinct**
$\mathbf{x}$-dependent key		$\mathbf{x}$-dependent key		$\mathbf{x}$-dependent key
		$\mathrm{sk}_{\mathbf{x}} = \mathbf{r} \cdot d_{\mathbf{x}}$		

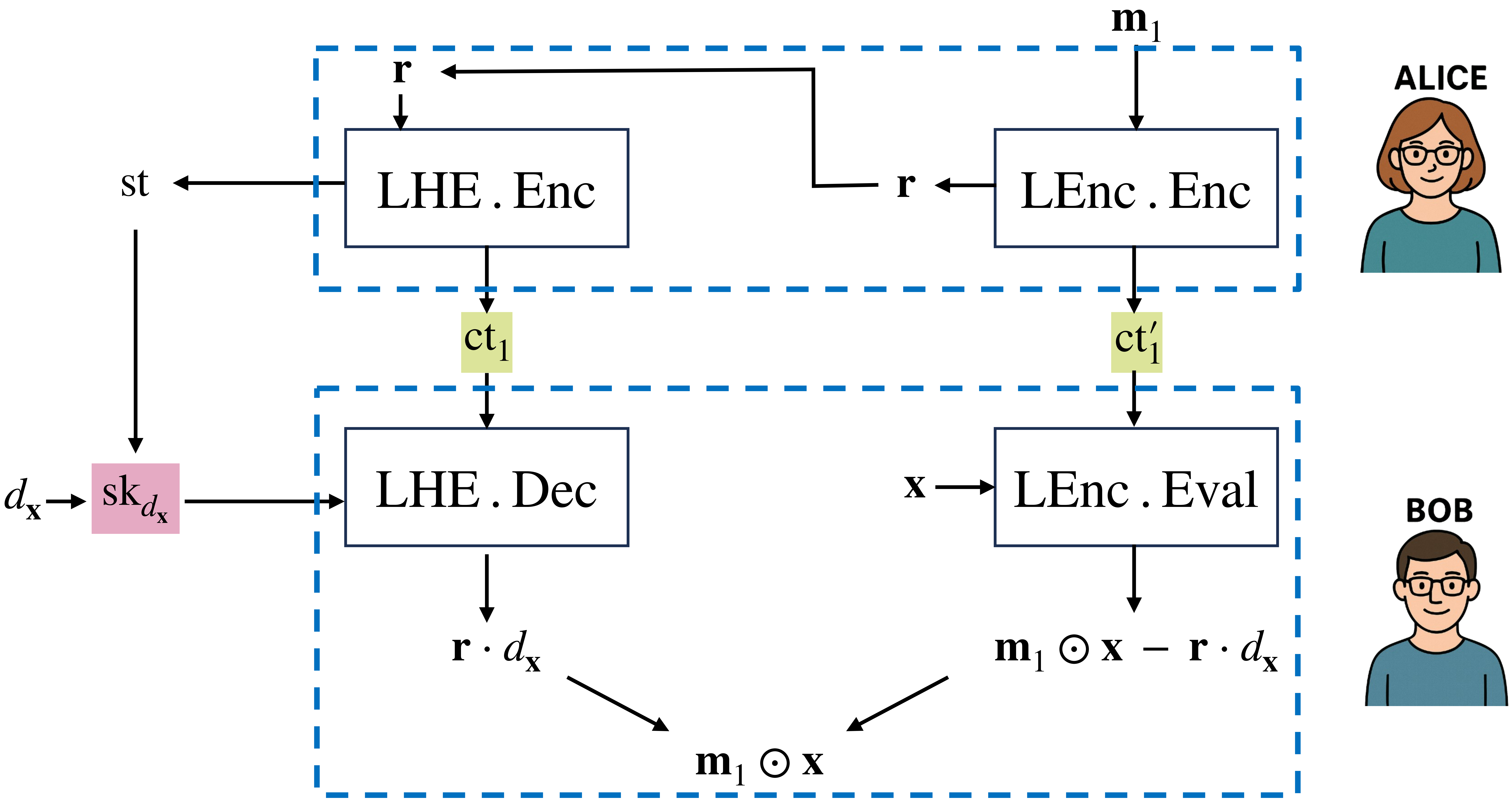
Constructing Batch-Select from RingLWE



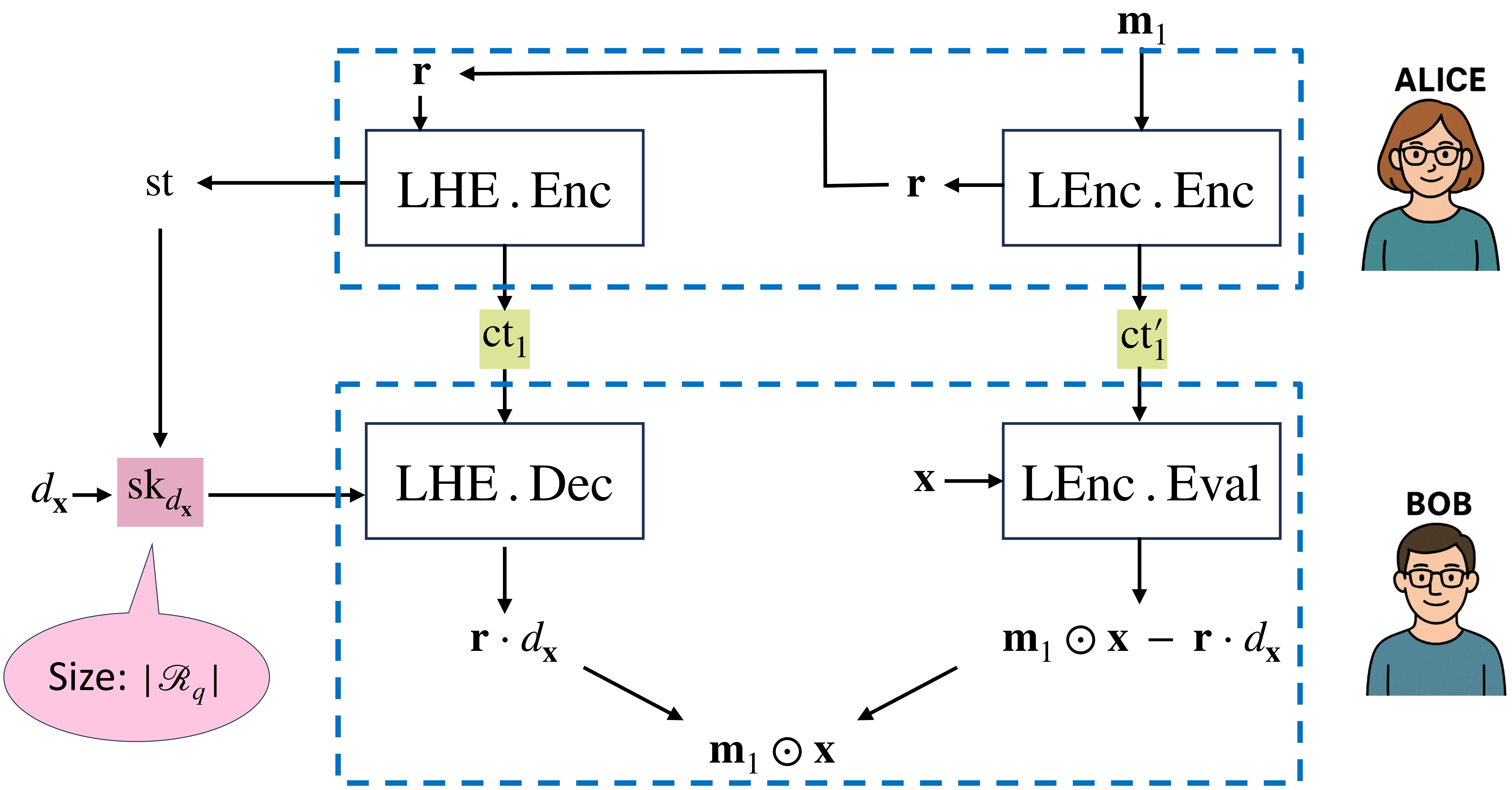
Constructing Batch-Select from RingLWE



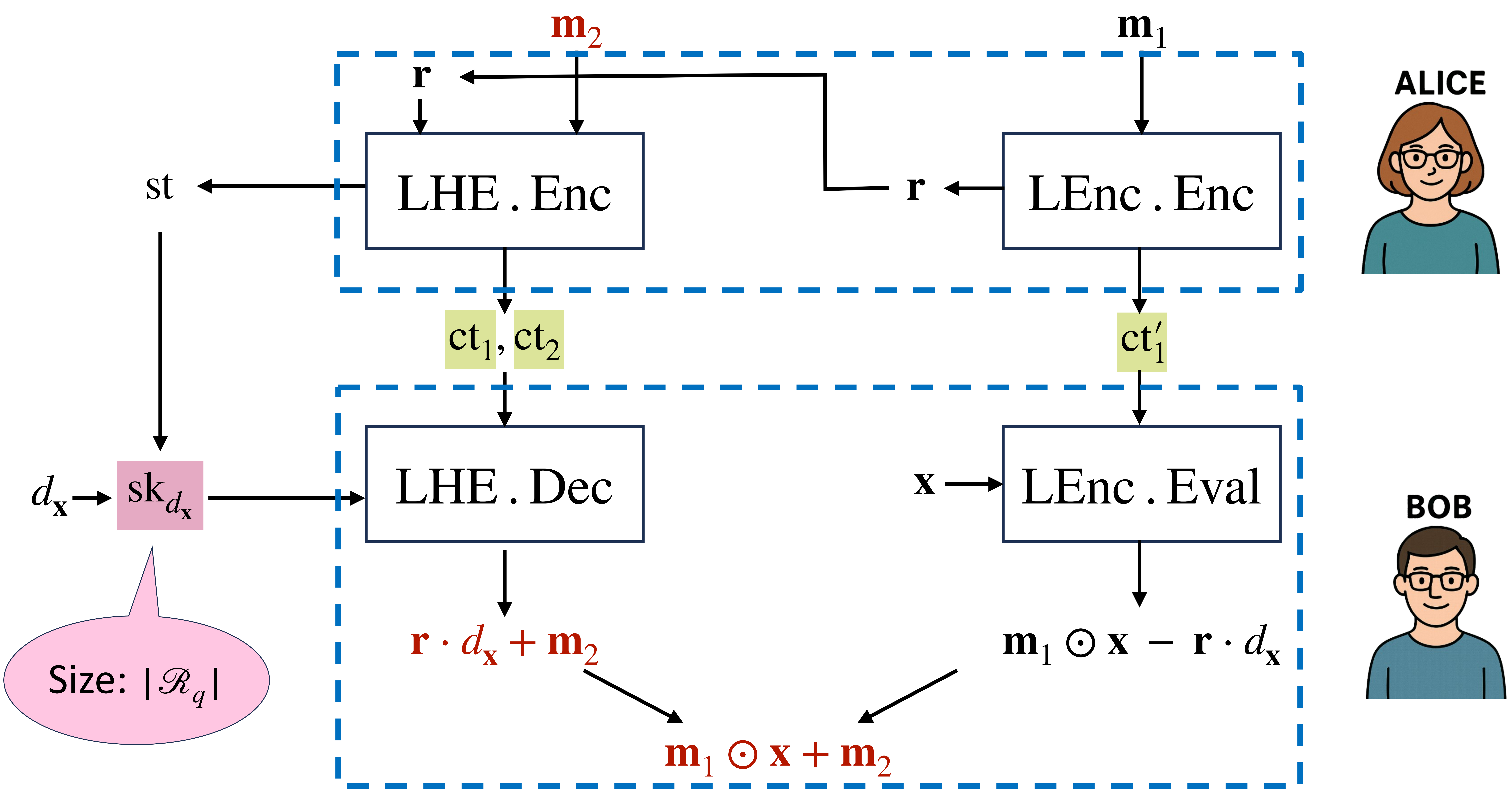
Constructing Batch-Select from RingLWE



Constructing Batch-Select from RingLWE



Constructing Batch-Select from RingLWE



Open Questions

- *Practical* label compression from other assumptions?
- Can succinct garbling *without offline phase* be made practical?

Open Questions

- *Practical* label compression from other assumptions?
- Can succinct garbling *without offline phase* be made practical?

Thank you!

https://ia.cr/2024/2048

References

- **[AIKW13]**: Applebaum, B., Ishai, Y., Kushilevitz, E., & Waters, B. (2013, August). **Encoding Functions with Constant Online Rate or How to Compress Garbled Circuits Keys**. In *Annual Cryptology Conference* (pp. 166-184). Berlin, Heidelberg: Springer Berlin Heidelberg.

- **[GS18]**: Garg, S., & Srinivasan, A. (2018). **Adaptively secure garbling with near optimal online complexity**. In Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37 (pp. 535-565). Springer International Publishing.

- **[GOS18]**: Garg, S., Ostrovsky, R., & Srinivasan, A. (2018, July). **Adaptive garbled RAM from laconic oblivious transfer**. In *Annual International Cryptology Conference* (pp. 515-544). Cham: Springer International Publishing.

- **[ABIKLV23]**: Applebaum, B., Beimel, A., Ishai, Y., Kushilevitz, E., Liu, T., & Vaikuntanathan, V. (2023, June). **Succinct computational secret sharing**. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (pp. 1553-1566).

- **[DKLLMR23]**: Döttling, N., Kolonelos, D., Lai, R. W., Lin, C., Malavolta, G., & Rahimi, A. (2023, April). **Efficient laconic cryptography from learning with errors**. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 417-446). Cham: Springer Nature Switzerland.

- Images generated with ChatGPT