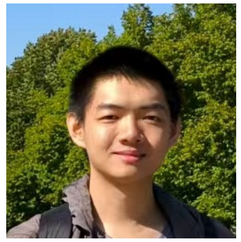
FABLE: Batched Evaluation on Confidential Lookup Tables in 2PC

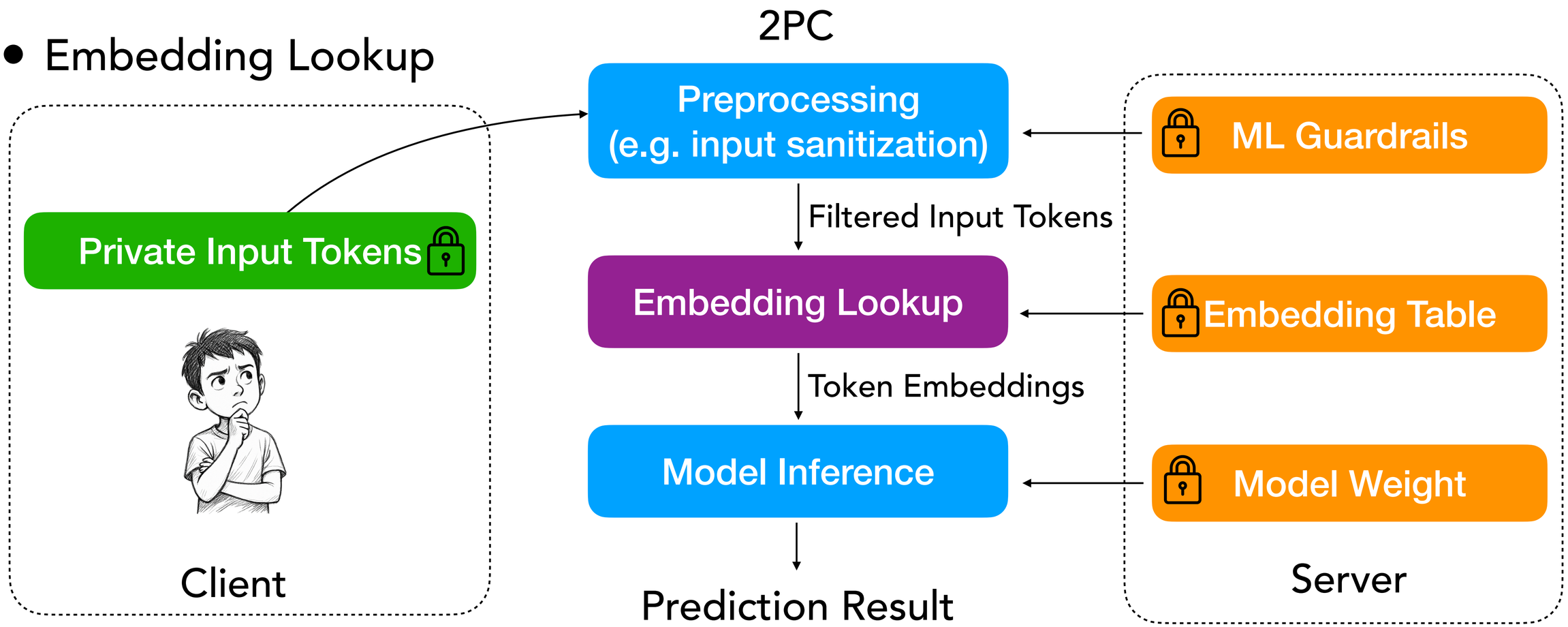
Zhengyuan Su*, Qi Pang†, Simon Beyzerov†, Wenting Zheng†

**Tsinghua University, †Carnegie Mellon University*



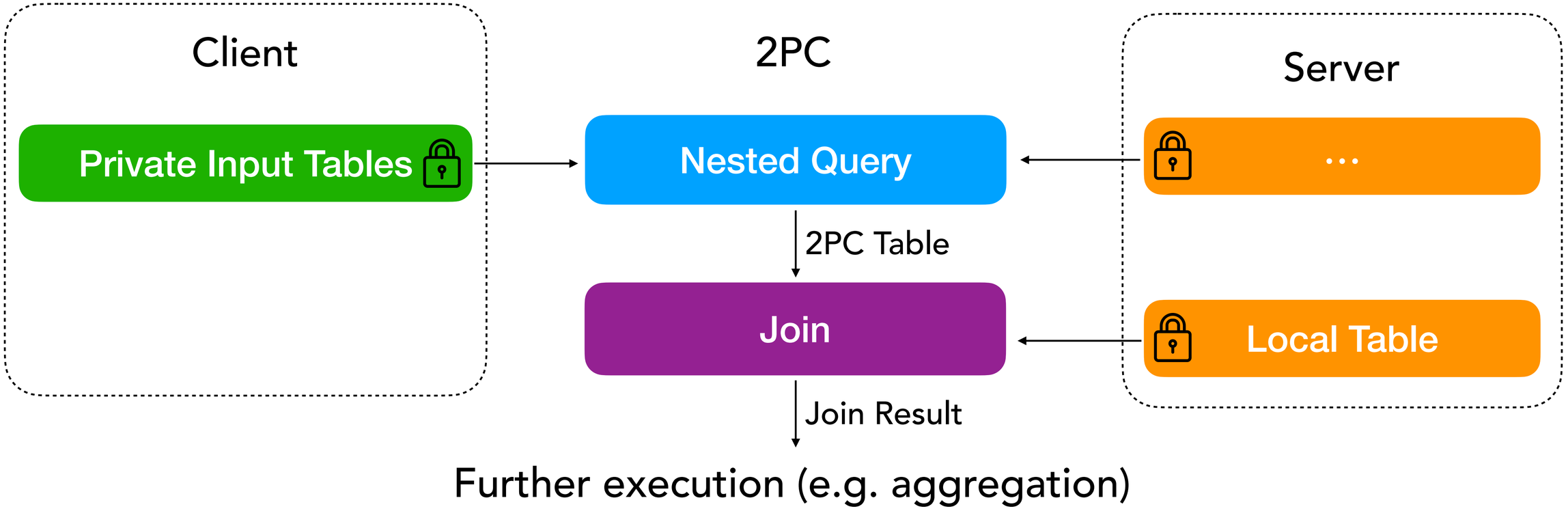

Secure LUT Evaluation is Commonly Used in 2PC

- Embedding Lookup



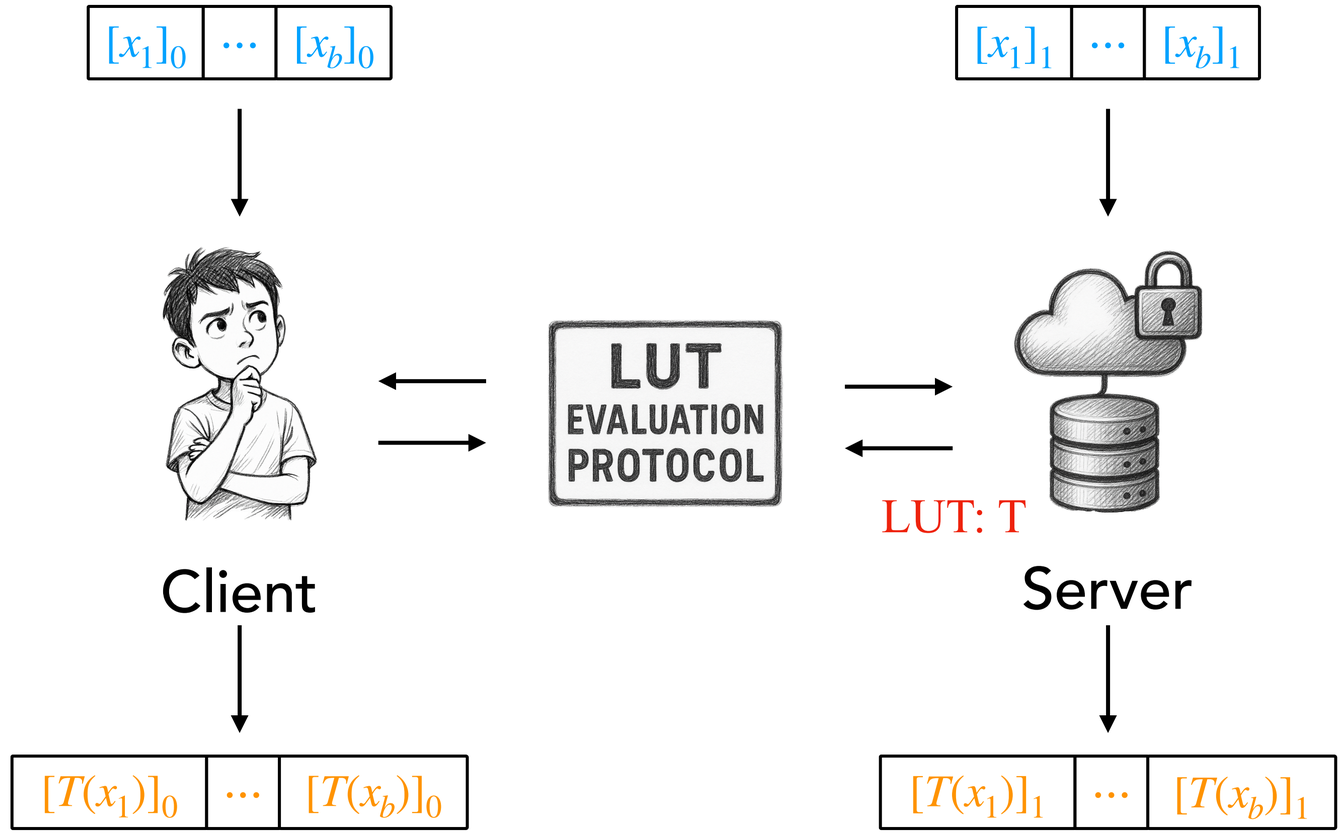
Secure LUT Evaluation is Commonly Used in 2PC

- Query Execution



Setting & Threat Model

- Semi-honest parties
- Server has the confidential LUT
- Server and client hold a batch of secret shared inputs in 2PC
- Server and client obtain batched secret shared outputs in 2PC



Prior Works

- LUT as a 2PC primitive[IKM+13, DKS+17, BHS+23]
  - Target small and public LUTs
  - Communication linear/superlinear in table sizes
- Distributed ORAM's read[DS17, VHG23]
  - Target two-server settings

Not scalable & LUT is not confidential

Heavyweight client

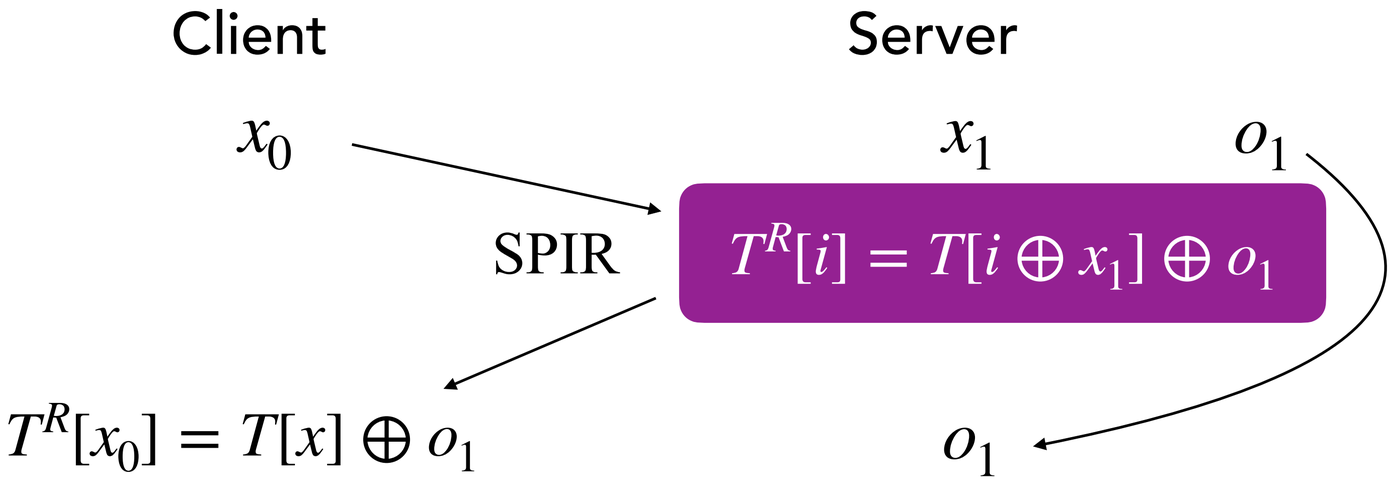
Can we design a secure LUT evaluation protocol that can scale to **large tables**, guarantee **LUT confidentiality**, support **lightweight client** and have **concrete server efficiency**?

Our Solution — FABLE

- Our idea: PIR + symmetric security + batching techniques
  - PIR: scalability and client lightweightness
  - Symmetric security: LUT confidentiality
  - Batching: concrete efficiency

PIR-based LUT Evaluation — A Naive Construction

- A variant of 2P-DUORAM[VHG23]
- Pros
  - Sublinear communication
  - Lightweight client
  - LUT confidentiality
- Cons: poor concrete performance
  - Table needs to be re-prepared for every input

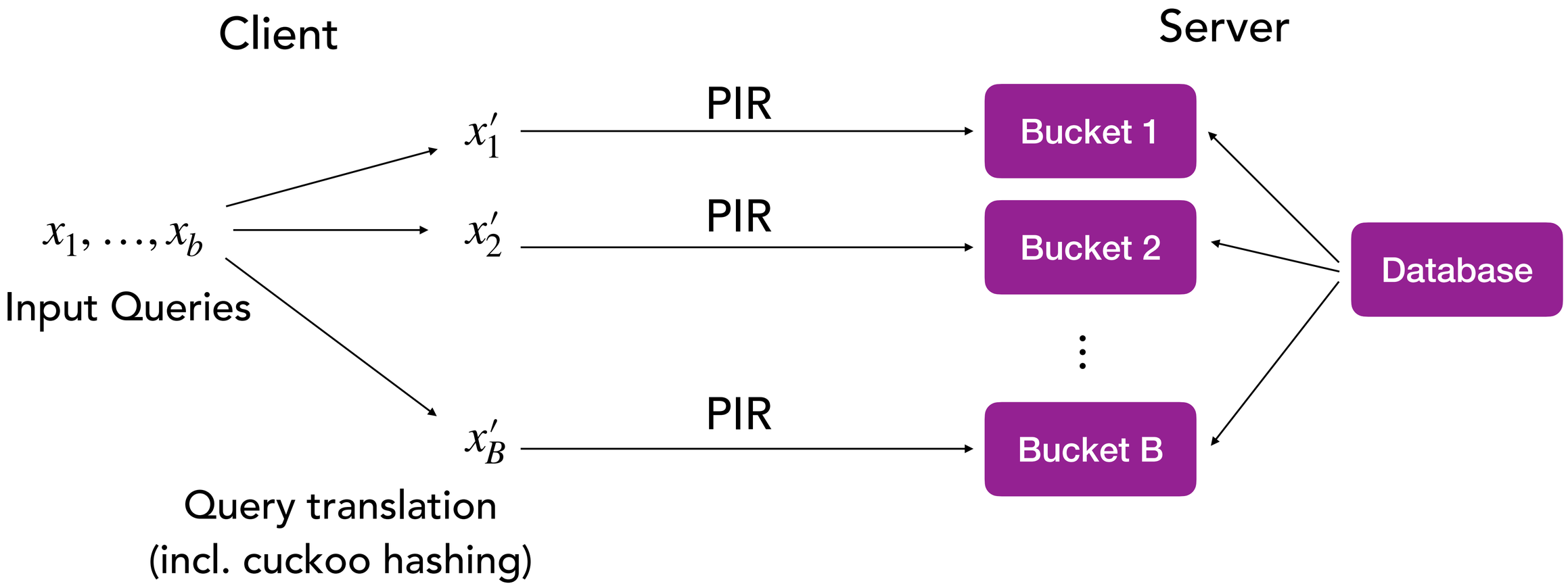


Insight: Batching is a Common Workload!

- Embedding Lookup
  - A piece of text contains hundreds of words
  - Embedding lookup receives a large batch of tokens
- Query Execution
  - Tables normally have thousands of rows
  - Table join = LUT lookup for each row

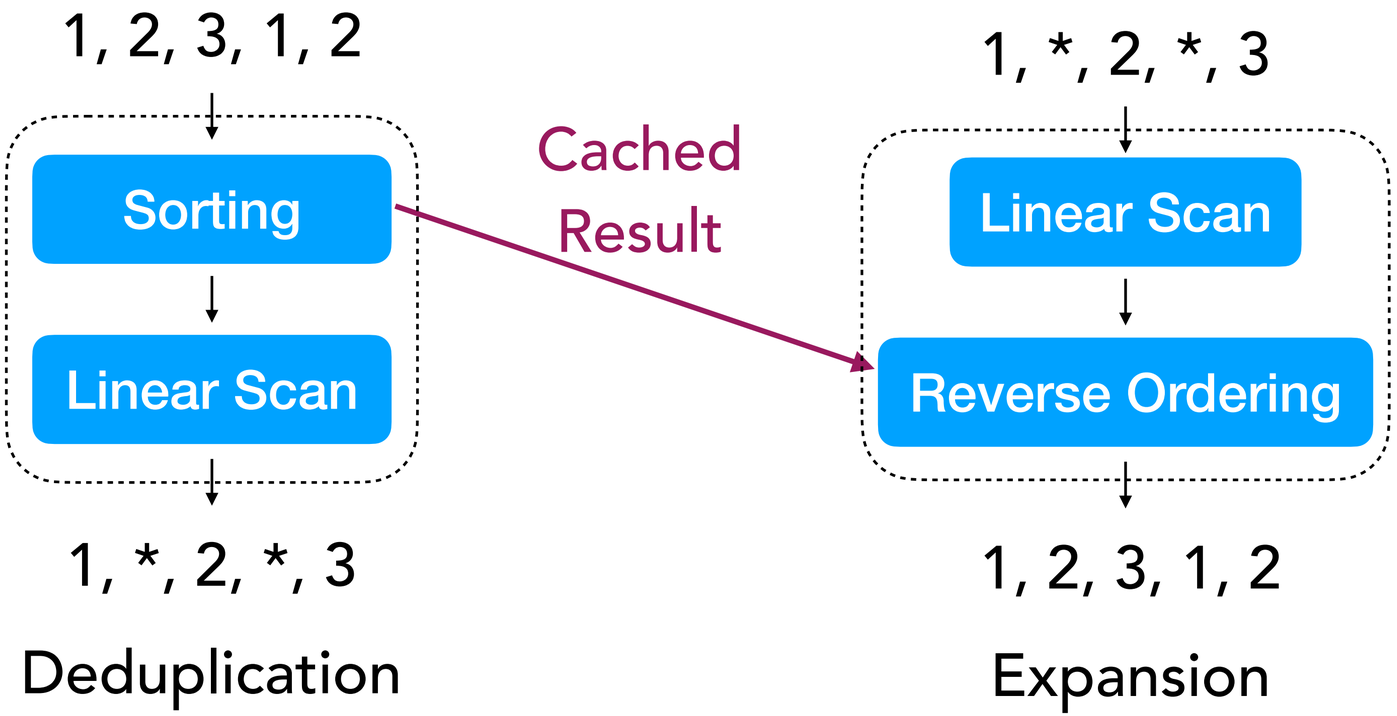
LUT evaluation protocol based on batch PIR for better amortized performance!

Overview of Batch PIR



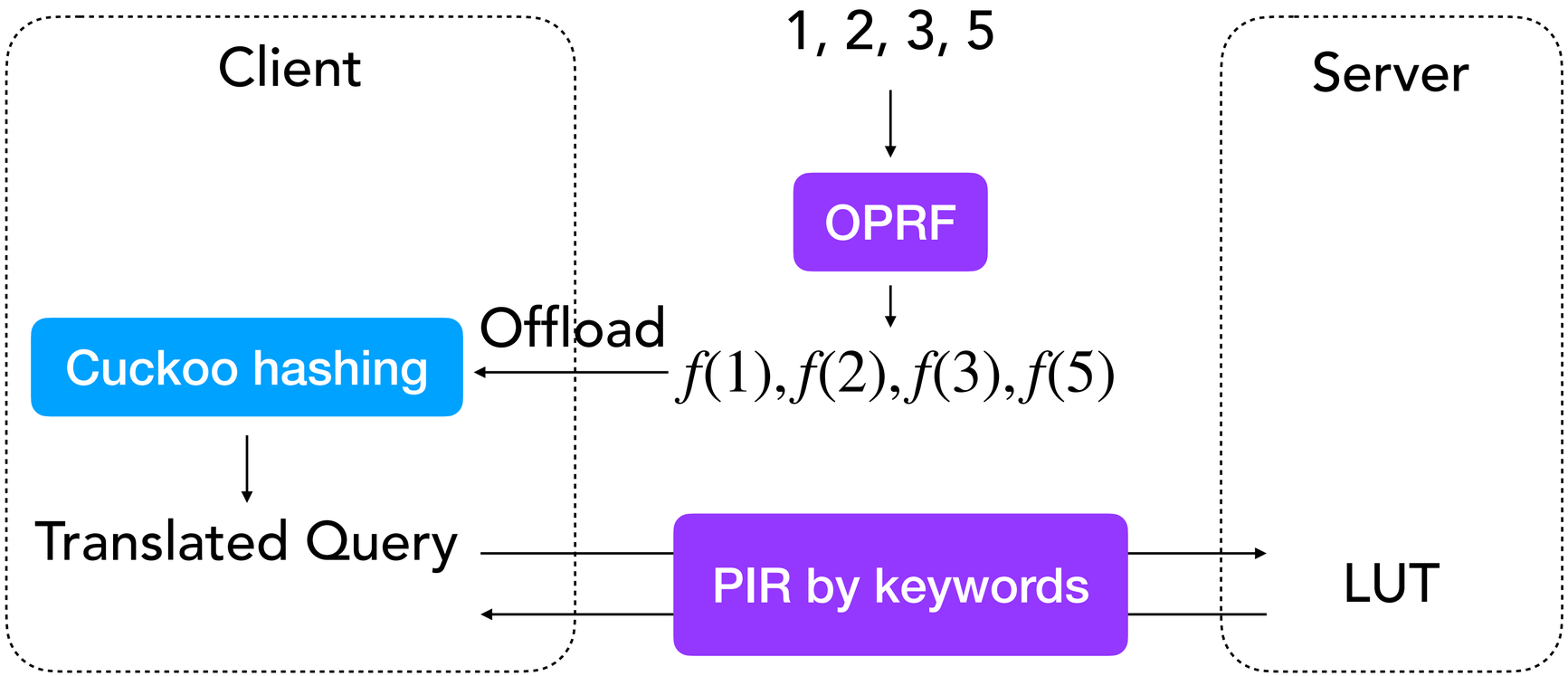
Challenge 1. Efficient Query Translation in 2PC

- Input needs to be deduplicated
  - Novel protocols with caching to avoid recomputation
  - Saved 84.1% comm.



Challenge 1. Efficient Query Translation in 2PC

- Input needs to be deduplicated — Novel protocols with caching
- Cuckoo hashing in 2PC is impractical — Offloading + PIR by keywords



Challenge 2. Protect LUT’s Confidentiality

- SOTA Batch PIRs use HE ciphertexts for interaction
- To prevent information leakage from ciphertexts
  - Encrypted message — Database masking
  - Noise — Noise flooding

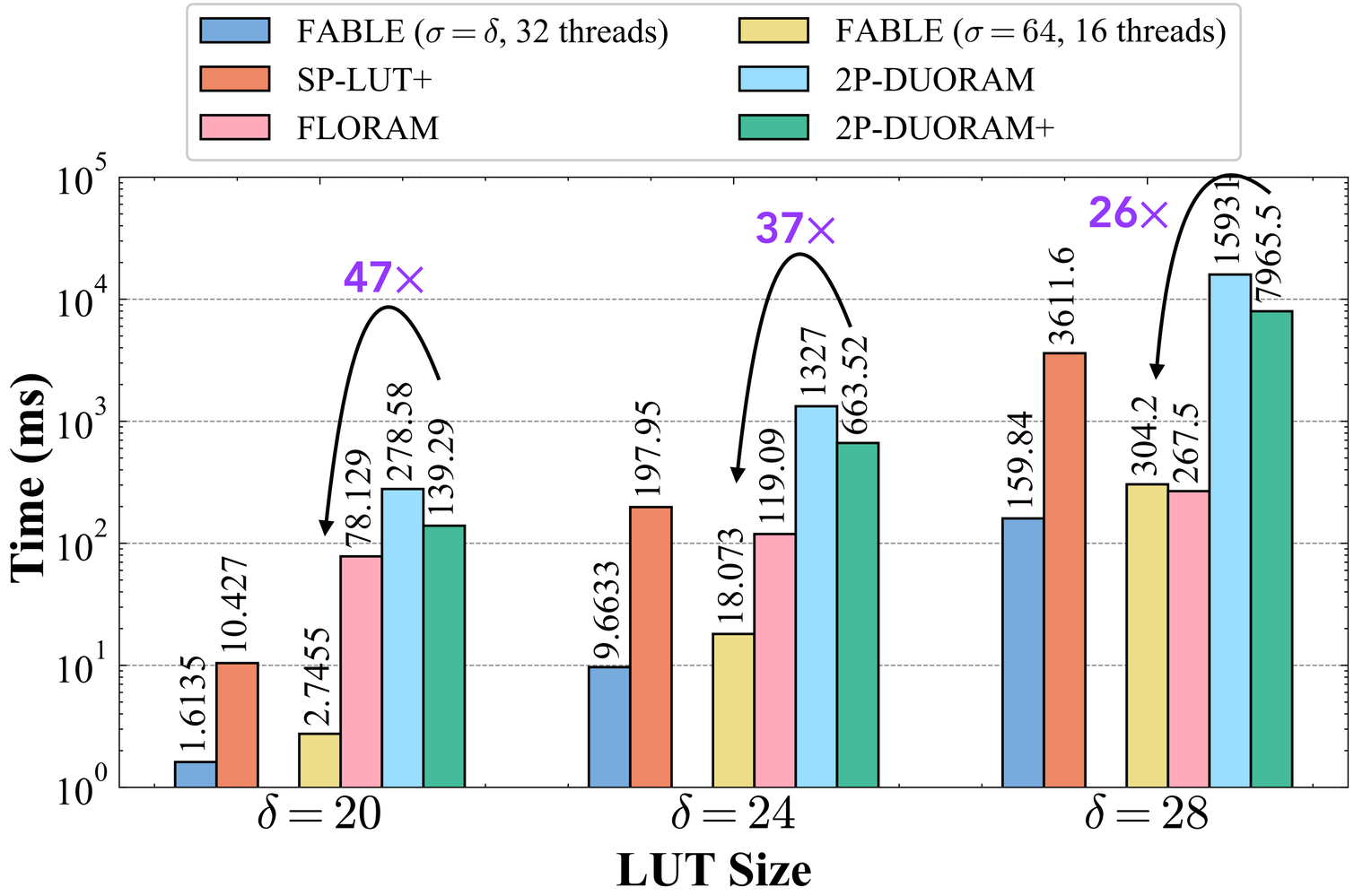
Evaluation — Asymptotic Complexity

- Sublinear client computation and communication

	Client Computation	Server Computation	Communication
FABLE	**$O(\sqrt{N})$**	**O(N)**	**$O(\sqrt{N})$**
SP-LUT+	O(log N)	O(N)	O(N)
FLUTE	$O(N^2)$	$O(N^2)$	O(N)
FLORAM-CPRG	O(N)	O(N)	O(log N)
2P-DUORAM	O(N)	O(N)	O(log N)

Evaluation — Amortized Execution Time

- Concretely efficient
- Orders of magnitude faster



Amortized execution time w.r.t. table size under LAN

Evaluation — Applications

- Secure Embedding Lookup
  - Baseline: Crypten + MOTION (for MT generation)
  - 12,868s → 28.22s (456×)
- Secure Query Execution
  - Baseline: Senate’s sort-compare-shuffle circuit
  - 149s → 10s (15×)

Conclusion

We propose FABLE, a scalable and efficient protocol for confidential LUT evaluation.

FABLE enjoys lightweight client computation, concretely efficient server computation, scalable communication, and LUT confidentiality.

FABLE shows applicability to ML and analytics workloads.

Email: su-zy21@tsinghua.org.cn

Paper:

Code: