Simultaneous-Message and Succinct Secure Computation

Eurocrypt 2025

Elette Boyle   Abhishek Jain   **Sacha Servan-Schreiber**   Akshayaram Srinivasan

NTT Research   Reichman University   MIT CSAIL   University of Toronto

SMS

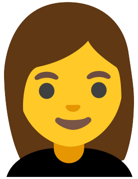
Secure Computation

Eurocrypt 2025

Elette Boyle Abhishek Jain **Sacha Servan-Schreiber** Akshayaram Srinivasan

NTT Research Reichman University MIT CSAIL University of Toronto

Secure Computation



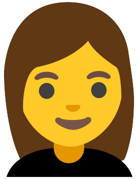
Alice

Bob

Secure Computation

Secure Computation

Private Inputs

x  Alice

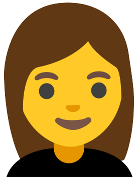
Want: $f(x, y)$

 y Bob

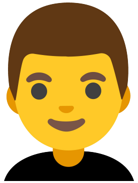
What can we dream of?

What can we dream of?

Inspired by Diffie-Hellman Key Exchange

x 

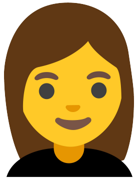
Alice

 y

Bob

What can we dream of?

Inspired by Diffie-Hellman Key Exchange

x 

Alice

$$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$$

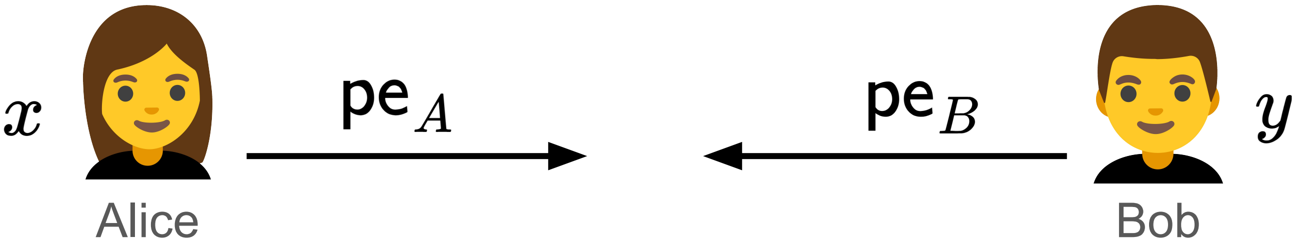
 y

Bob

$$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$$

What can we dream of?

Inspired by Diffie-Hellman Key Exchange

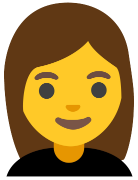
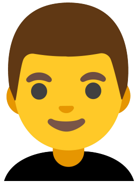


$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

What can we dream of?

Inspired by Diffie-Hellman Key Exchange

x  $\xrightarrow{\mathsf{pe}_A}$ $\xleftarrow{\mathsf{pe}_B}$  y

Alice Bob

$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

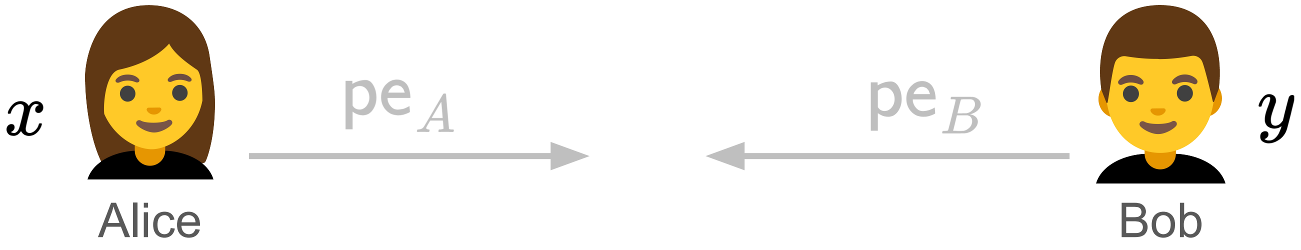
$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

$f(x, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

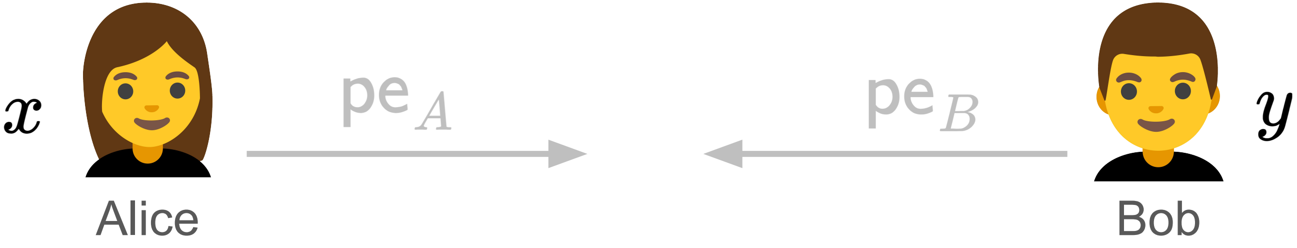
$f(x, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]

Attack: Compute $f(x_1, y), f(x_2, y) \ldots, f(x_n, y)$ using the same pe_B



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

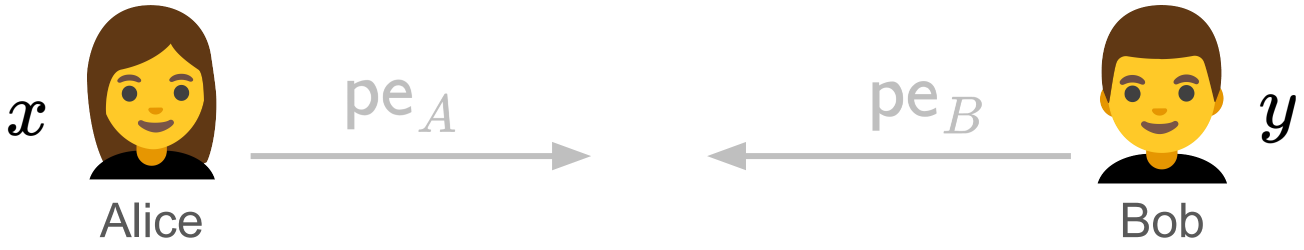
$f(x, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]

Attack: Compute $f(x_1, y), f(x_2, y) \ldots, f(x_n, y)$ using the same pe_B



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

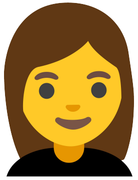
$f(x_1, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]

Attack: Compute $f(x_1, y), f(x_2, y) \ldots, f(x_n, y)$ using the same pe_B

x  $\xrightarrow{\mathsf{pe}_A}$ $\xleftarrow{\mathsf{pe}_B}$  y

Alice Bob

$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

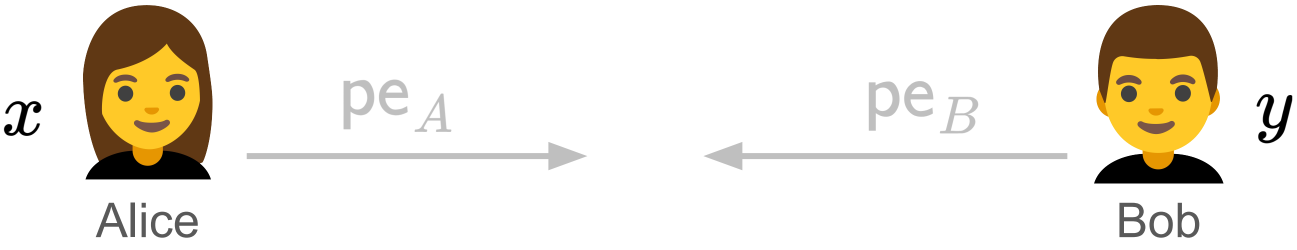
$f(x_2, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]

Attack: Compute $f(x_1, y), f(x_2, y) \ldots, f(x_n, y)$ using the same pe_B



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

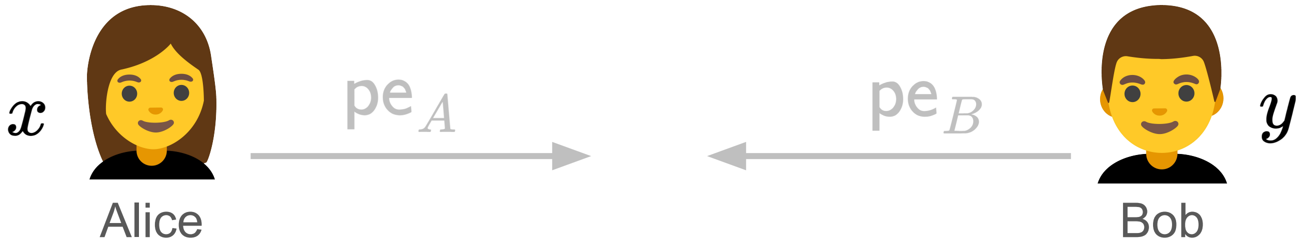
$f(x_3, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Impossible for arbitrary functions

Two-round lower-bound for two party computation [HLP'11]

Attack: Compute $f(x_1, y), f(x_2, y) \ldots, f(x_n, y)$ using the same pe_B



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

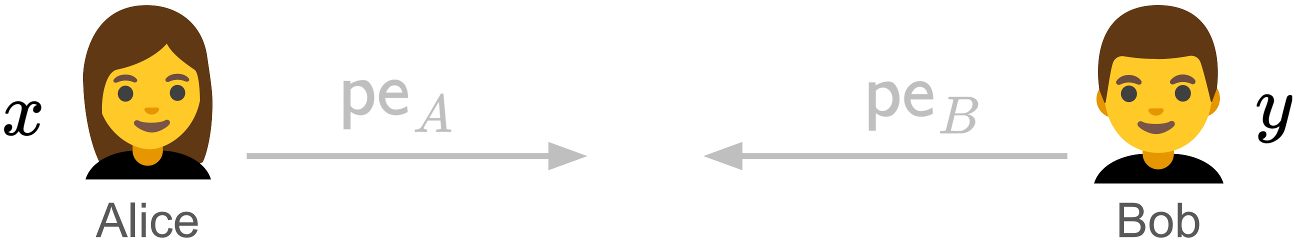
$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

$f(x_3, y) \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$f(x, y) \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Attack: Alice learns more than just $f(x, y)$

Possible for "secret shared" outputs



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

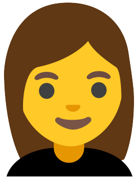
$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

$z_A \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$z_B \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Possible for “secret shared” outputs

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

x  $\xrightarrow{\mathsf{pe}_A}$ $\xleftarrow{\mathsf{pe}_B}$  y

Alice | Bob

$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

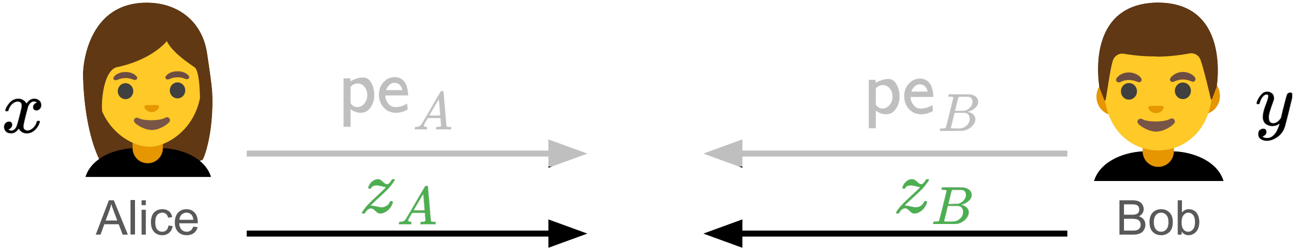
$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

$z_A \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$z_B \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

Possible for “secret shared” outputs

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



$\mathsf{pe}_A \leftarrow \mathsf{Encode}_A(x)$

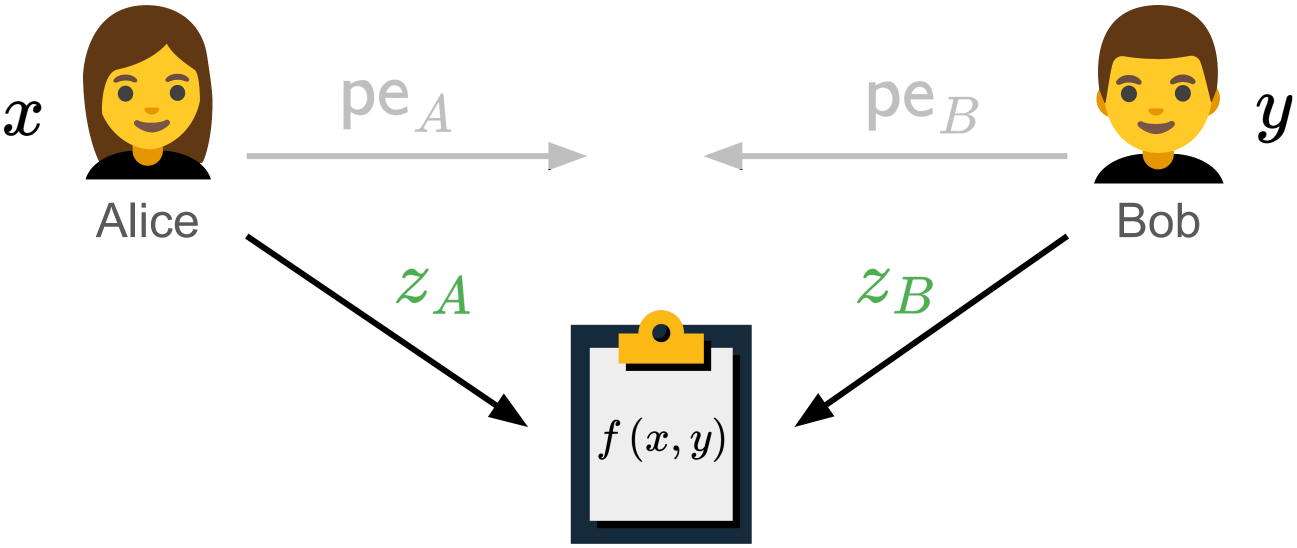
$\mathsf{pe}_B \leftarrow \mathsf{Encode}_B(y)$

$z_A \leftarrow \mathsf{Decode}_A(x, \mathsf{pe}_B)$

$z_B \leftarrow \mathsf{Decode}_B(y, \mathsf{pe}_A)$

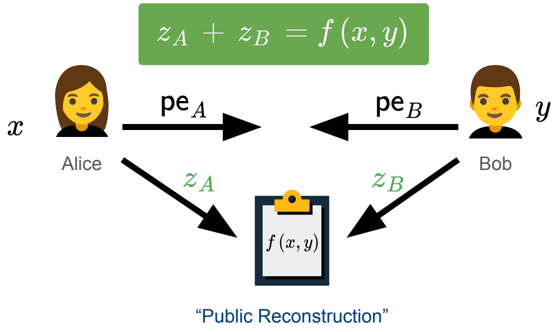
Possible for “secret shared” outputs

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

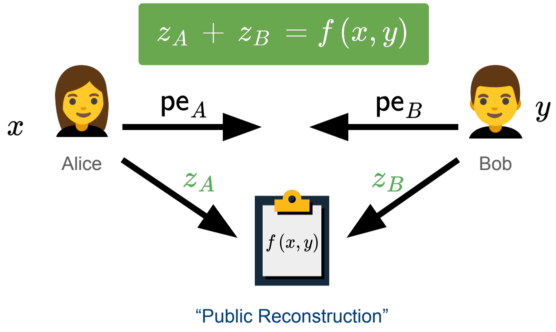


“Public Reconstruction”

Possible for "secret shared" outputs



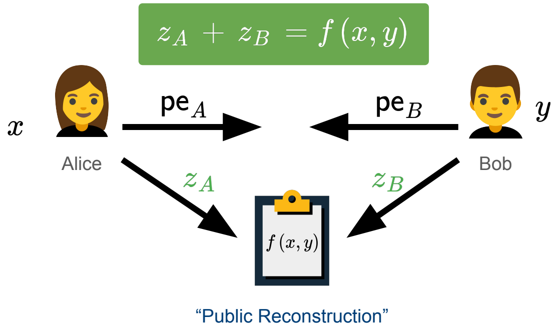
Possible for "secret shared" outputs



Spooky Encryption[DHRW'16]

From LWE or iO

Possible for "secret shared" outputs



Spooky Encryption[DHRW'16]

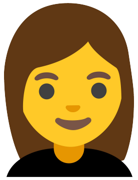
From LWE or iO

Multi-Key Homomorphic Secret Sharing[CDHJSS'16]

From DCR

Can we go further?

What can we dream of?

x   y

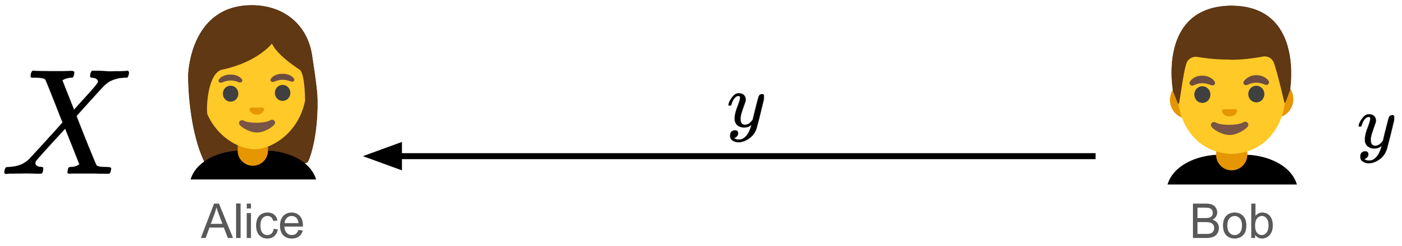
Alice

Bob

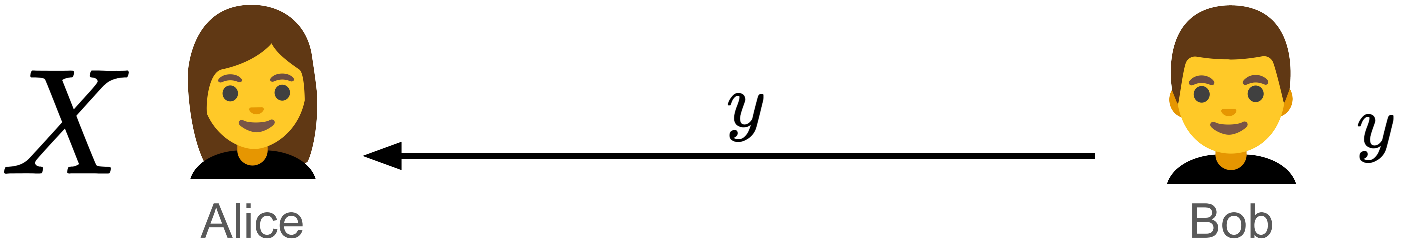
What can we dream of?

What can we dream of?

What can we dream of?

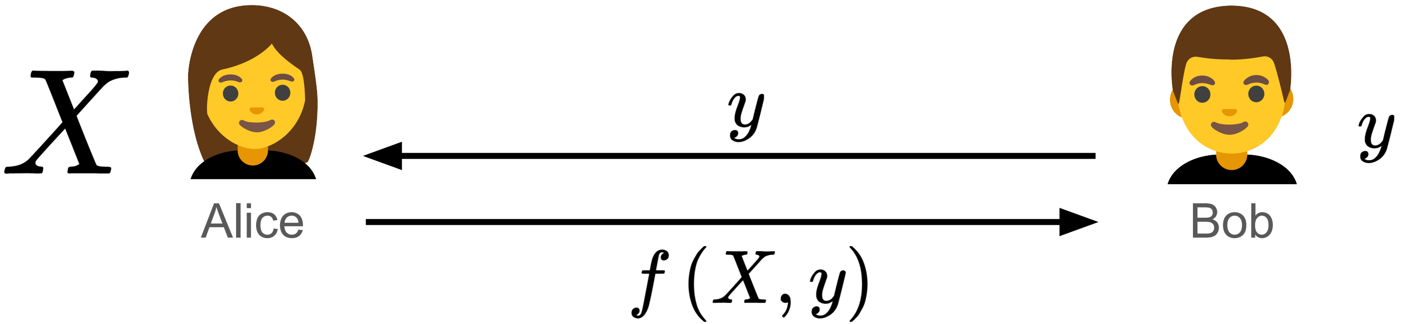


What can we dream of?



$$f\left(X, y\right)$$

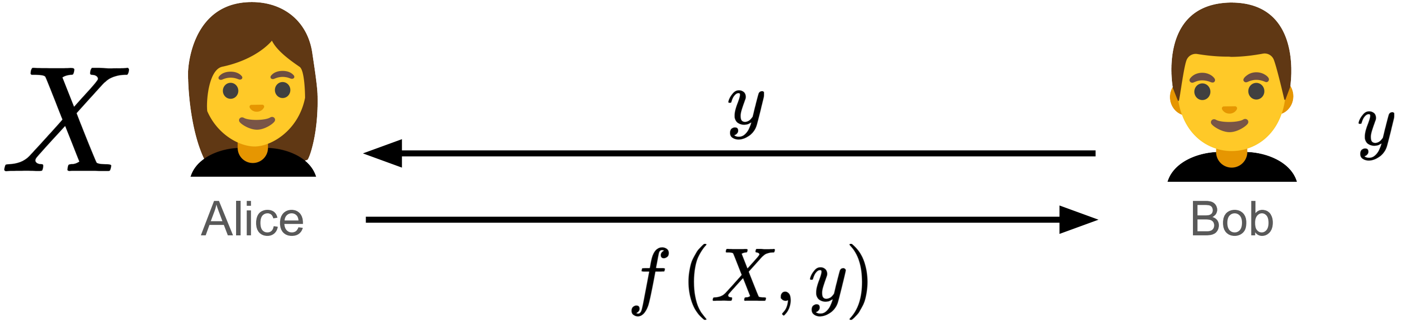
What can we dream of?



What can we dream of?

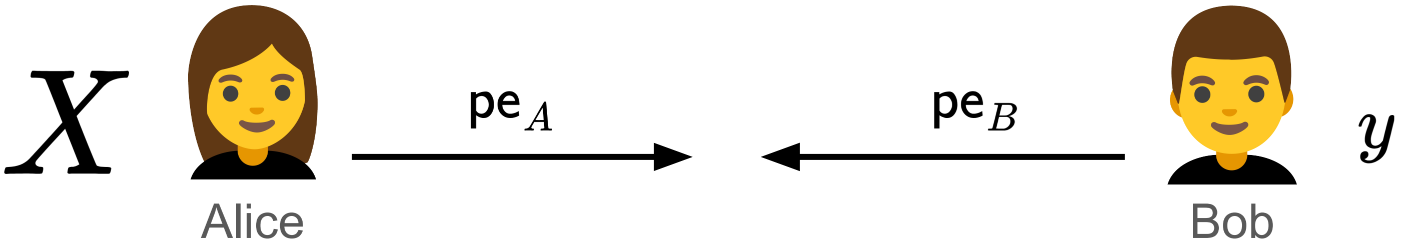
Alice learns y

Optimal communication $|y|$



Can we get a "*fully succinct*" protocol?

$$|\mathsf{pe}_\sigma| \leq (|X|^\epsilon + |f(X,y)|^\epsilon) \text{ for all } \sigma \in \{A, B\}$$



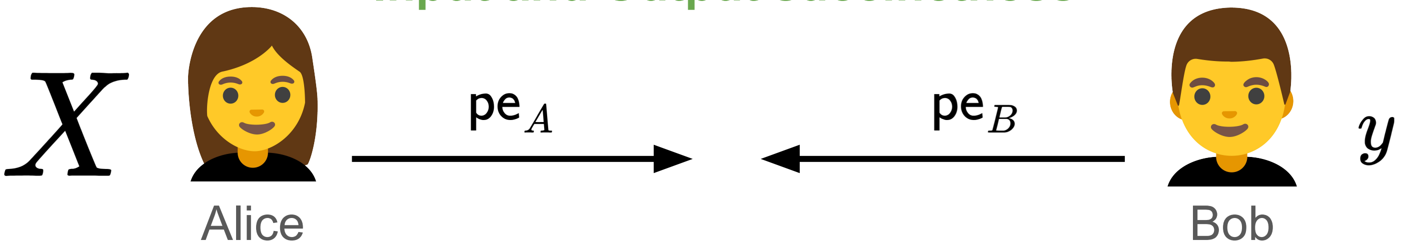
$(\mathsf{pe}_A, \mathsf{st}_A) \leftarrow \mathsf{Encode}_A(f, X)$

$(\mathsf{pe}_B, \mathsf{st}_B) \leftarrow \mathsf{Encode}_B(f, y)$

Can we get a "*fully succinct*" protocol?

$$|\mathsf{pe}_\sigma| \leq \left(|X|^\epsilon + |f(X,y)|^\epsilon\right) \text{ for all } \sigma \in \{A, B\}$$

"Input and Output succinctness"



$(\mathsf{pe}_A, \mathsf{st}_A) \leftarrow \mathsf{Encode}_A(f, X)$

$(\mathsf{pe}_B, \mathsf{st}_B) \leftarrow \mathsf{Encode}_B(f, y)$

	Assumptions	Input Succinct	Function Succinct	Comments
Succinct HSS [ARS'24]	**LWE / DCR**	✓	✓	**Vector OLE Functions**

	Assumptions	Input Succinct	Function Succinct	Comments
Succinct HSS [ARS'24]	**LWE / DCR**	✓	✓	**Vector OLE Functions**
SMS (this work)	**LWE**	✓	✓	**All functions**

	Assumptions	Input Succinct	Function Succinct	Comments
Succinct HSS [ARS'24]	LWE / DCR	✓	✓	Vector OLE Functions
SMS (this work)	LWE	✓	✓	All functions
SMS (this work)	iO + SSB Hash	✓	✓	Batch functions

	Assumptions	Input Succinct	Function Succinct	Comments
Succinct HSS [ARS'24]	**LWE / DCR**	✓	✓	**Vector OLE Functions**
SMS (this work)	**LWE**	✓	✓	**All functions**
SMS (this work)	**iO + SSB Hash**	✓	✓	**Batch functions**
Concurrent work [AMR'25]	**LWE**	✓	✓	**Near-optimal parameters**

Applications of SMS

SMS

Trapdoor hash functions

First construction supporting all functions

SMS

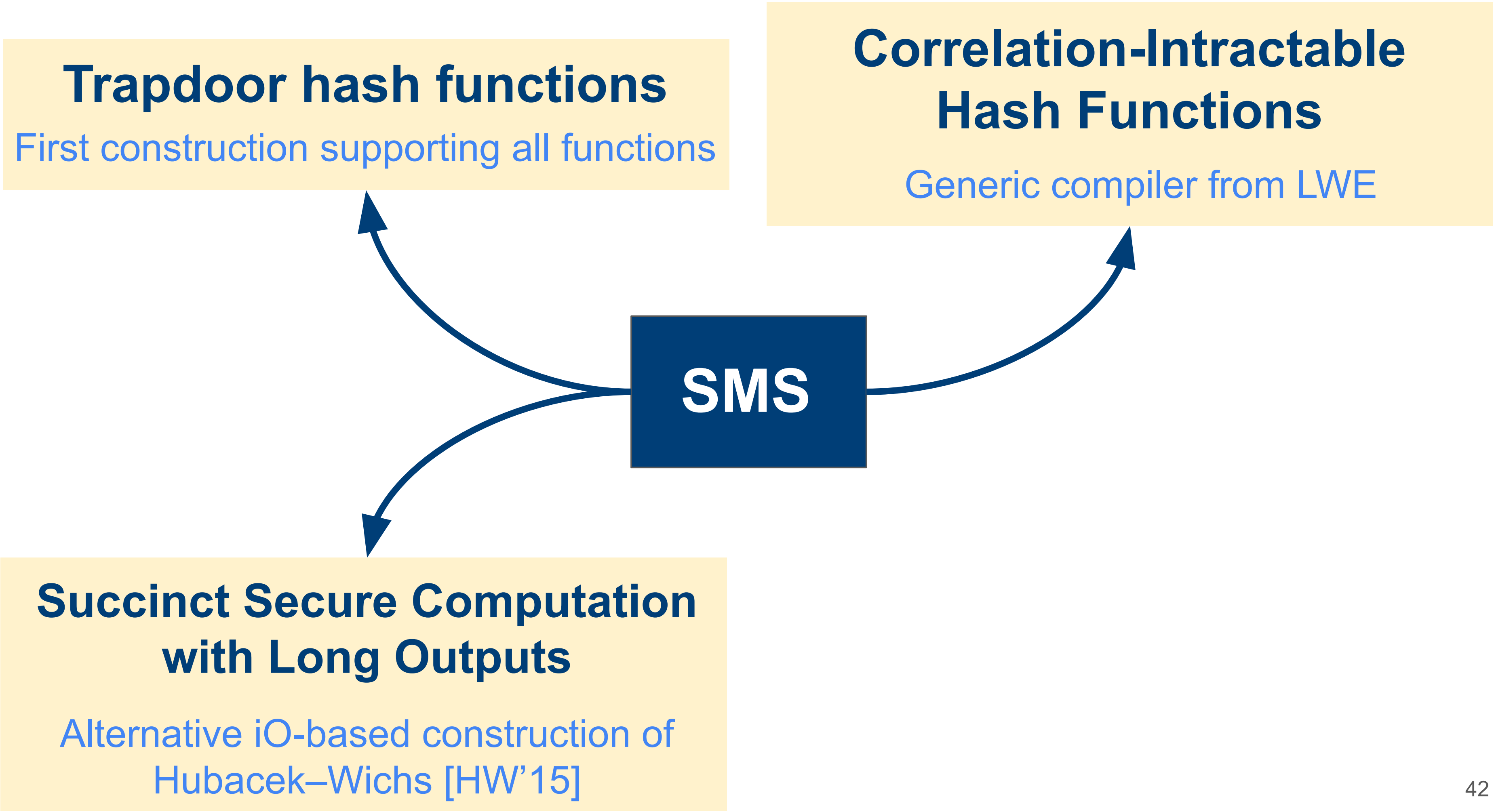
Trapdoor hash functions

First construction supporting all functions

Correlation-Intractable Hash Functions

Generic compiler from LWE

SMS


Trapdoor hash functions
First construction supporting all functions
Correlation-Intractable Hash Functions
Generic compiler from LWE
SMS
Succinct Secure Computation with Long Outputs
Alternative iO-based construction of Hubacek–Wichs [HW'15]

Trapdoor hash functions

First construction supporting all functions

Correlation-Intractable Hash Functions

Generic compiler from LWE

SMS

Succinct Secure Computation with Long Outputs

Alternative iO-based construction of Hubacek–Wichs [HW’15]

Rate-1 FHE

Generic compiler from any FHE scheme

SMS Construction

Ingredient I: FHE from LWE with "nice" decryption

$$\mathsf{FHE.KeyGen}\left(1^\lambda\right) : \mathsf{sk} \xleftarrow{R} \left(\mathbb{Z}_q^{n-1}, 1\right)$$

$$\mathsf{FHE.Encrypt}\left(\mathsf{sk},\ x\right) : \left(-\mathbf{a},\ \langle \mathbf{a}, \mathsf{sk}\rangle + \frac{q}{p}x + \mathsf{noise}\right)$$

$$\mathsf{FHE.Decrypt}\left(\mathsf{sk},\ \mathsf{ct}\right) : \lceil \langle \mathsf{ct}, \mathsf{sk}\rangle \rfloor_p$$

$$\langle \mathsf{ct}, \mathsf{sk}\rangle = \frac{q}{p}x + \mathsf{noise}$$

Ingredient I: FHE from LWE with "nice" decryption

$$\mathsf{FHE.KeyGen}\left(1^\lambda\right) : \mathsf{sk} \xleftarrow{R} \left(\mathbb{Z}_q^{n-1}, 1\right)$$

$$\mathsf{FHE.Encrypt}\left(\mathsf{sk}, x\right) : \left(-\mathbf{a}, \ \langle \mathbf{a}, \mathsf{sk}\rangle + \frac{q}{p}x + \mathsf{noise}\right)$$

$$\mathsf{FHE.Decrypt}\left(\mathsf{sk}, \ \mathsf{ct}\right) : \ \lceil \langle \mathsf{ct}, \mathsf{sk} \rangle \rfloor_p$$

$$\boxed{\langle \mathsf{ct}, \mathsf{sk}\rangle \ = \ \frac{q}{p}x + \mathsf{noise}}$$

"Near linear decryption"

Ingredient II: GVW Evaluation Algorithms

Building blocks from [GVW'15]:

- $\mathsf{EvalPK}\,(\mathsf{crs}\,,C) \;\to\; \mathbf{A}_C\cdot$

  Input: CRS and a circuit $C:\;\{0,1\}^{\alpha}\;\to\;\mathbb{Z}_q^{\beta}$

  Output: a public matrix $\mathbf{A}_C\;\in\mathbb{Z}_q^{n\times k}$

- $\mathsf{EvalCT}\left(\mathsf{crs}\,,\,\mathbf{u}_1,\,\ldots,\,\mathbf{u}_\alpha,\,\mathbf{v}_1,\,\ldots,\,\mathbf{v}_\beta, C,\,\hat{a}\right)\;\to\mathbf{w}_C$

  Input: CRS, $\alpha\;+\;\beta$ ciphertexts, the circuit C and public input $\hat{a}$ where:

$$\mathbf{u}_i\;=\;\mathbf{s}^\top\mathbf{A}_i+\hat{a}\,[i]\cdot\mathbf{G}+\text{noise},\;\;\text{for all } i\;\in[\alpha]$$

$$\mathbf{v}_i\;=\;\mathbf{s}^\top\mathbf{B}_i+\hat{\mathbf{b}}\,[i]\cdot\mathbf{G}+\text{noise},\;\;\text{for all } i\;\in[\beta]$$

  Output: a ciphertext $\mathbf{w}_C\;=\;\mathbf{s}^\top\left(\mathbf{A}_C+\left\langle C\left(\hat{a}\right),\,\hat{\mathbf{b}}\right\rangle\cdot\mathbf{G}\right)+\text{noise}$

SMS Construction

Getting input succinctness

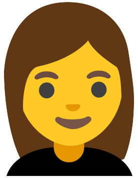
Building SMS with Input Succinctness

X 👩 Alice

👨 y Bob

Building SMS with Input Succinctness

$$\mathsf{Hash}\left(X\right) \rightarrow \mathsf{pe}_A$$

X 

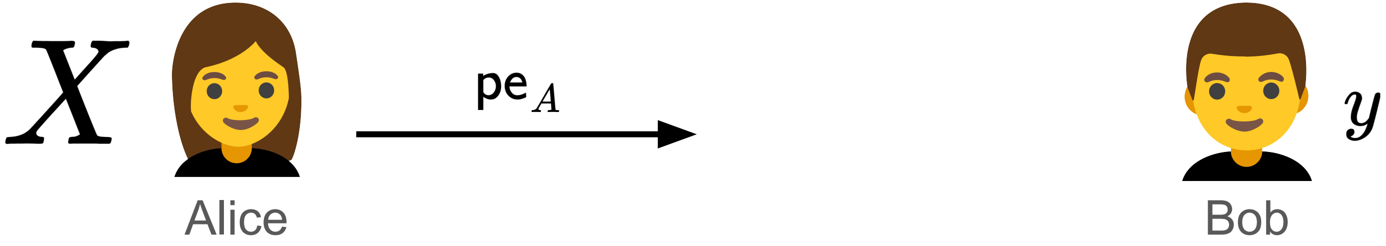
Alice

 y

Bob

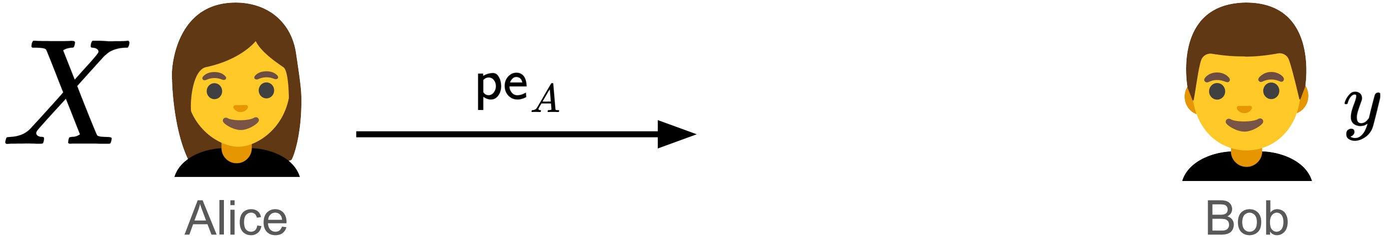
Building SMS with Input Succinctness

$$\mathsf{Hash}(X) \to \mathsf{pe}_A$$



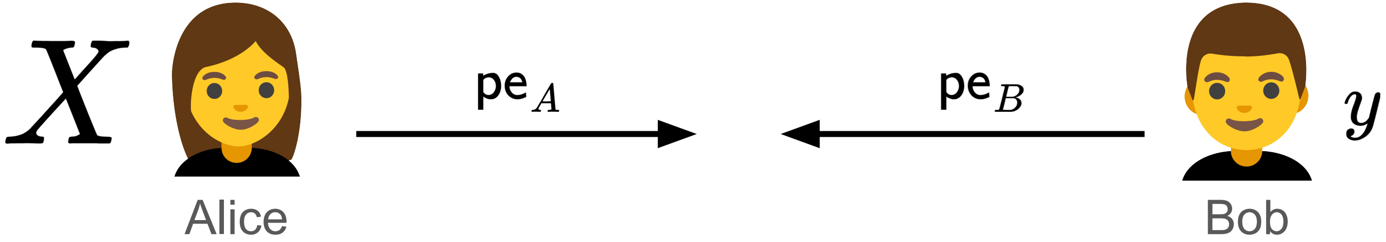
Building SMS with Input Succinctness

$$\mathsf{Hash}(X) \to \mathsf{pe}_A \qquad \left.\begin{array}{l} \mathsf{ct}_y := \mathsf{Encrypt}(🔑, y) \\ \mathsf{ct} \;\; := \mathsf{Encrypt}(🗝, 🔑) \end{array}\right\} \mathsf{pe}_B$$



Building SMS with Input Succinctness

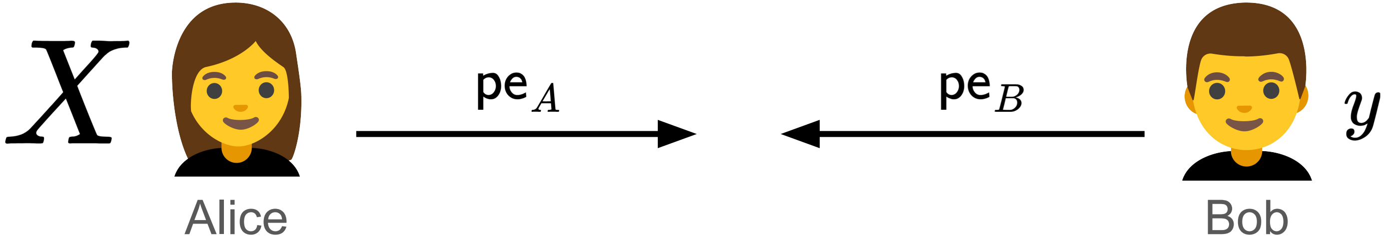
$$\mathsf{Hash}(X) \to \mathsf{pe}_A \qquad \left.\begin{aligned} \mathsf{ct}_y &:= \mathsf{Encrypt}(🔑, y) \\ \mathsf{ct} &:= \mathsf{Encrypt}(🔑, 🔑) \end{aligned}\right\} \mathsf{pe}_B$$



Building SMS with Input Succinctness

$$\mathsf{Hash}(X) \rightarrow \mathsf{pe}_A$$

$$\left.\begin{aligned}\mathsf{ct}_y &:= \mathsf{Encrypt}(🔑, y)\\ \mathsf{ct} &:= \mathsf{Encrypt}(🗝, 🔑)\end{aligned}\right\}\ \mathsf{pe}_B$$

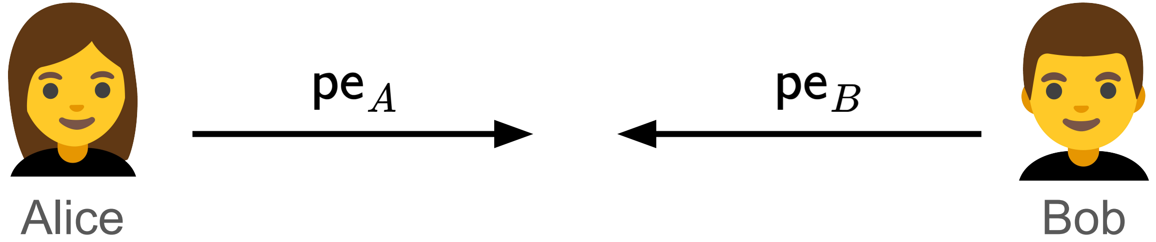


$$\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}(f, X, \mathsf{ct}_y)$$

Building SMS with Input Succinctness

$$\mathsf{Hash}\left(X\right) \rightarrow \mathsf{pe}_A$$

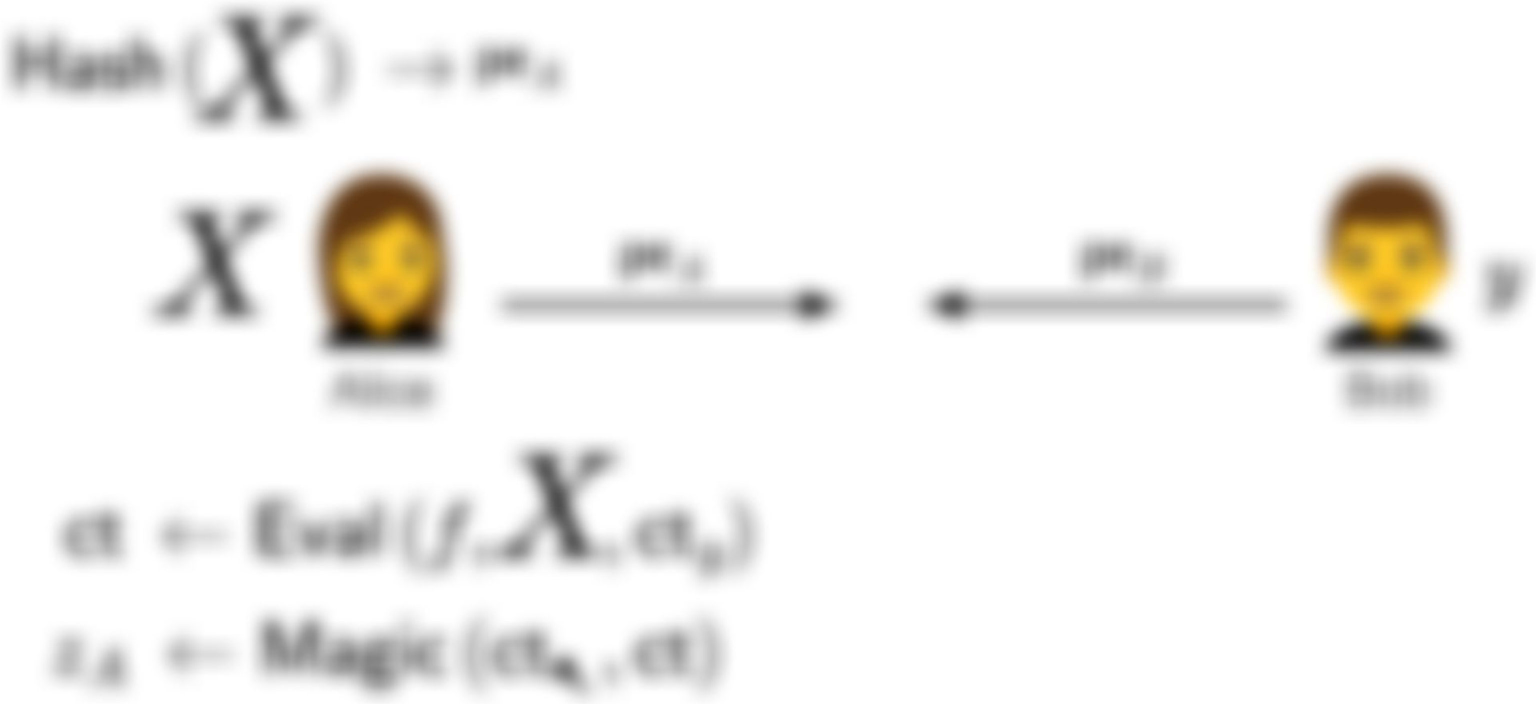
$$\left.\begin{array}{l} \mathsf{ct}_y := \mathsf{Encrypt}\left(\text{🔑}, y\right) \\ \mathsf{ct} := \mathsf{Encrypt}\left(\text{🗝}, \text{🔑}\right) \end{array}\right\} \mathsf{pe}_B$$

X  y

$$\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}\left(f, X, \mathsf{ct}_y\right)$$

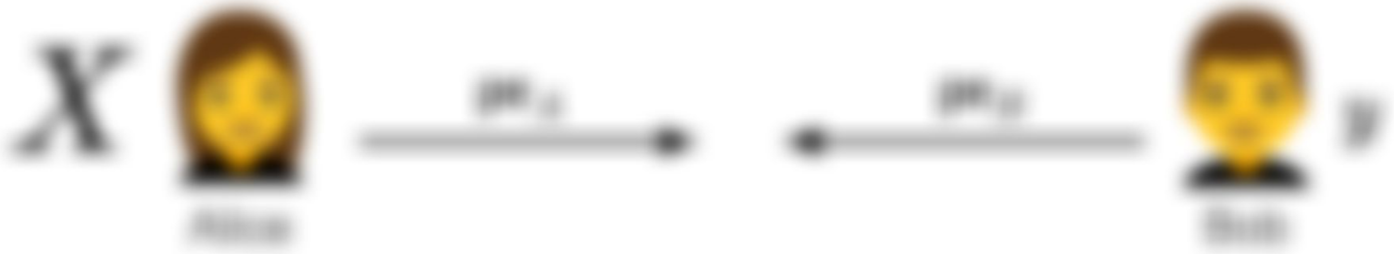
$$z_A \leftarrow \mathsf{Magic}\left(\mathsf{ct}_{\text{🔑}}, \hat{\mathsf{ct}}\right)$$

Building SMS with Input Succinctness



Building SMS with Input Succinctness

$\mathsf{EvalPK}(X) \rightarrow \mathsf{pe}_A$



$\hat{\mathsf{ct}} \leftarrow \mathsf{EvalCT}(f, X, \mathsf{ct}_y)$

$z_A \leftarrow$ Near linear decryption

Building SMS with Input Succinctness

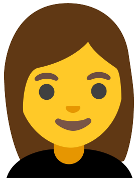
$$f : \{0,1\}^{\mathsf{BIG}} \times \{0,1\}^{\mathsf{small}} \to \{0,1\}$$

Building SMS with Input Succinctness

$f : \{0,1\}^{\mathsf{BIG}} \times \{0,1\}^{\mathsf{small}} \to \{0,1\}$

Building SMS with Input Succinctness

X



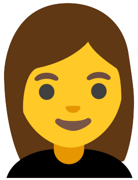
Alice

$$f : \{0,1\}^{\mathsf{BIG}} \times \{0,1\}^{\mathsf{small}} \to \{0,1\}$$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}\,(f,\ X,\ \mathsf{ct})$

X

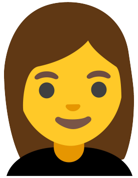


Alice

$f : \{0,1\}^{\mathsf{BIG}} \times \{0,1\}^{\mathsf{small}} \to \{0,1\}$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct
and computes $\mathsf{FHE.Eval}\,(f,\ X,\ \mathsf{ct})$

X 

Alice

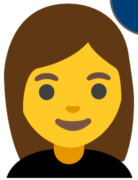
$\mathbf{A}_C \leftarrow \mathsf{EvalPK}\,(\mathsf{crs}\,,C)$

$$f : \{0,1\}^{\mathsf{BIG}} \times \{0,1\}^{\mathsf{small}} \to \{0,1\}$$

Building SMS with Input Succinctness

$$|\mathbf{A}_C| = \mathsf{poly}\left(\mathsf{depth}\left(C\right), \lambda\right)$$

"It's very small"

X  $\xrightarrow{\mathbf{A}_C}$  y

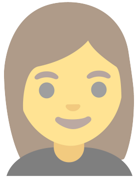
Alice

Bob

$$\mathbf{A}_C \leftarrow \mathsf{EvalPK}\left(\mathsf{crs}, C\right)$$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X  $\xrightarrow{\mathbf{A}_C}$  y

Alice

Bob

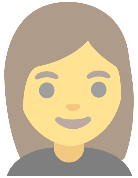
$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X



Alice

$\xrightarrow{\mathbf{A}_C}$

Bob

y

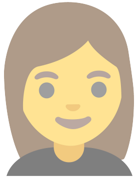
$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct
and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X  $\xrightarrow{\mathbf{A}_C}$

Alice

 y

Bob

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

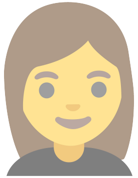
$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$

$\mathbf{s} \leftarrow (1, \mathsf{random}) \in \mathbb{Z}_q^n$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes FHE. Eval (f, X, ct)

X  $\xrightarrow{\mathbf{A}_C}$  y

Alice Bob

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

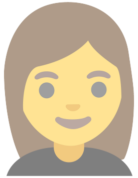
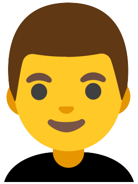
$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$

$\mathbf{s} \leftarrow (1, \mathsf{random}) \in \mathbb{Z}_q^n$

$\mathbf{u}_i = \mathbf{s}^\top \mathbf{A}_i + \mathsf{ct}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\alpha]$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X  $\xrightarrow{\mathbf{A}_C}$  y

Alice

Bob

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$$

$$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$$

$$\mathbf{s} \leftarrow (1, \mathsf{random}) \in \mathbb{Z}_q^n$$

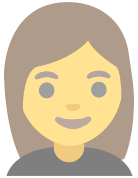
$$\mathbf{u}_i = \mathbf{s}^\top \mathbf{A}_i + \mathsf{ct}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\alpha]$$

$$\mathbf{v}_i = \mathbf{s}^\top \mathbf{B}_i + \mathsf{sk}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\beta]$$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes FHE. Eval (f, X, ct)

X



Alice

$\xrightarrow{\mathbf{A}_C}$

Bob

y

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$

$\mathbf{s} \leftarrow (1, \mathsf{random}) \in \mathbb{Z}_q^n$

Nested encryption of y →

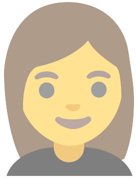
$$\mathbf{u}_i = \mathbf{s}^\top \mathbf{A}_i + \mathsf{ct}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\alpha]$$

$$\mathbf{v}_i = \mathbf{s}^\top \mathbf{B}_i + \mathsf{sk}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\beta]$$

Building SMS with Input Succinctness

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X



Alice

$\xrightarrow{\mathbf{A}_C}$

Bob

y

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$

$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$

$\mathbf{s} \leftarrow (1, \mathsf{random}) \in \mathbb{Z}_q^n$

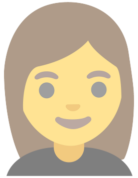
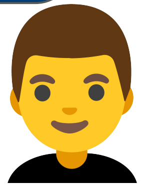
Encryption of sk $\longrightarrow$

$$\mathbf{u}_i = \mathbf{s}^\top \mathbf{A}_i + \mathsf{ct}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\alpha]$$
$$\mathbf{v}_i = \mathbf{s}^\top \mathbf{B}_i + \mathsf{sk}[i] \cdot \mathbf{G} + \mathsf{noise}, \quad \text{for all } i \in [\beta]$$

Building SMS with Input S

Size poly $(|y|, \lambda)$
"Independent of $|X|$"

C takes as input an FHE ciphertext ct and computes $\mathsf{FHE.Eval}(f, X, \mathsf{ct})$

X  $\xrightarrow{\mathbf{A}_C}$ $\xleftarrow{(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)}$  y

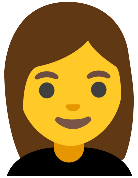
Alice

Bob

$\mathbf{A}_C \leftarrow \mathsf{EvalPK}(\mathsf{crs}, C)$

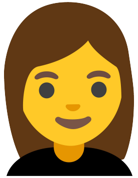
$$\mathsf{sk} \leftarrow \mathsf{FHE.KeyGen}(1^\lambda)$$
$$\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{sk}, y)$$
$$\mathbf{s} \leftarrow (1, \text{random}) \in \mathbb{Z}_q^n$$
$$\mathbf{u}_i = \mathbf{s}^\top \mathbf{A}_i + \mathsf{ct}[i] \cdot \mathbf{G} + \text{noise}, \quad \text{for all } i \in [\alpha]$$
$$\mathbf{v}_i = \mathbf{s}^\top \mathbf{B}_i + \mathsf{sk}[i] \cdot \mathbf{G} + \text{noise}, \quad \text{for all } i \in [\beta]$$

Building SMS with Input Succinctness



Alice

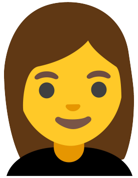
Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

Building SMS with Input Succinctness

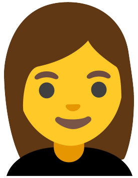


Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

Building SMS with Input Succinctness



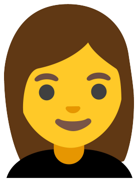
Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

$$\mathbf{w}_C[1] = \mathbf{s}^\top (\mathbf{A}_C + \langle C(\mathsf{ct}), \mathsf{sk}\rangle \cdot \mathbf{G})[1] + \mathsf{noise}$$ // correctness of EvalCT

Building SMS with Input Succinctness



Alice

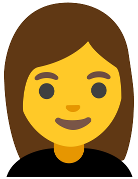
$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

$$\mathbf{w}_C[1] = \mathbf{s}^\top (\mathbf{A}_C + \langle C(\mathsf{ct}), \mathsf{sk}\rangle \cdot \mathbf{G})[1] + \mathsf{noise} \qquad \text{// correctness of EvalCT}$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle C(\mathsf{ct}), \mathsf{sk}\rangle + \mathsf{noise} \qquad \text{// because } \mathbf{s}[1] = 1$$

Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

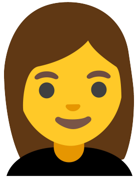
$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

$$\mathbf{w}_C[1] = \mathbf{s}^\top (\mathbf{A}_C + \langle C(\mathsf{ct}), \mathsf{sk}\rangle \cdot \mathbf{G})[1] + \mathsf{noise} \qquad \text{// correctness of EvalCT}$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle C(\mathsf{ct}), \mathsf{sk}\rangle + \mathsf{noise} \qquad \text{// because } \mathbf{s}[1] = 1$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle \mathsf{FHE.Eval}(f, (X, \mathsf{ct})), \mathsf{sk}\rangle + \mathsf{noise}$$

Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

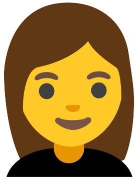
$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

$$\mathbf{w}_C[1] = \mathbf{s}^\top (\mathbf{A}_C + \langle C(\mathsf{ct}), \mathsf{sk}\rangle \cdot \mathbf{G})[1] + \text{noise} \qquad \text{// correctness of EvalCT}$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle C(\mathsf{ct}), \mathsf{sk}\rangle + \text{noise} \qquad \text{// because } \mathbf{s}[1] = 1$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle \mathsf{FHE.Encrypt}(\mathsf{sk}, f(X, y)), \mathsf{sk}\rangle + \text{noise} \qquad \text{// correctness}$$

Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$$\mathbf{w}_C \leftarrow \mathsf{EvalCT}(\mathsf{crs}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta, C, \mathsf{ct})$$

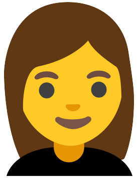
$$\mathbf{w}_C[1] = \mathbf{s}^\top (\mathbf{A}_C + \langle C(\mathsf{ct}), \mathsf{sk}\rangle \cdot \mathbf{G})[1] + \mathsf{noise} \qquad \text{// correctness of EvalCT}$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle C(\mathsf{ct}), \mathsf{sk}\rangle + \mathsf{noise} \qquad \text{// because } \mathbf{s}[1] = 1$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \langle \mathsf{FHE.Encrypt}(\mathsf{sk}, f(X, y)), \mathsf{sk}\rangle + \mathsf{noise} \qquad \text{// correctness}$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \mathsf{noise} \qquad \text{// near-linear decryption of FHE}$$

Building SMS with Input Succinctness

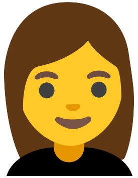


Alice

$$(\mathsf{ct}\,,\,\mathbf{u}_1,\,\ldots,\,\mathbf{u}_\alpha,\,\mathbf{v}_1,\,\ldots,\,\mathbf{v}_\beta)$$

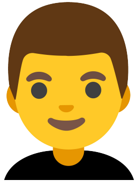
$$z_A := \mathbf{s}^\top \mathbf{A}_C\,[1] + \tfrac{q}{p} f\,(X, y) + \text{noise}$$

Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

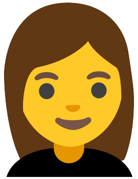


Bob

$$\mathbf{s} := (1, \mathsf{random})$$

$$z_A := \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \mathsf{noise}$$

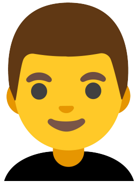
Building SMS with Input Succinctness



Alice

$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$

$\mathbf{A}_C$



Bob

$\mathbf{s} := (1, \mathsf{random})$

$$z_A := \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \mathsf{noise}$$

Building SMS with Input Succinctness

Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$\mathbf{A}_C$

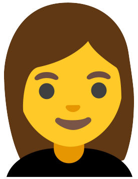
Bob

$\mathbf{s} := (1, \mathsf{random})$

$$z_A := \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \mathsf{noise}$$

$$z_B := -\left(\mathbf{s}^\top \mathbf{A}_C\right)[1]$$

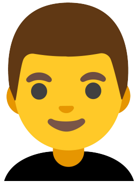
Building SMS with Input Succinctness



Alice

$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$

$\mathbf{A}_C$



Bob

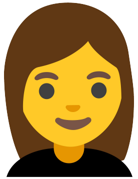
$\mathbf{s} := (1, \mathsf{random})$

$z_A := \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \text{noise}$

$z_B := -\left(\mathbf{s}^\top \mathbf{A}_C\right)[1]$

$$z_A + z_B = \frac{q}{p} f(X, y) + \text{noise}$$

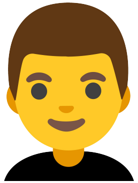
Building SMS with Input Succinctness



Alice

$$(\mathsf{ct}, \mathbf{u}_1, \ldots, \mathbf{u}_\alpha, \mathbf{v}_1, \ldots, \mathbf{v}_\beta)$$

$\mathbf{A}_C$

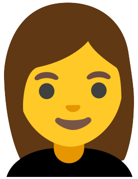


Bob

$\mathbf{s} := (1, \mathsf{random})$

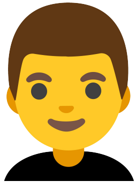
$$z_A := \lceil \mathbf{s}^\top \mathbf{A}_C[1] + \frac{q}{p} f(X, y) + \mathsf{noise} \rfloor_p \qquad z_B := -\lceil (\mathbf{s}^\top \mathbf{A}_C)[1] \rfloor_p$$

Building SMS with Input Succinctness



Alice

Lemma (Rounding of Noisy Shares): Assuming LWE with *superpolynomial modulus-to-noise ratio*, rounding of two noisy shares results in additive shares.



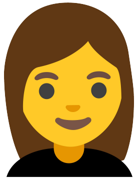
Bob

$\mathbf{s} := (1, \mathsf{random})$

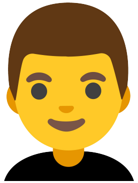
$$z_A := \lceil \mathbf{s}^\top \mathbf{A}_C[1] + \tfrac{q}{p} f(X, y) + \mathsf{noise} \rfloor_p = \mathbf{s}^\top \mathbf{A}_C[1] + f(X, y) \pmod{p}$$

$$z_B := -\lceil (\mathbf{s}^\top \mathbf{A}_C)[1] \rfloor_p = -(\mathbf{s}^\top \mathbf{A}_C)[1] \pmod{p}$$

Building SMS with Input Succinctness



Alice



Bob

$\mathbf{s} := (1, \mathsf{random})$

$$z_A := \left\lceil \mathbf{s}^\top \mathbf{A}_C[1] + \tfrac{q}{p} f(X, y) + \mathsf{noise} \right\rfloor_p$$

$$= \mathbf{s}^\top \mathbf{A}_C[1] + f(X, y) \pmod{p}$$

$$z_B := -\left\lceil \left(\mathbf{s}^\top \mathbf{A}_C\right)[1] \right\rfloor_p$$

$$= -\left(\mathbf{s}^\top \mathbf{A}_C\right)[1] \pmod{p}$$

$$z_A + z_B = f(X, y)$$

Long outputs?

Long outputs?

Too long to explain;
Idea: “Bootstrap” from SMS for vector OLE [ARS’24]

Questions?

Email: 3s@mit.edu
ePrint: ia.cr/**2025/096**

Simultaneous-Message and Succinct Secure Computation

Elette Boyle[1,2], Abhishek Jain[1,3], Sacha Servan-Schreiber[4*], and Akshayaram Srinivasan[5]

[1] NTT Research
[2] Reichman University
[3] JHU
[4] MIT
[5] University of Toronto

References

[HLP'11]: S. Halevi, Y. Lindell, and B. Pinkas. "Secure Computation on the Web: Computing without Simultaneous Interaction."

[BGG+'14]: D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. "Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE, and Compact Garbled Circuits."

[GVW'15]: S. Gorbunov, V. Vaikuntanathan, and H. Wee. "Predicate Encryption for Circuits from LWE."

[HW'15]: P. Hubacek and D. Wichs. "On the Communication Complexity of Secure Function Evaluation with Long Output."

[DHRW'16]: D. Dodis, S. Halevi, R. D. Rothblum, and D. Wichs. "Spooky Encryption and Its Applications."

[QWW'18]: W. Quach, H. Wee, and D. Wichs. "Laconic Function Evaluation and Applications."

[ARS'24]: D. Abram, L. Roy, and P. Scholl. "Succinct Homomorphic Secret Sharing."

[AMR’25]: D. Abram, G. Malavolta, L. Roy. “Succinct Oblivious Tensor Evaluation and Applications: Adaptively-Secure Laconic Function Evaluation and Trapdoor Hashing for All Circuits.”

[BJSSS'25]: E. Boyle, A. Jain, S. Servan-Schreiber, and A. Srinivasan. "Simultaneous-Message and Succinct Secure Computation."