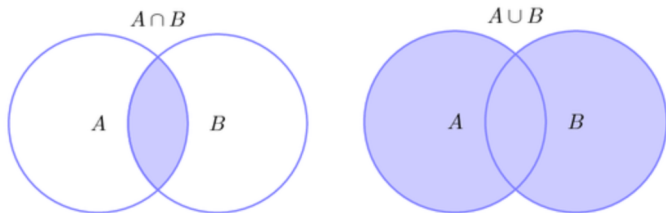


Extension of Two-Party Private Set Operations



Yu Chen
Shandong University

Outline

1 PSO in Unbalanced Setting

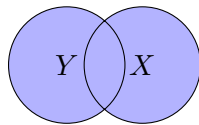
2 PSO in Multi-Party Setting

Outline

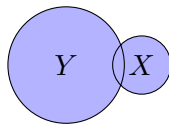
1 PSO in Unbalanced Setting

2 PSO in Multi-Party Setting

Motivation of Unbalanced Setting



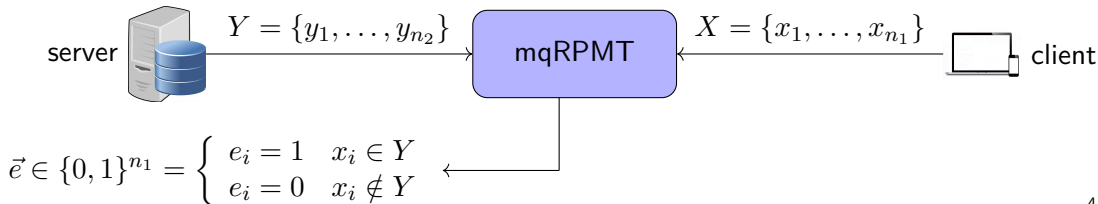
$n_2 \approx n_1$
balanced case



$n_2 \gg n_1$
unbalanced case

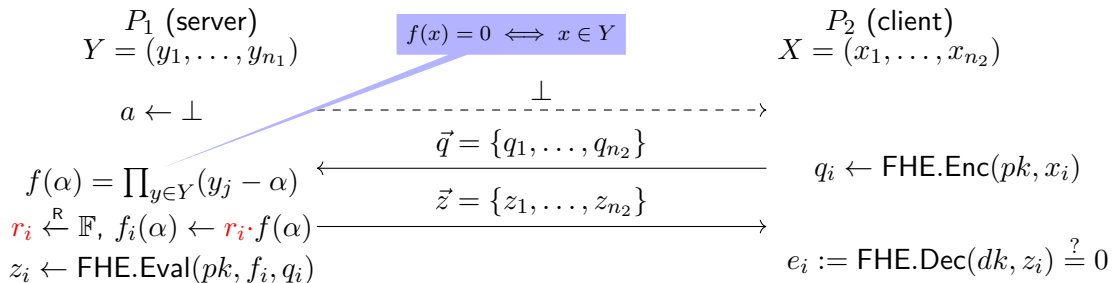
PSO (mqRPMT) designed for balanced setting are not efficient in unbalanced setting, particularly when n_2 is huge (communication cost scales linearly in both n_1 and n_2).

- **Goal:** build mqRPMT whose communication complexity is linear in n_1 but sublinear in n_2



Prior Work in PSI

The backbone Sigma mqPMT protocol
underlies unbalanced PSI [CLR17, CHLR18, CMdG⁺21]



Key idea: use multiplicative masking to hide $Y \setminus X$, and enable client to test

- communication cost: $2n_2$ FHE ciphertext.
- computation cost: n_1 multiplication in \mathbb{F} + $O(n_2 \log n_1)$ FHE evaluation

Unbalanced mqRPMT from FHE

Directly tweaking Sigma mqPMT to mqRPMT only yields mqRPMT*
↪ leak intersection size to the client.



Binbin Tu, **Yu Chen**, Qi Liu, Cong Zhang

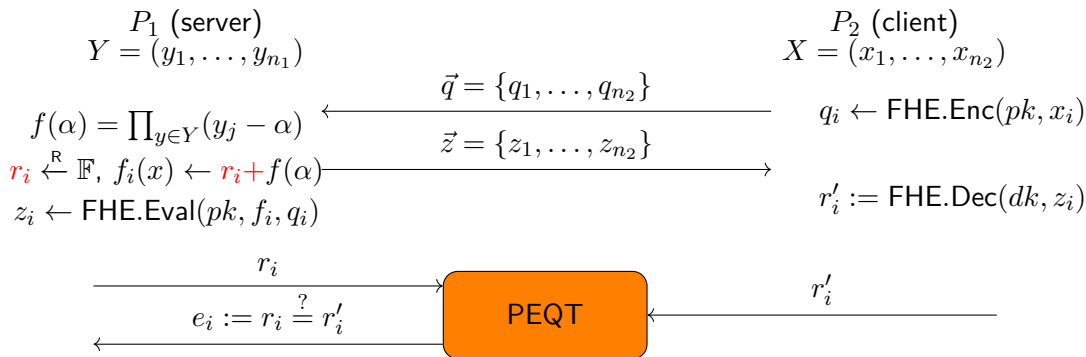
Fast Unbalanced Private Set Union from Fully Homomorphic Encryption

ACM CCS 2023

Technique: use different masking method

- additional optimizations are necessary but omit from the talk

Unbalanced mqRPMT from FHE (Oversimplified)



Key idea: use additive masking to hide Y , and disable client to test

- cost is roughly same as above plus PEQT cost

Unbalanced mqRPMT from BatchPIR

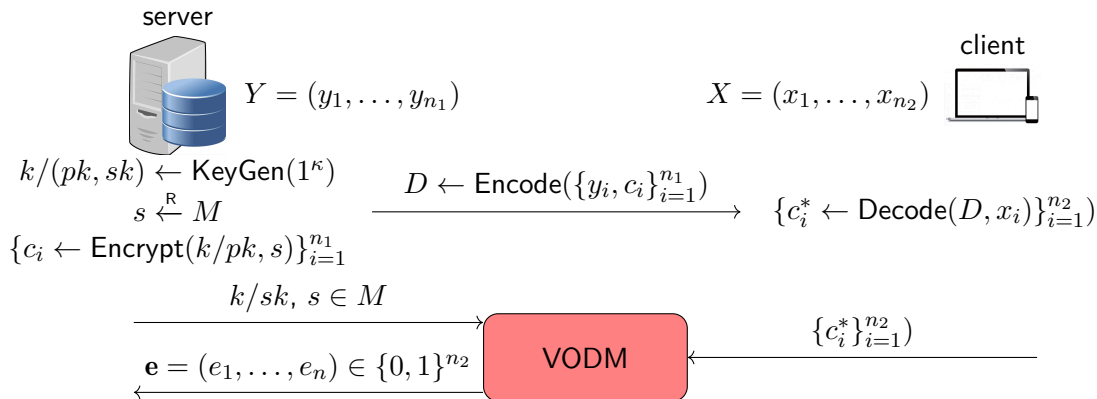
*The mqRPMT construction underlying [ZCL⁺23] is suitable for the balanced setting.
Can we adapt it to the unbalanced setting as well?*



Cong Zhang, **Yu Chen**, Weiran Liu, Liqiang Peng, Meng Hao, Anyu Wang,
Xiaoyun Wang

Unbalanced Private Set Union with Reduced Computation and Communication
ACM CCS 2024

mqRPMT from OKVS+Encryption+VODM: Revisited



Step 1: server (oblivious encoding) + client (oblivious decoding)

- communication scales linearly in $|D|$, which in turn linear in n_1

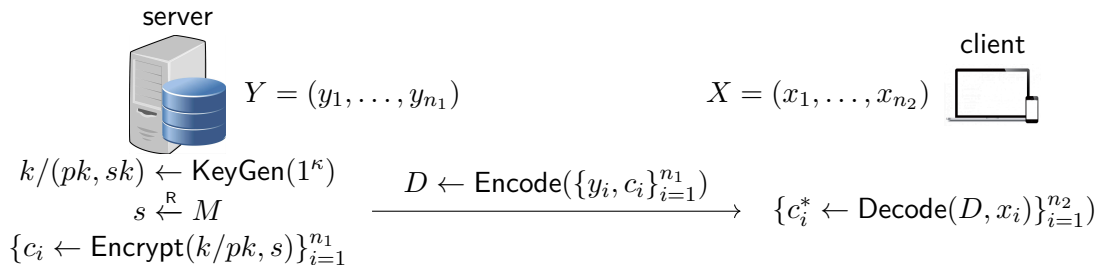
Step 2: server and client engage VODM

- communication scales linearly in n_2

Attain Sublinear Communication Complexity in Large Set

The communication complexity of step 2 is inherently linear in n_2 .

We focus on reducing the communication complexity of step 1.



Key Observation: The above approach achieves “oblivious” decoding by directly transmitting the entire D to the client, which is the root of linear complexity.

Can we achieve oblivious decoding without transmitting the entire D ?

Oblivious Key-Value Store

Existing SOTA OKVS schemes are binary linear OKVS

- a.k.a. the essence of Encode algorithm is solving the following linear equation:

$$\begin{bmatrix} -\text{row}(x_1) - \\ -\text{row}(x_2) - \\ \vdots \\ -\text{row}(x_n) - \end{bmatrix}_{n \times m} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (1)$$

where $\text{row} : \mathcal{K} \rightarrow \{0, 1\}^m$ is defined by the Encode algorithm and its random tape.

- The essence of Decode algorithm is computing subset sum of $D = (d_1, \dots, d_m)$:

$$\text{Decode}(D, x) = \langle \text{row}(x), D \rangle := \sum_{j=1}^m \text{row}(x)_j d_j = \sum_{\text{row}(x)_j=1} d_j$$

Fact: the binary vector $\text{row}(x)$ has a long sparse part!

$$\text{row}(x) := \underbrace{\text{sparse}(x)}_{\text{constant weight } w} \parallel \underbrace{\text{dense}(x)}_{\text{random}} \in \{0, 1\}^{s+d}$$

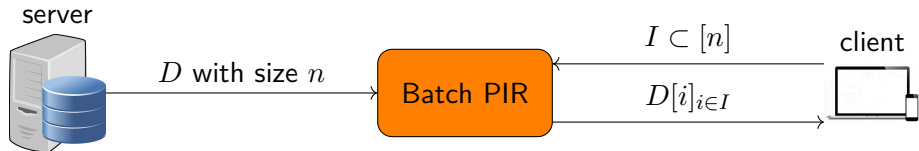
where $s = O(m)$, $d = o(m)$.

Using the linearity of inner-product, Decode can be re-written as:

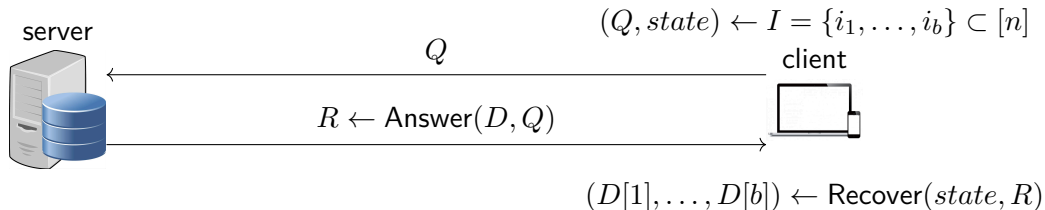
$$\text{Decode}(D, x) = \langle \text{row}(x), D = D_0 \parallel D_1 \rangle = \langle \text{sparse}(x), D_0 \rangle + \langle \text{dense}(x), D_1 \rangle$$

where $|D_0| = s = O(m)$, $|D_1| = d = o(m)$.

Batch Private Information Retrieval (Batch PIR)



An explicit construction of Batch PIR consists of (Query, Answer, Recover):



Batch PIR

Batch PIR scheme satisfies the following properties:

- **Correctness:** For any dataset D , all distinct inputs $I = \{i_1, \dots, i_b\}$,
 $(Q, state) \leftarrow \text{Query}(I)$:

$$\text{Recover}(state, \text{Answer}(D, Q)) = (D[i_1], \dots, D[i_b])$$

- **Query privacy:** For any distinct batch query sets I_1, I_2 with $|I_1| = |I_2|$:

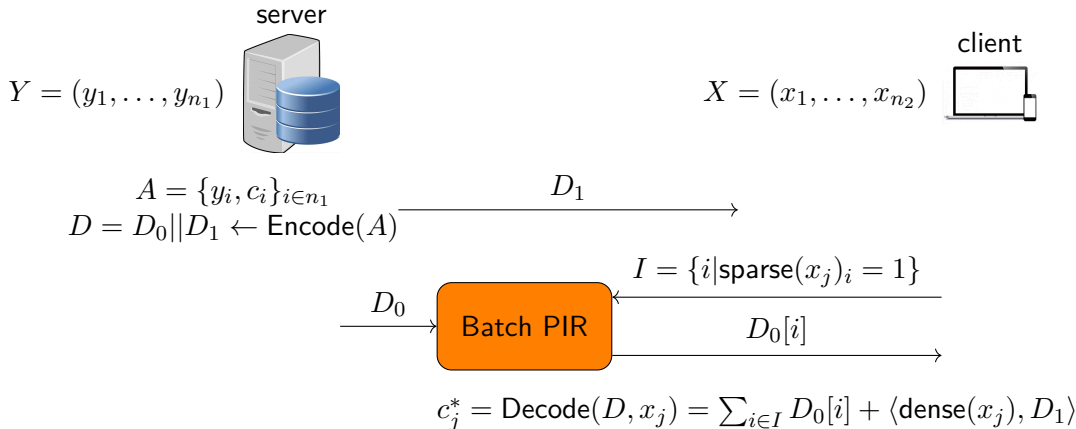
$$Q_0 \approx_c Q_1$$

where $(Q_\beta, state_\beta) \leftarrow \text{Query}(I_\beta)$.

- **Compactness:** $|Q| + |R| = o(n)$

Solution: Sparse OKVS + Batch PIR

Idea: directly transmitting the dense but short part, employing batch PIR to transmitting the sparse yet long part.



The overall communication complexity is $o(n_1)$

Outline

1 PSO in Unbalanced Setting

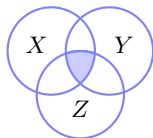
2 PSO in Multi-Party Setting

Multi-Party Private Set Operations

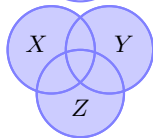
PSO has been extensively studied in the last four decades.

- The research community focuses on the two-party setting.
- Multi-party receives much less attention.

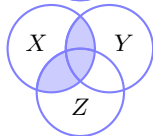
Multi-party PSO is more useful in real-world applications.



$X \cap Y \cap Z$	intersection
$ X \cap Y \cap Z $	cardinality
$f(X \cap Y \cap Z)$	general computation



$X \cup Y \cup Z$	union
$ X \cup Y \cup Z $	cardinality
$f(X \cup Y \cup Z)$	general computation



$X \cap (Y \cup Z)$	finite set operations
$ X \cap (Y \cup Z) $	cardinality
$f(X \cap (Y \cup Z))$	general computation

Why MPSO is Difficult?

The extension of two-party to multi-party is not easy (even in the semi-honest setting).



Security is more stringent ($m = \#$ parties)

- two-party scenario: $m = 2 \leadsto$ no collusion attack
- multi-party scenario: arbitrary $m \geq 3 \leadsto$ have to defend against collusion attack



Functionality is more expressive

- two-party scenario: only intersection, union, and computation on intersection
- multi-party scenario: the number of operations explosively blows up in m

SOTA of MPSO

Multi-party PSI (MPSI) has been well-studied in the last decades

Multi-party PSU (MPSU) and its variants

- No MPSU based on OT and symmetric-key techniques is secure against arbitrary collusion.
 - No MPSU achieves both linear computation and linear communication complexity.
 - No protocol is able to compute the cardinality or general function of the union.
-

Generic MPSO protocols

- No MPSO is able to compute arbitrary number of set operations over the private sets.
- No MPSO is able to compute the cardinality or general function of the set through a finite number of set operations.

Our Work on MPSU



Minglang Dong, Cong Zhang, Yujie Bai, **Yu Chen**

Efficient Multi-Party Private Set Union Without Non-Collusion Assumptions

USENIX Security 2025

MPSU and its variants

- The first MPSU protocol based on OT and symmetric-key techniques that is secure against arbitrary collusion.
- The first MPSU protocol achieving both linear computation and linear communication complexity.
- The first protocol realizing the functionality to compute the cardinality or general function of the union.

Our Work on MPSO



Minglang Dong, **Yu Chen**, Cong Zhang, Yujie Bai, Yang Cao

Multi-Party Private Set Operations from Predicative Zero-Sharing

ACM CCS 2025


Generic MPSO

- The first MPSO protocol that can compute arbitrary finite number of set operations over the private sets, and compute the cardinality or general function of the resulting set.


Thanks for Your Attention!

Any Questions?

Reference I

-  Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal.
Labeled PSI from fully homomorphic encryption with malicious security.
In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, pages 1223–1237. ACM, 2018.
-  Hao Chen, Kim Laine, and Peter Rindal.
Fast private set intersection from homomorphic encryption.
In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 1243–1255. ACM, 2017.
-  Kelong Cong, Radames Cruz Moreno, Mariana Botelho da Gama, Wei Dai, Ilia Iliashenko, Kim Laine, and Michael Rosenberg.
Labeled PSI from homomorphic encryption with reduced computation and communication.
In CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 1135–1150. ACM, 2021.

Reference II

-  Cong Zhang, Yu Chen, Weiran Liu, Min Zhang, and Dongdai Lin.
Optimal private set union from multi-query reverse private membership test.
In *USENIX Security 2023*, 2023.
<https://eprint.iacr.org/2022/358>.