Communication-Efficient Private Join and Compute over Distributed Input Sets

Yunqing Sun

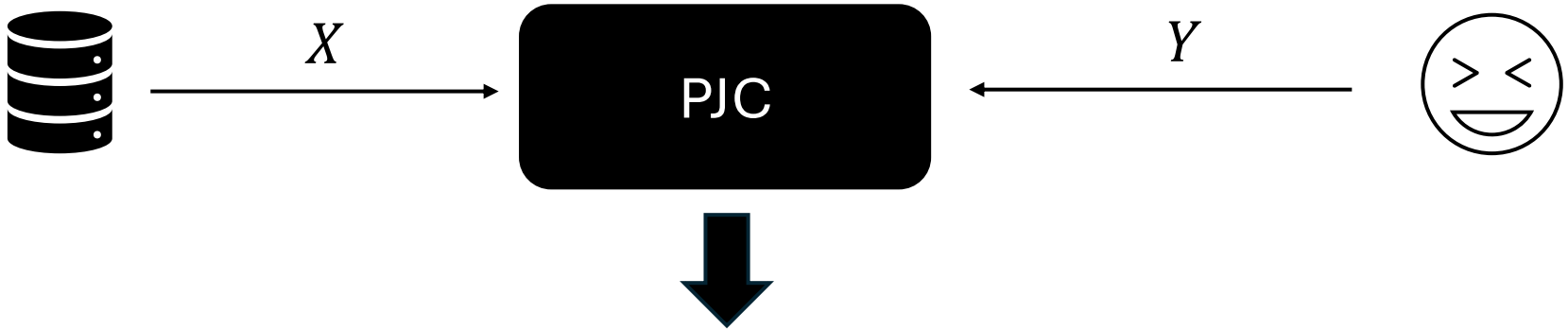
Northwestern University

Joint work with Hanlin Liu, Xinran Cai, Xiao Wang, and Wei Dong

Presented on 2026-October-27 at WPEC 2026

NIST Workshop on Privacy-Enhancing Cryptography 2026

Private Join and Compute (PJC)



Computation over Intersection:
Ex. sum, inner product, cardinality, etc.

Advertisement Conversion with PJC



TRANSACTION DATA PROVIDERS

ADVERTISEMENT VENDORS

Cust_id	Spending Amount
1	$50
2	$10
3	$100
4	$20
5	$70

X

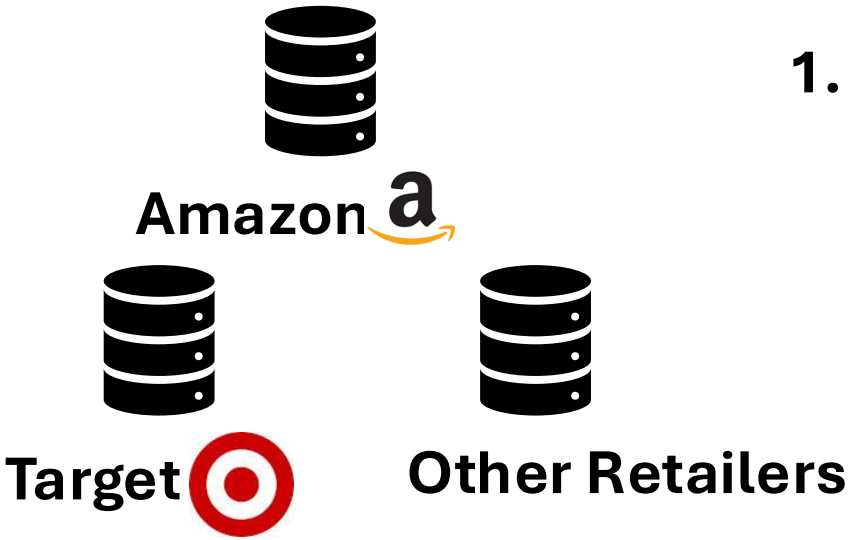
Weighted advertising conversion

$$\Sigma \text{ Spending(id)} \times \frac{1}{\text{Duration(id)}}$$

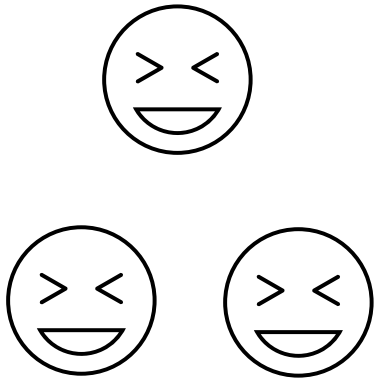
Cust_id	Duration
1	300 s
3	200 s
7	30 s
2	30 s
6	20 s

Y

PJC over Distributed Inputs



1. Pair-wise 2-party PJC
   - Leaks intermediate info
   - Poorly scalable



TRANSACTION DATA PROVIDERS

ADVERTISEMENT VENDORS

Cust_id	Spending Amount
1	$50
2	$10

Cust_id	Spending Amount
3	$100
4	$20
5	$70

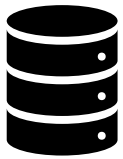
Cust_id	Duration
1	300 s
3	200 s

Cust_id	Duration
7	30 s
2	30 s
6	20 s

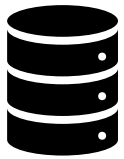
PJC over Distributed Inputs



Amazon



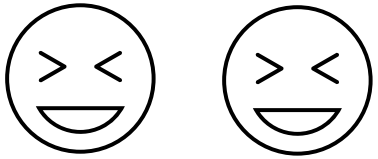
Target



Other Retailers

1. Pair-wise 2-party PJC
   - Leaks intermediate info
   - Poorly scalable
2. Server-based generic 2-Party Computation (2PC)
   - Additively share each cell





TRANSACTION DATA PROVIDERS

Cust_id	Spending Amount
1	$50
2	$10

Cust_id	Spending Amount
3	$100
4	$20
5	$70

ADVERTISEMENT VENDORS

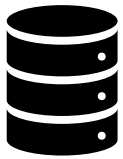
Cust_id	Duration
1	300 s
3	200 s

Cust_id	Duration
7	30 s
2	30 s
6	20 s

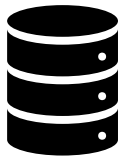
PJC over Distributed Inputs



Amazon



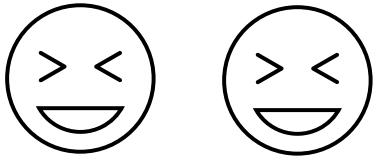
Target



Other Retailers

1. Pair-wise 2-party PJC
   - Leaks intermediate info
   - Poorly scalable
2. Server-based generic 2-Party Computation (2PC)
   - Additively share each cell
   - Each share to one server





TRANSACTION DATA PROVIDERS

ADVERTISEMENT VENDORS

Cust_id	Spending Amount
3	$100
4	$20
5	$70

→

Cust_id	Spending Amount
1	$38
3	$19
3	$15



S1

+

Cust_id	Spending Amount
2	$62
1	$1
2	$55

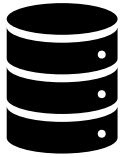


S2

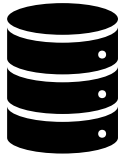
PJC over Distributed Inputs



Amazon

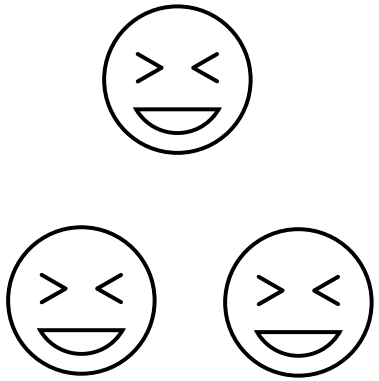


Target



Other Retailers

1. Pair-wise 2-party PJC
   - Leaks intermediate info
   - Poorly scalable
2. Server-based generic 2-Party Computation (2PC)
   - Additively share each cell
   - Each share to one server





S1

Partial Provider Share

Partial Vendor Share

2PC

Partial Provider Share

Partial Vendor Share



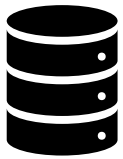
S2

Recovered inner product

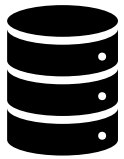
PJC over Distributed Inputs



Amazon



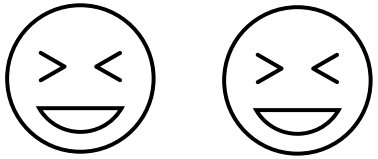
Target

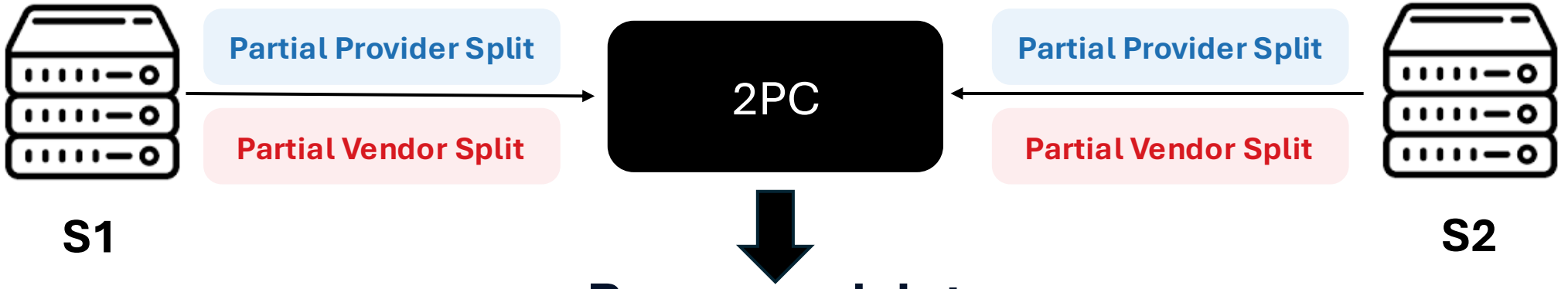


Other Retailers

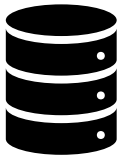
1. Pair-wise 2-party PJC
   - Leaks intermediate info
   - Poorly scalable
2. Server-based generic 2-Party Computation (2PC)
   - Additively share each cell
   - Each share to one server
   - Efficiency issue







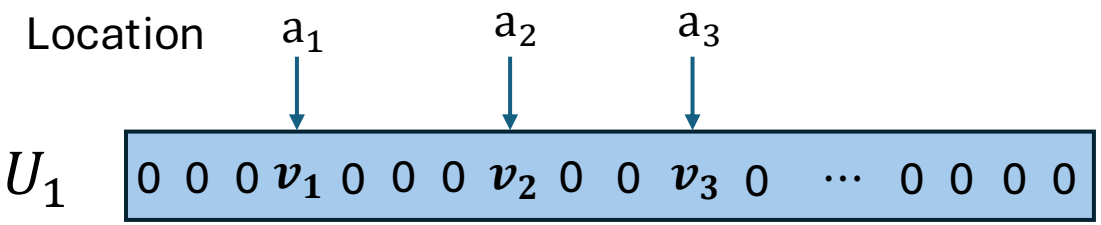
Solution: Vector Representation and Approximation

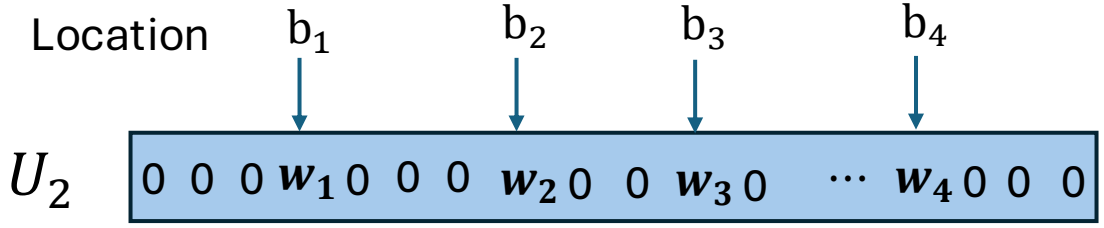


element	payload
a_1	v_1
a_2	v_2
a_3	v_3

element	payload
b_1	w_1
b_2	w_2
b_3	w_3
b_4	w_4







Observation

1. Set $v_i = w_i = 1$, compute PSI cardinality
2. Set v_i =1, compute PSI summation
3. Set v_i, w_i, compute PSI inner product

Method for Efficiency and Security

1. Sketch algorithm for data stream
2. Secure approximation

AMS Sketch

Let $\boldsymbol{x}, \boldsymbol{y} \in Z^N, \boldsymbol{r} \in \{-1, +1\}^N$ be a random four-wise independent vector.

Define $Z_x = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{x}[i],\ Z_{\mathrm{y}} = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{y}[i]$.

Then $\mathrm{E}[Z_x \cdot Z_y] = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ and $\mathrm{Var}[Z_x \cdot Z_y] = \mathrm{O}(\|\boldsymbol{x}\|_2^2 \|\boldsymbol{y}\|_2^2)$

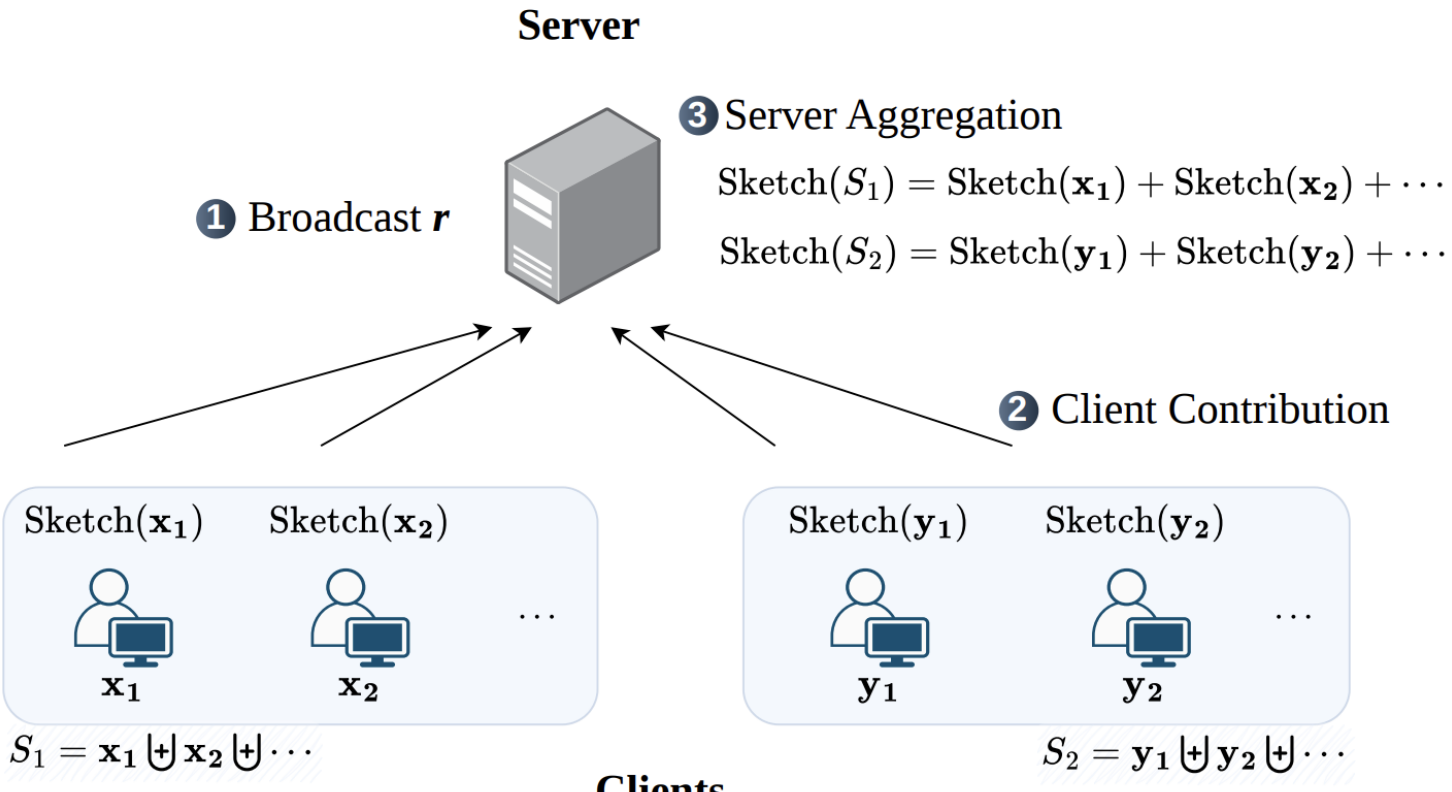
Define group $S_1 = \boldsymbol{x_1} \uplus \boldsymbol{x_2} \uplus \cdots$, group $S_2 = \boldsymbol{y_1} \uplus \boldsymbol{y_2} \uplus \cdots$, then $Z_{S_1} = Z_{\boldsymbol{x_1}} + Z_{\boldsymbol{x_2}} + \cdots$ and $Z_{S_2} = Z_{\boldsymbol{y_1}} + Z_{\boldsymbol{y_2}} + \cdots$.

AMS Sketch

Let $\boldsymbol{x}, \boldsymbol{y} \in Z^N, \boldsymbol{r} \in \{-1, +1\}^N$ be a random four-wise independent vector.

Define $Z_x = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{x}[i],\ Z_y = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{y}[i]$.

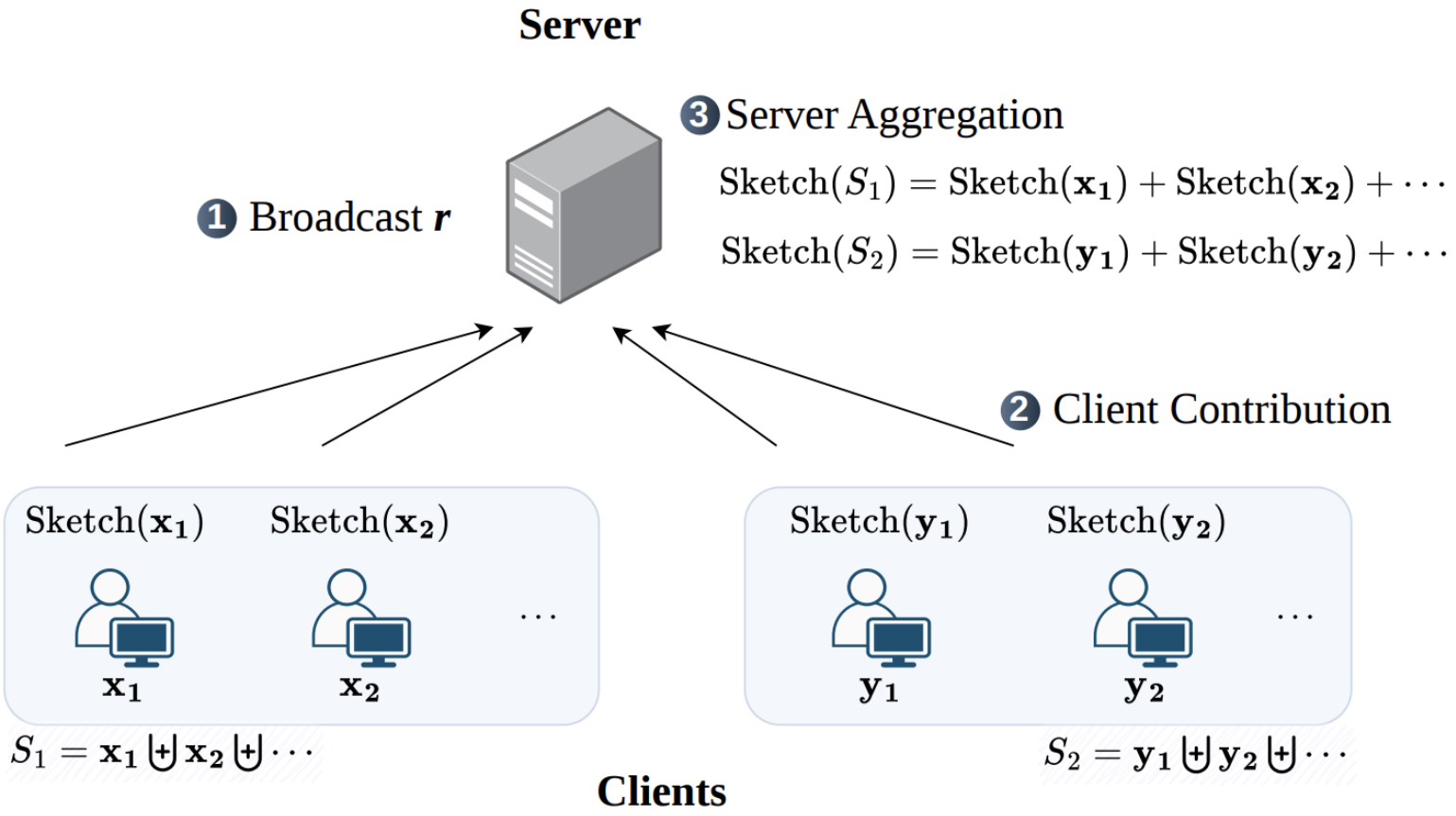
Then $\mathrm{E}[Z_x \cdot Z_y] = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ and $\mathrm{Var}[Z_x \cdot Z_y] = \mathrm{O}(\|\boldsymbol{x}\|_2^2 \|\boldsymbol{y}\|_2^2)$



Plaintext randomness does not ensure privacy of private sets!

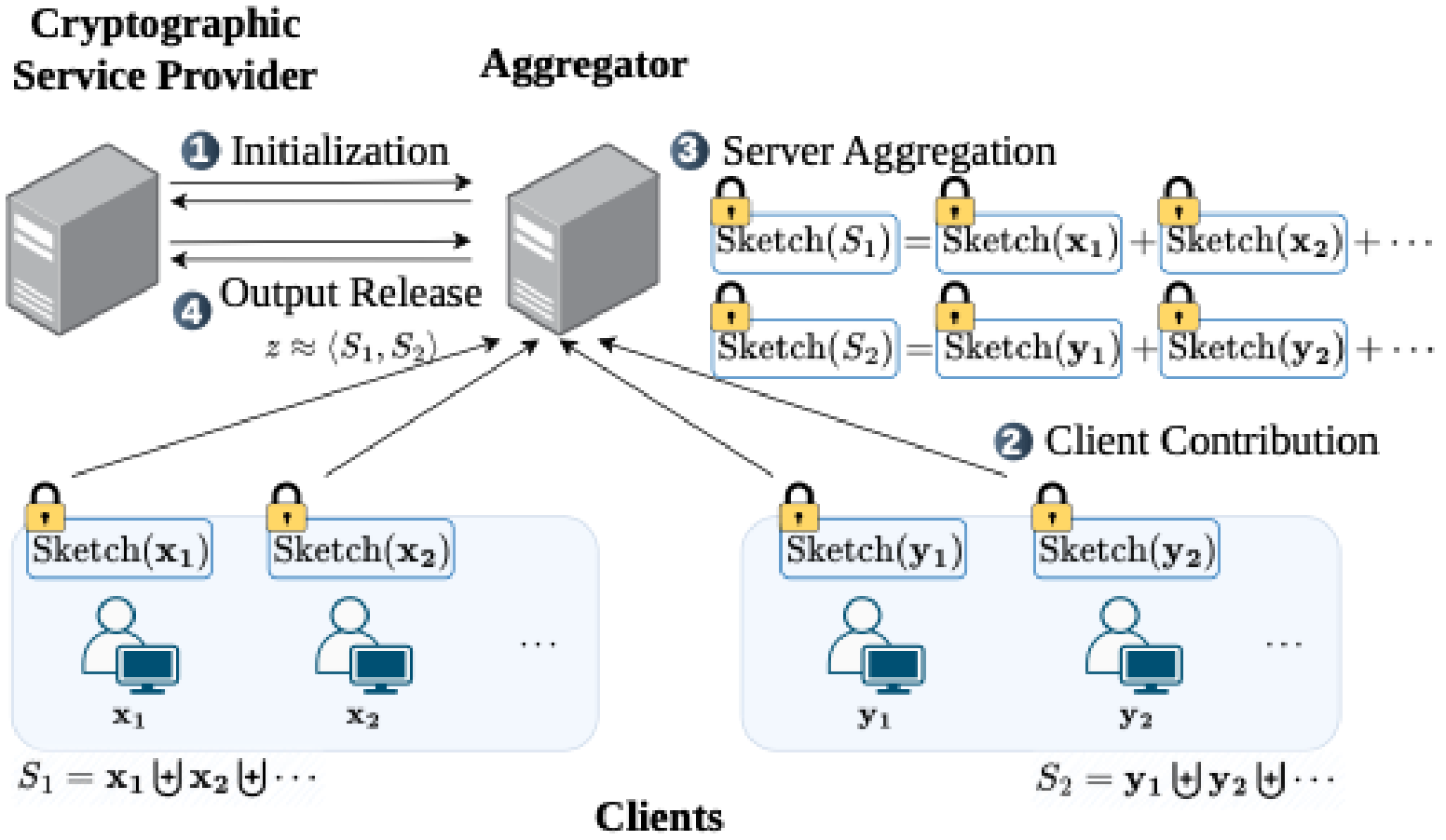
General Framework with Privacy (in HE)

For Privacy: We need *Homomorphic Encryption* of $\boldsymbol{r}$ ($[\![\boldsymbol{r}]\!]$)
For Communication-efficiency: We need to send a short seed $[\![\boldsymbol{s}]\!]$, *s.t.* $[\![\boldsymbol{r}]\!] = \boldsymbol{G}([\![\boldsymbol{s}]\!])$.



General Framework with Privacy (in HE)

For Privacy: We need *Homomorphic Encryption* of $\boldsymbol{r}$ ($[\![\boldsymbol{r}]\!]$)
For Communication-efficiency: We need to send a short seed $[\![\boldsymbol{s}]\!]$, *s.t.* $[\![\boldsymbol{r}]\!] = \boldsymbol{G}([\![\boldsymbol{s}]\!])$.
Key Issue: Can we find a $\boldsymbol{G}(\cdot)$ that is *computational efficient in HE?*



AMS Sketch (four-wise independent)

Let $\boldsymbol{x}, \boldsymbol{y} \in Z^N, \boldsymbol{r} \in \{-1, +1\}^N$ be a random four-wise independent vector.

Define $Z_x = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{x}[i],\ Z_y = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{y}[i]$.

Then $\mathrm{E}[Z_x \cdot Z_y] = \langle \mathrm{x}, \mathrm{y} \rangle$ and $\mathrm{Var}[Z_x \cdot Z_y] = \mathrm{O}(\|x\|_2^2 \|y\|_2^2)$

Generation of $\boldsymbol{r} \in \{-\mathbf{1}, +\mathbf{1}\}^{\boldsymbol{N}}$: we generate $\boldsymbol{r}[i]$ for each $\boldsymbol{x}[i] \neq 0$

Classical Instantiation of four-wise independent: $G: [N]^4 \to \{0,1\}^N$

$$G(a,b,c,d)[i] = (a \cdot i^3 + b \cdot i^2 + c \cdot i + d) \bmod p \bmod 2$$

Modular operation from Z_p to Z_2 is expensive in HE

Adapted AMS Sketch (Δ-inadmissible distribution)

Let $\boldsymbol{x}, \boldsymbol{y} \in Z^N, \boldsymbol{r} \leftarrow \boldsymbol{\mathcal{D}}$, where $\boldsymbol{\mathcal{D}}$ is a **Δ-inadmissible distribution** over $\{-1,+1\}^N$.

Define $Z_x = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{x}[i]$, $Z_y = \sum_{i\in[N]} \boldsymbol{r}[i]\boldsymbol{y}[i]$.

Then $\mathrm{E}[Z_x \cdot Z_y] = \langle x, y\rangle$ and $\mathrm{Var}[Z_x \cdot Z_y] \le \Delta(\mathcal{D})\|x\|_2^2\|y\|_2^2$

Generation of $\boldsymbol{r}$: log log n-inadmissible

Sample $\boldsymbol{s} \leftarrow \{0,1\}^n$, and choose random $\boldsymbol{A} \in \{0,1\}^{N\times n}$, where $HW(\boldsymbol{A_i}) = d, d > 1, n \ge 4d, N \le n^{d/2}, \boldsymbol{A_i} \neq \boldsymbol{A_j}\ for\ all\ i \neq j,$ and compute $\boldsymbol{r} = (-1)^{\boldsymbol{A}\cdot\boldsymbol{s} \bmod 2}$

$\boldsymbol{A_i}$

0	1	0	1	0	1	0	0	1
1	0	1	1	0	0	0	1	0
0	1	0	0	1	0	1	0	1
1	0	0	0	0	1	0	1	1
0	0	1	1	0	0	1	0	1

x $\boldsymbol{s}$

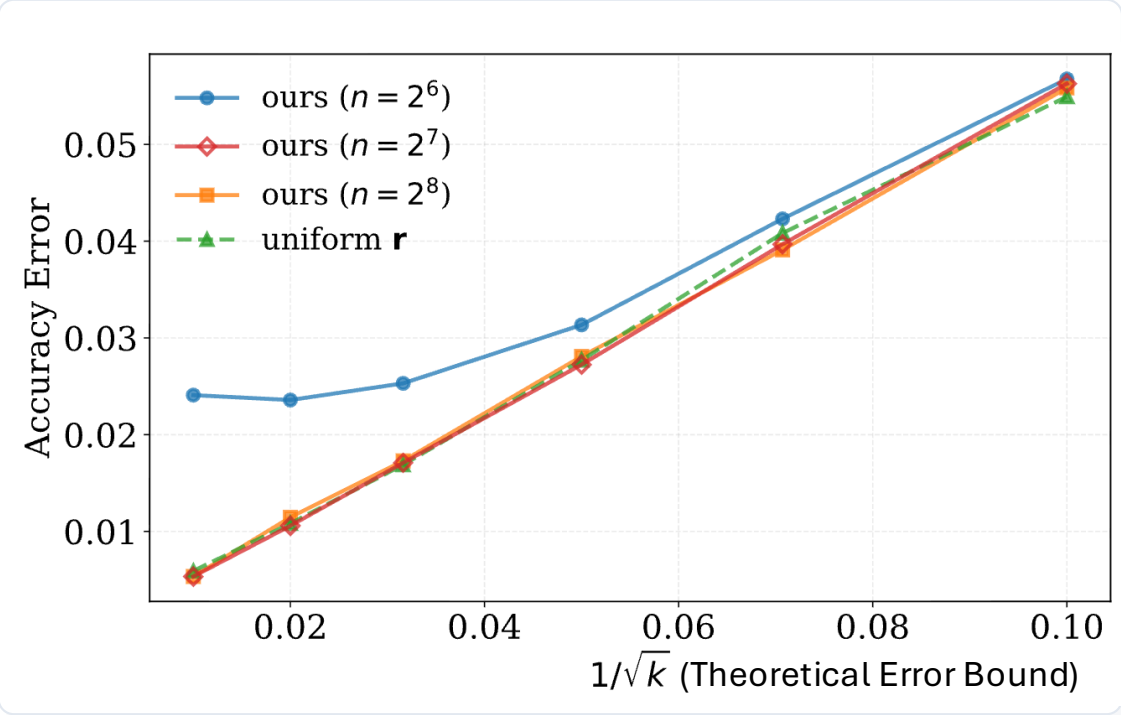
0
0
1
1
0
1
0
0
1

$\bmod 2$

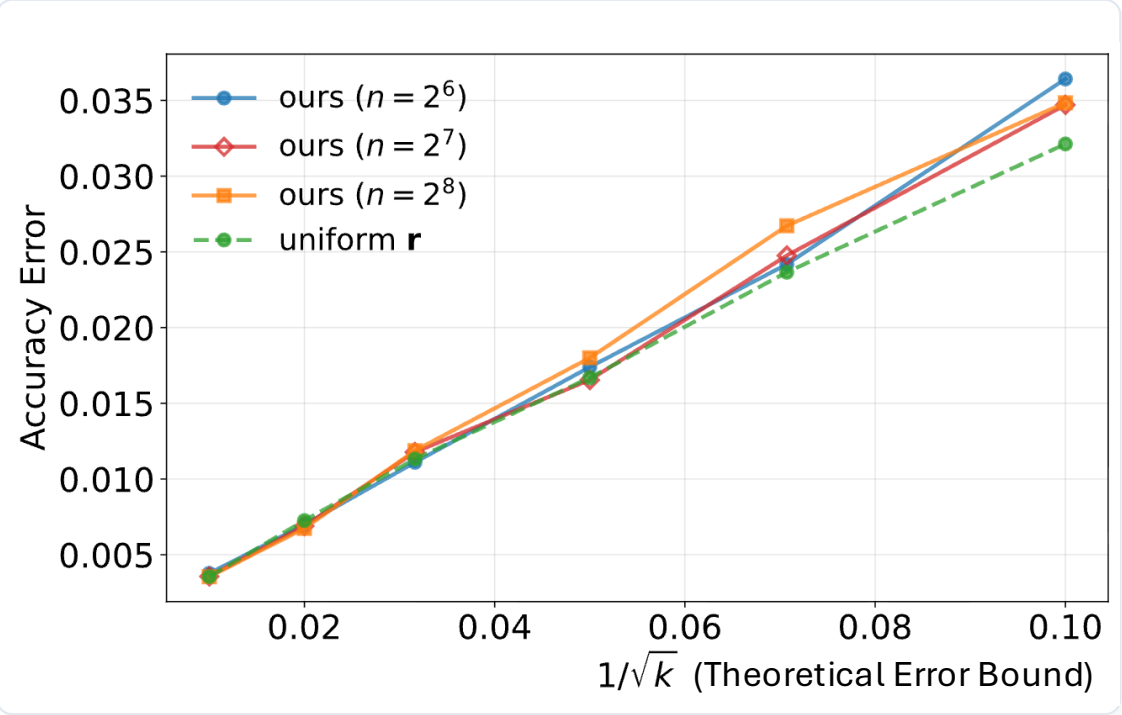
- On-demand generation:
  - generate $\boldsymbol{r}[i]$ for each $x[i] \neq 0$
  - *Pick $\boldsymbol{A_i}$ as needed*
- HE friendly evaluation:
  - generate $\boldsymbol{r}[i]$ needs d-1 mul.

Accuracy of Adapted AMS Sketch

Empirical accuracy error comparison (Tested on uniform data)



Empirical accuracy error comparison (Tested on UCI bag of words)



Performance of PJC Approximation

Server Init. : 2 HE scalar multiplication
Client Local Sketch.: choose $d \leq 8$, at most 3 level of HE multiplication
Server Agg. and Output: Threshold Dec and 2PC computation

Measured cost by phase

Phase	Computation	Communication	Publish size
Server Init (pre-processing).	12.12 s	419 MB	420 MB
Client Local Sketch (offline).	30 ms / elem.	0.25 MB / client	—
Server Agg. and Output	0.97 s	15.5 MB	—

Thank you!

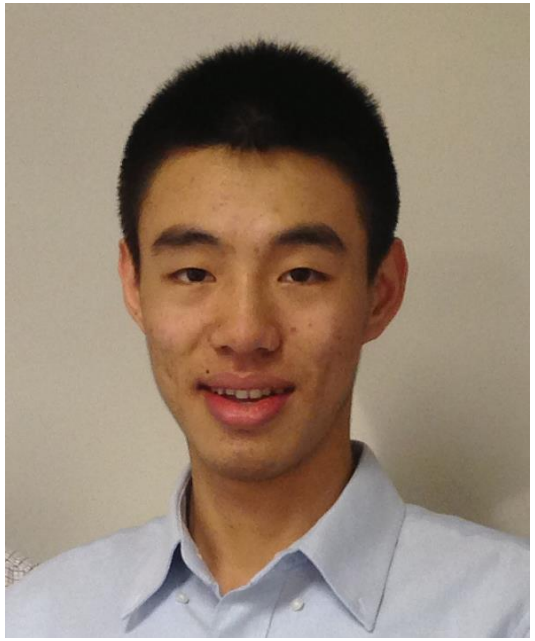
Acknowledgement



Xinran Cai



Hanlin Liu



Xiao Wang



Wei Dong