Shuffle-based Private Set Union: Faster and More Secure

Yanxue Jia, Shi-Feng Sun, Hong-Sheng Zhou, Jiajun Du, Dawu Gu

饮 水 思 源 • 爱 国 荣 校

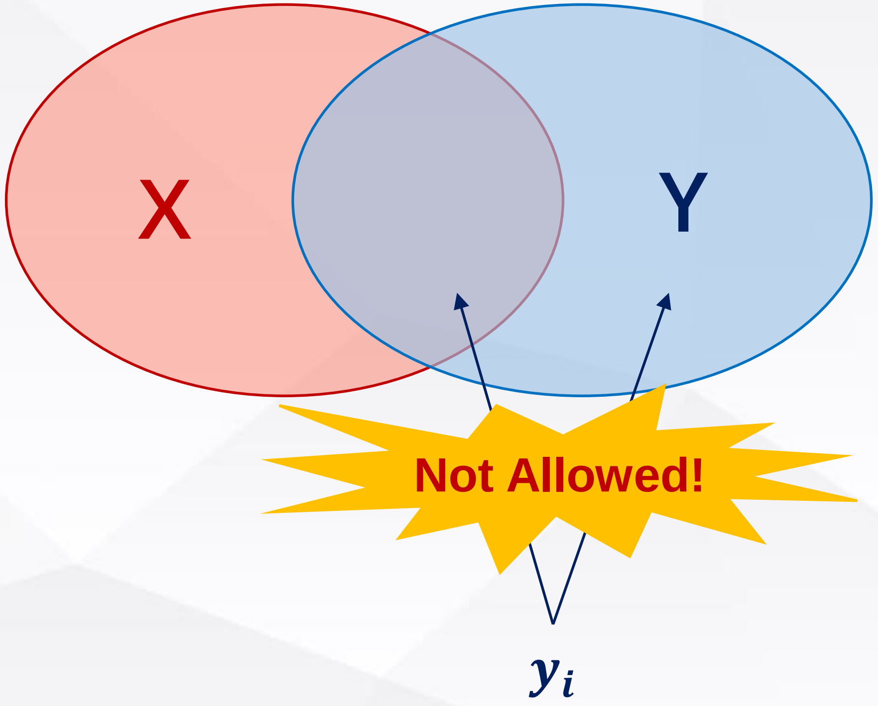
Contents

- **Private Set Union (PSU)**
- **Our Contributions**
- **Our Main Ideas**
- **Performance Comparison**

Private Set Union (PSU)

Sender (X)

knows nothing.



Receiver (Y)

Obtains $\mathrm{X} \cup Y$,
but knows nothing about $\mathrm{X} \cap Y$.

Semi-honest setting

Our Contributions

Public Key	Symmetric Key
[KS05], [Fri07], [DC17]	[KRTW19] **Ours**

- Point out a security issue incurred by the bucketing technique;
- Design two PSU protocols based on symmetric key operations without using the bucketing technique;
- Consider unbalanced datasets;
- Perform a comprehensive evaluation & comparison.

[KS05] Lea Kissner and Dawn Xiaodong Song. Privacy-preserving set operations. In Victor Shoup, editor, CRYPTO 2005, volume 3621 of LNCS, pages 241–257.

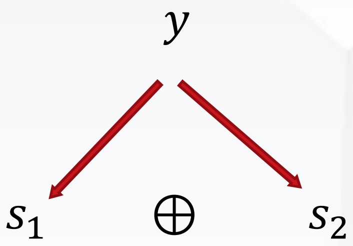
[Fri07] Keith B. Frikken. Privacy-preserving set union. In Jonathan Katz and Moti Yung, editors, ACNS 07, volume 4521 of LNCS, pages 237–252.

[DC17] Alex Davidson and Carlos Cid. An efficient toolkit for computing private set operations. In Josef Pieprzyk and Suriadi Suriadi, editors, ACISP 17, Part II, volume 10343 of LNCS, pages 261–278.

[KRTW19] Vladimir Kolesnikov, Mike Rosulek, Ni Trieu, and Xiao Wang. Scalable private set union from symmetric-key techniques. In Steven D. Galbraith and Shiho Moriai, editors, ASIACRYPT 2019, Part II, volume 11922 of LNCS, pages 636–666.

Our Main Ideas

A simple fact



Given x:

If $x = y$: $x \oplus s_1 = s_2$
Else: $x \oplus s_1 \neq s_2$

Our Main Ideas

- $(1, n)$-PSU

Sender (x)

Receiver (Y)

Change

For each $y_i \in Y$:
    Share y_i to s_i^1 and s_i^2, s.t. $y_i = s_i^1 \oplus s_i^2$;
Set $S_1 = \{s_1^1, s_2^1, \cdots, s_n^1\}$ and $S_2 = \{s_1^2, s_2^2, \cdots, s_n^2\}$;

$\xleftarrow{\quad S_1 \quad}$

Obtain $I = \{s_1^1 \oplus x, s_2^1 \oplus x, \cdots, s_n^1 \oplus x\}$;

If $x = y_i$, $s_i^1 \oplus x = s_i^2$

$\xrightarrow{\quad I \quad}$

Check if $I \cap S_2 \neq \emptyset$,
    if so, $x \in Y$, set $b = 1$;
    else, $x \notin Y$, set $b = 0$;

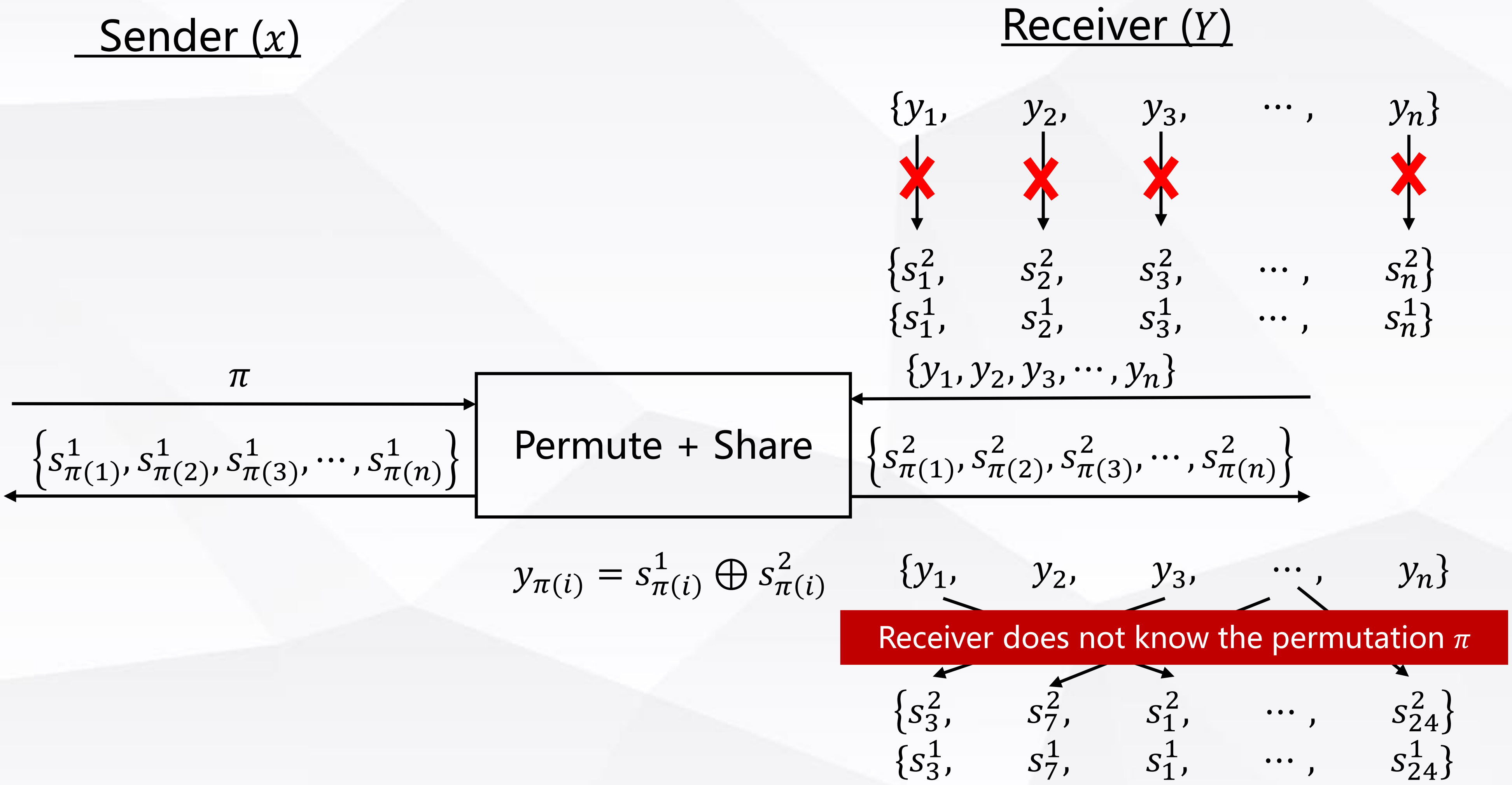
If $b = 0$: $x = (s_1^1 \oplus x) \oplus s_1^1$;
Else, ……

The receiver can know s_i^2 → The receive know $y_i \in X \cap Y$

Not Allowed!

Our Main Ideas

- $(1, n)$-PSU



Sender (x)

Receiver (Y)

$\{y_1, \quad y_2, \quad y_3, \quad \cdots, \quad y_n\}$

$\{s_1^2, \quad s_2^2, \quad s_3^2, \quad \cdots, \quad s_n^2\}$

$\{s_1^1, \quad s_2^1, \quad s_3^1, \quad \cdots, \quad s_n^1\}$

π

$\{y_1, y_2, y_3, \cdots, y_n\}$

Permute + Share

$\left\{s_{\pi(1)}^1, s_{\pi(2)}^1, s_{\pi(3)}^1, \cdots, s_{\pi(n)}^1\right\}$

$\left\{s_{\pi(1)}^2, s_{\pi(2)}^2, s_{\pi(3)}^2, \cdots, s_{\pi(n)}^2\right\}$

$$y_{\pi(i)} = s_{\pi(i)}^1 \bigoplus s_{\pi(i)}^2$$

$\{y_1, \quad y_2, \quad y_3, \quad \cdots, \quad y_n\}$

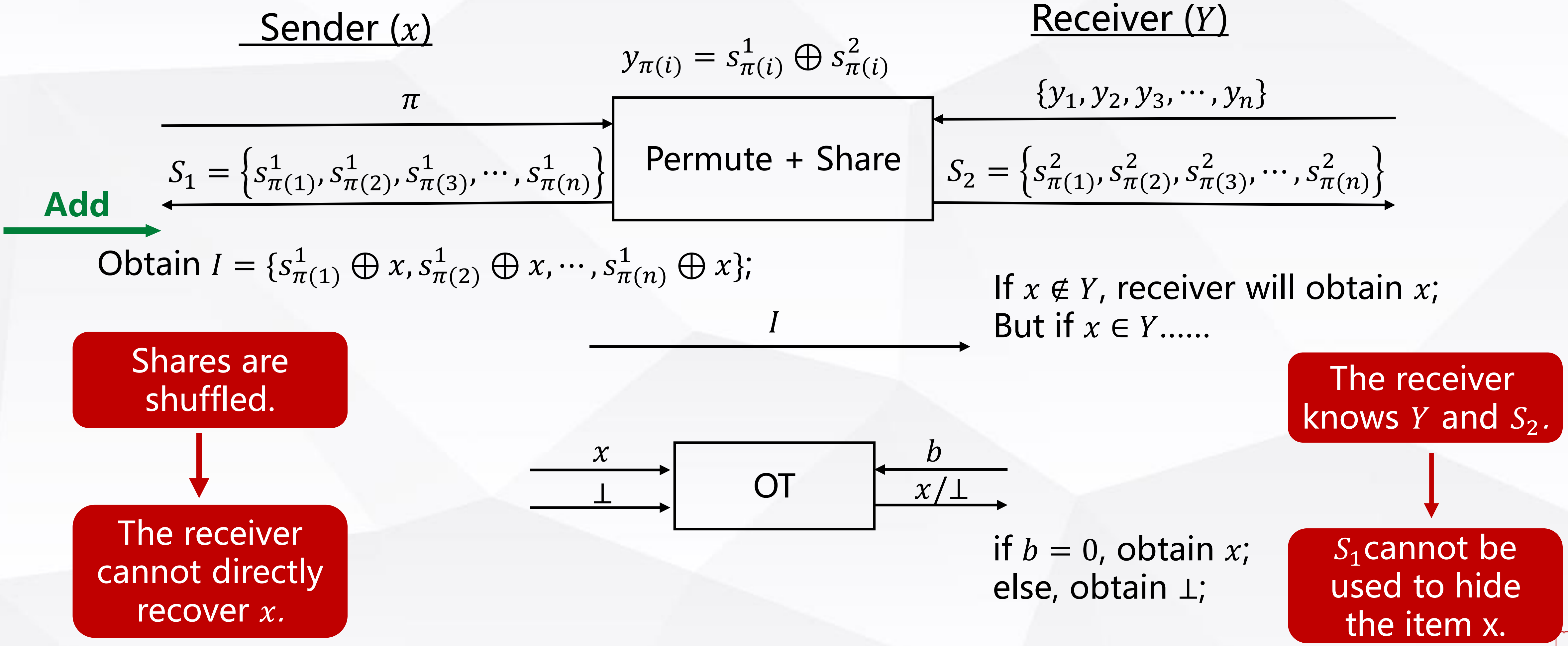
Receiver does not know the permutation π

$\{s_3^2, \quad s_7^2, \quad s_1^2, \quad \cdots, \quad s_{24}^2\}$

$\{s_3^1, \quad s_7^1, \quad s_1^1, \quad \cdots, \quad s_{24}^1\}$

Our Main Ideas

- $(1, n)$-PSU



Sender (x)

Receiver (Y)

$y_{\pi(i)} = s^1_{\pi(i)} \oplus s^2_{\pi(i)}$

π

$\{y_1, y_2, y_3, \cdots, y_n\}$

Permute + Share

$S_1 = \left\{s^1_{\pi(1)}, s^1_{\pi(2)}, s^1_{\pi(3)}, \cdots, s^1_{\pi(n)}\right\}$

$S_2 = \left\{s^2_{\pi(1)}, s^2_{\pi(2)}, s^2_{\pi(3)}, \cdots, s^2_{\pi(n)}\right\}$

Add

Obtain $I = \{s^1_{\pi(1)} \oplus x, s^1_{\pi(2)} \oplus x, \cdots, s^1_{\pi(n)} \oplus x\}$;

I

If $x \notin Y$, receiver will obtain x;
But if $x \in Y$……

Shares are shuffled.

The receiver cannot directly recover x.

x

$\perp$

OT

b

$x/\perp$

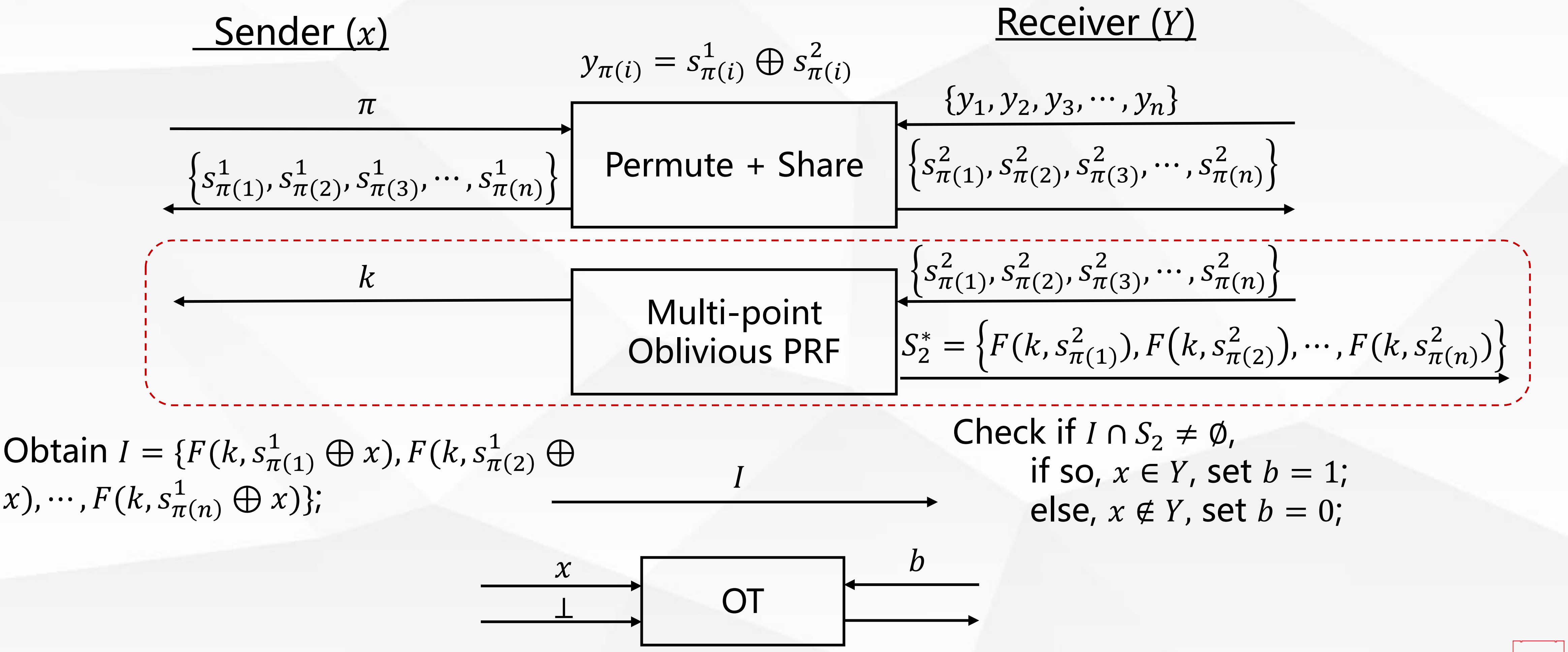
if $b = 0$, obtain x;
else, obtain $\perp$;

The receiver knows Y and S_2.

S_1cannot be used to hide the item x.

Our Main Ideas

- $(1, n)$-PSU



Sender (x)

Receiver (Y)

$y_{\pi(i)} = s^1_{\pi(i)} \oplus s^2_{\pi(i)}$

π

$\{y_1, y_2, y_3, \cdots, y_n\}$

Permute + Share

$\left\{s^1_{\pi(1)}, s^1_{\pi(2)}, s^1_{\pi(3)}, \cdots, s^1_{\pi(n)}\right\}$

$\left\{s^2_{\pi(1)}, s^2_{\pi(2)}, s^2_{\pi(3)}, \cdots, s^2_{\pi(n)}\right\}$

k

$\left\{s^2_{\pi(1)}, s^2_{\pi(2)}, s^2_{\pi(3)}, \cdots, s^2_{\pi(n)}\right\}$

Multi-point Oblivious PRF

$S_2^* = \left\{F(k, s^2_{\pi(1)}), F\left(k, s^2_{\pi(2)}\right), \cdots, F(k, s^2_{\pi(n)})\right\}$

Obtain $I = \{F(k, s^1_{\pi(1)} \oplus x), F(k, s^1_{\pi(2)} \oplus x), \cdots, F(k, s^1_{\pi(n)} \oplus x)\}$;

I

Check if $I \cap S_2 \neq \emptyset$,
if so, $x \in Y$, set $b = 1$;
else, $x \notin Y$, set $b = 0$;

x

$\perp$

OT

b

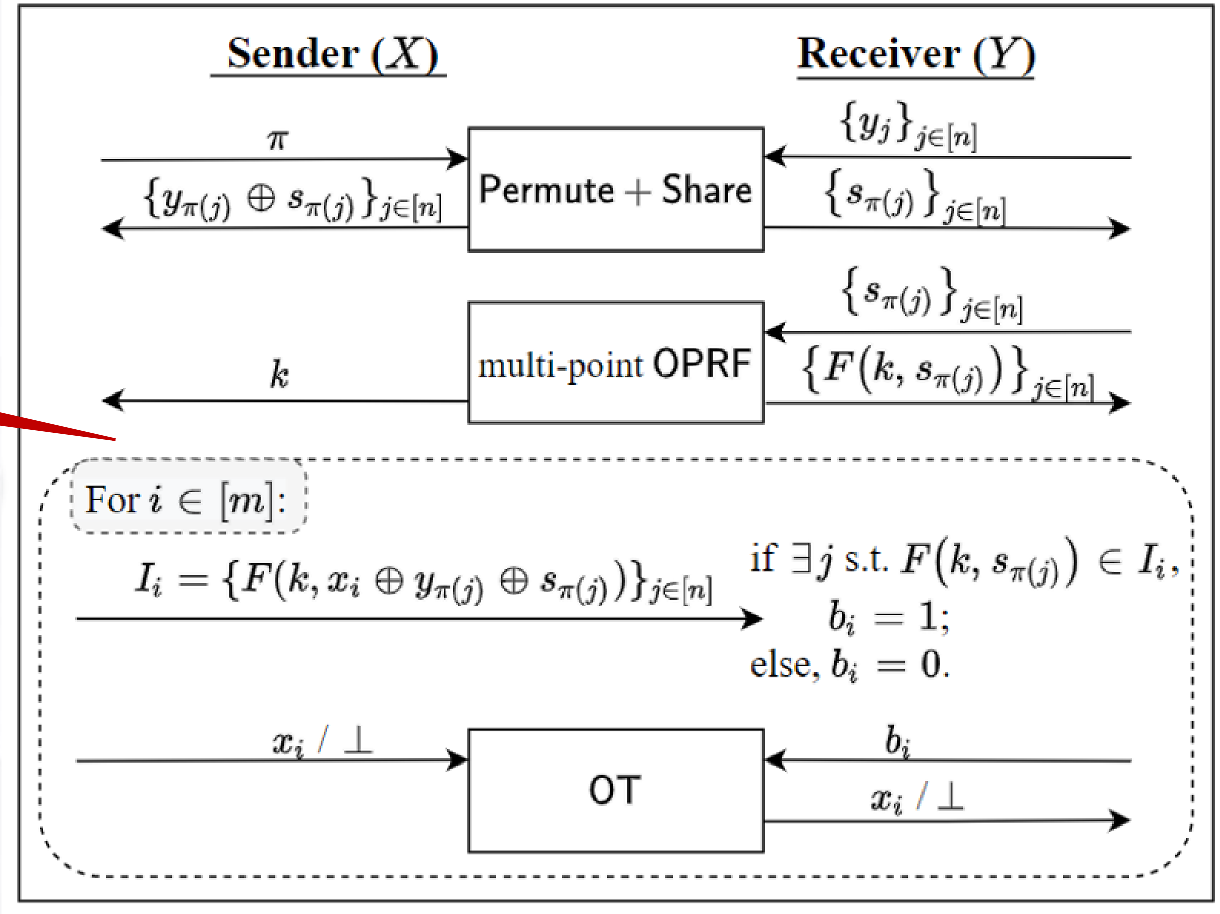
Our Main Ideas

- (m, n)-PSU

For each $x_i \in X$, generate a I_i

Computation and communication costs are both $O(mn)$!

Need to optimize the basic scheme!



Our Main Ideas

- (m, n)-PSU

Goal: to reduce the number of items in each set I_i.

$$I_i = \{F(k, s^1_{\pi(1)} \oplus x_i), F(k, s^1_{\pi(2)} \oplus x_i), \cdots, F(k, s^1_{\pi(n)} \oplus x_i)\};$$

From the sender's point of view, any item in Y may be equal to x_i.

Key idea: to reduce the number of "candidate" items in Y that may be equal to x_i.

Use Cuckoo hashing

Design of Our Protocols

Optimization via Cuckoo hashing

Only y_1 and y_3 may be equal to x_i.

↓

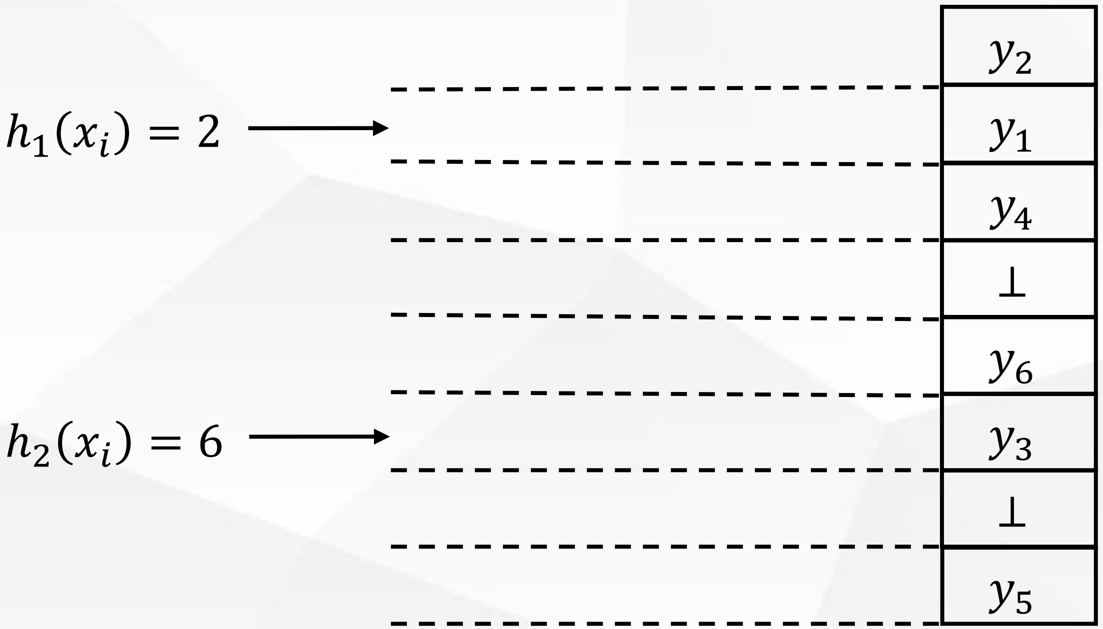
Only need to use the shares of y_1 and y_3 to generate I_i.

Sender (X)

For x_i:

Receiver (Y)

Insert Y into the Cuckoo hash table parameterized by $h_1()$ and $h_2()$.

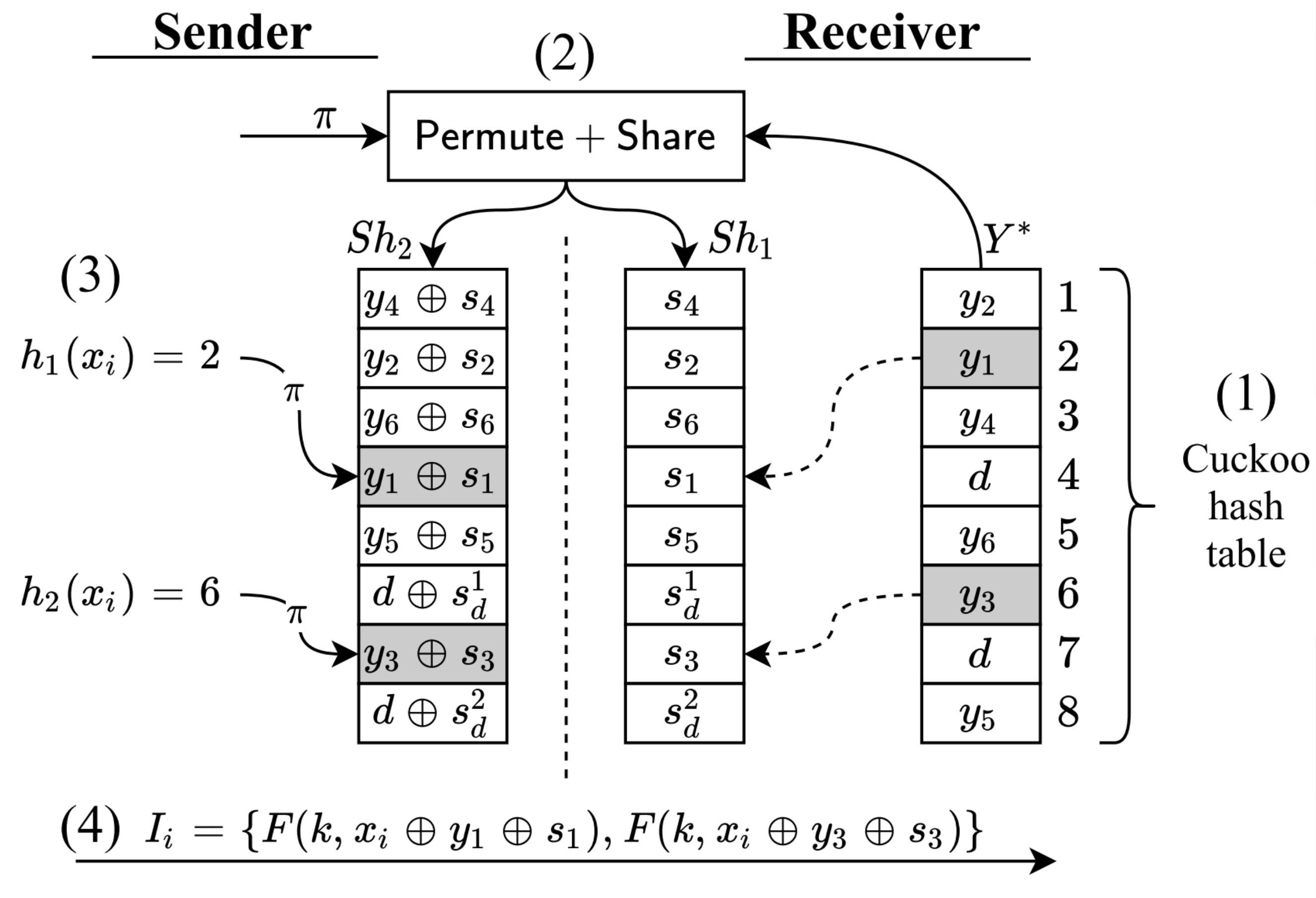


$h_1(x_i) = 2$

$h_2(x_i) = 6$

Our Main Ideas

Optimization via Cuckoo hashing



Our Main Ideas

A dual version

Shuffle the sender's set X

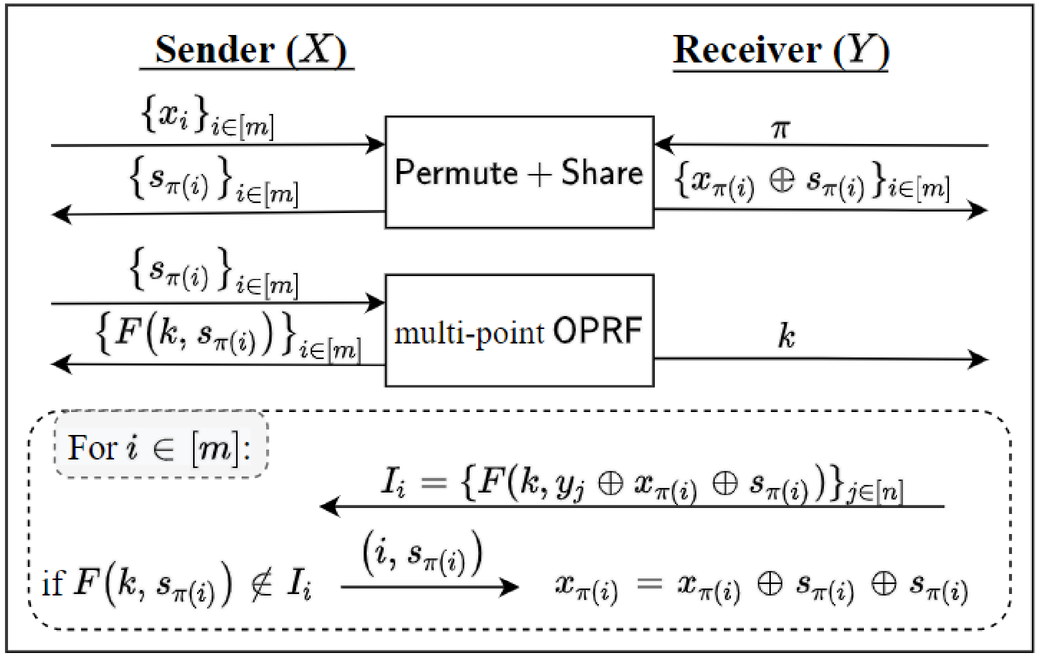


Fig. 6. Core idea of $\Pi^{\mathsf{S}}_{\mathsf{PSU}}$ for (m, n)-PSU.

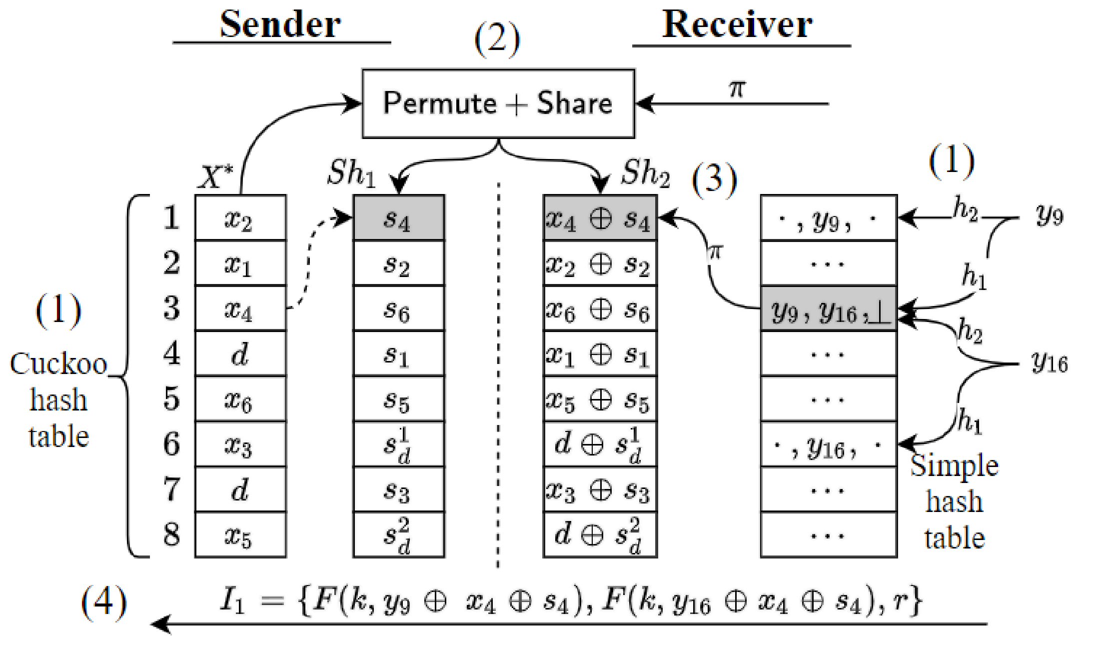


Fig. 7. $\Pi^{\mathsf{S}}_{\mathsf{PSU}}$: Optimization via Cuckoo hashing.

Performance Comparison

			2^{18}	2^{20}	2^{22}
Time (s)	WAN	[KRTW19]	86.358	333.037	1459.539
		Ours	**16.104**	**67.756**	**341.758**
	LAN	[KRTW19]	69.19	263.476	1191.703
		Ours	**10.751**	**48.703**	**251.091**
Comm. (MB)		[KRTW19]	600.62	2470.11	10233.28
		Ours	**307.192**	**1338.79**	**5779.599**

≈ 2 × slower than ours

[GMR+21] Gayathri Garimella, Payman Mohassel, Mike Rosulek, Saeed Sadeghian, and Jaspal Singh. Private set operations from oblivious switching. In Juan A. Garay, editor, Public-Key Cryptography – PKC 2021, pages 591-617.

Thanks

Q & A

https://eprint.iacr.org/2022/157

jiayanxue@sjtu.edu.cn