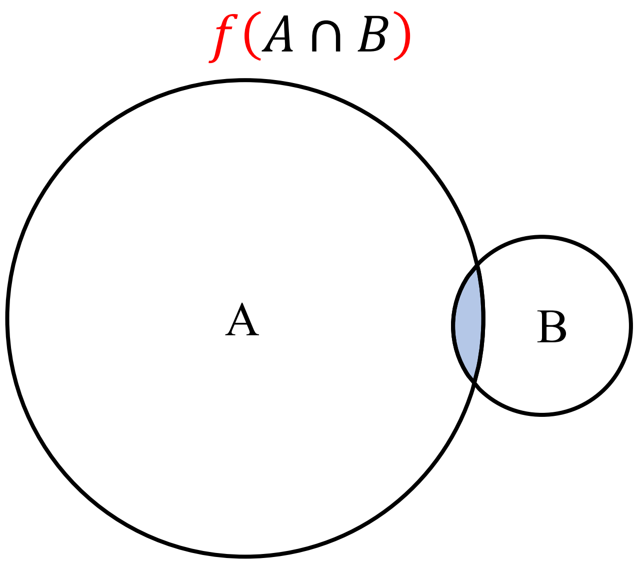
Unbalanced Circuit Private Set Intersection from Oblivious Key-Value Retrieval



Meng Hao[1], Weiran Liu[2], Liqiang Peng[2], Hongwei Li[3], Cong Zhang[4], Hanxiao Chen[1], Tianwei Zhang[1]

[1]Nanyang Technological University

[2]Alibaba Group

[3]Peng Cheng Laboratory

[4]Institute for Advanced Study, BNRist, Tsinghua University

Outline

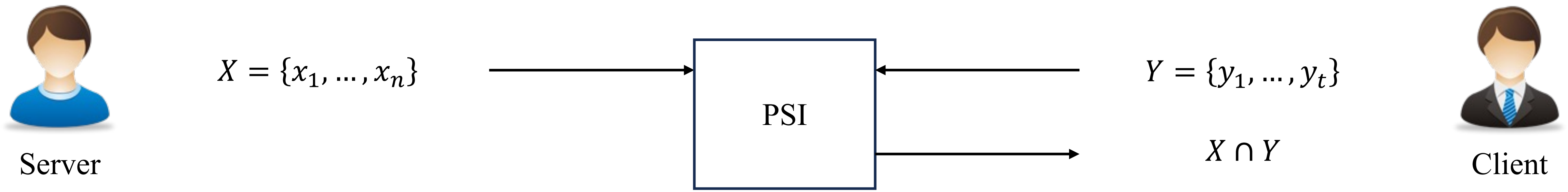
Outline

01 | Introduction to Circuit-PSI

1.1 Private Set Intersection (PSI)

PSI Functionality

PSI enables to compute the intersection of private sets. The client only learns the intersection, while the server learns nothing.



PSI Researches

PSI has been extensively studied in the last two decades.

- Balanced setting: [KKRT16, CM20, RR22] achieve linear complexity, and almost as efficient as insecure hash protocol.
- Unbalanced setting: [CLR17, CHLR18, CMdG+21] achieve sub-linear complexity of large set size.

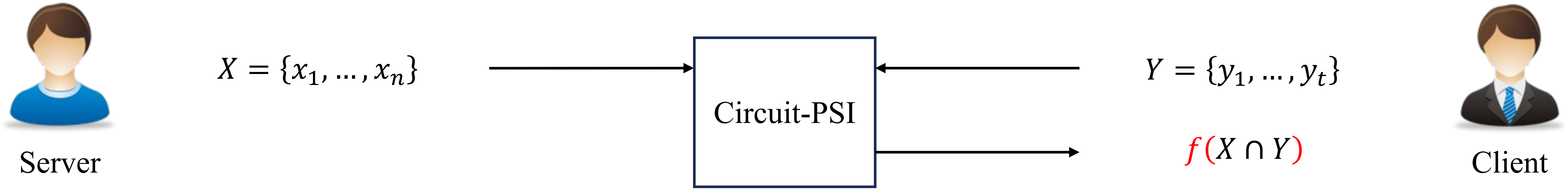
Other PSI variants have also been proposed.

- Both parties learn the output (two-sided PSI) [MPR+20]
- Client learns size of intersection (PSI-CA) [HFH99].
- Client learns the sum of the intersection (intersection-sum) [IKN+17].
- Parties compute some other function based on the intersection (e.g., threshold PSI [GS19]) or related payloads [BKM+20].

1.2 Circuit-PSI (CPSI)

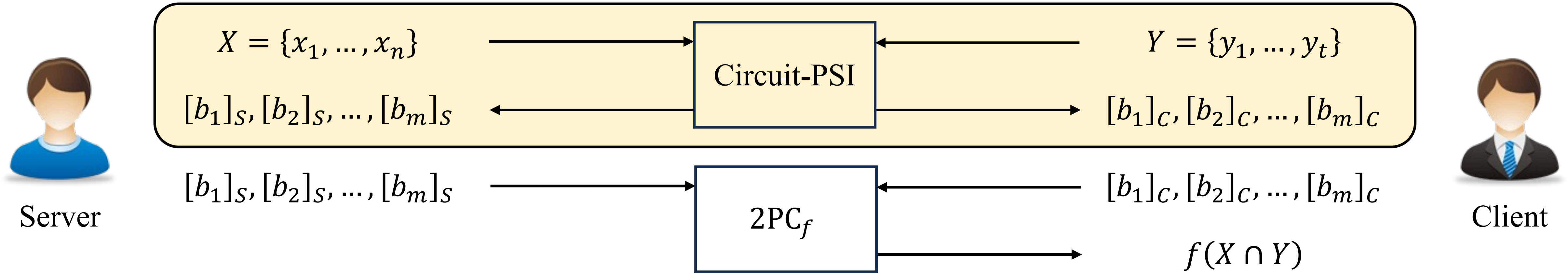
CPSI Functionality

CPSI enables the client to learn the result of a function on the intersection without leaking the intersection itself.



Here f is a post-processing function, e.g., the intersection cardinality $f = |X \cap Y|$, a threshold function $f = |X \cap Y| > \tau$?

CPSI Construction



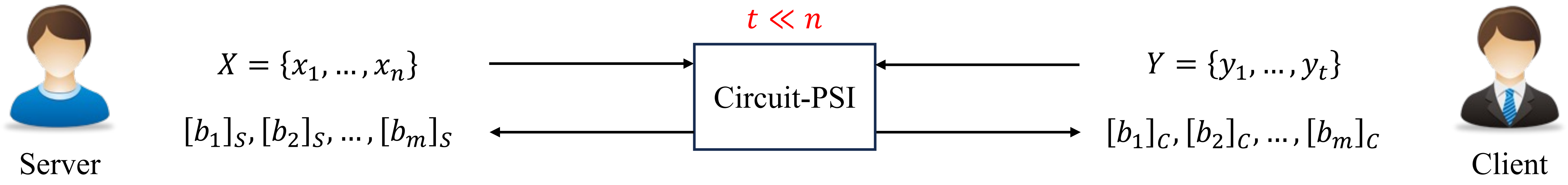
Here $[\cdot]$ denotes Boolean secret sharing, e.g., for $b \in \{0,1\}$, it holds $[b]_C \oplus [b]_S = b$. It is possible to attach payloads in $2PC_f$.

Remark: Typically $m = O(n)$ or $m = O(t)$. There is no differences in the balanced setting $(t = n)$, while in the unbalanced setting $(t \ll n)$ things change.

1.3 Unbalanced CPSI (UCPSI)

Unbalanced Setting

This work focuses on unbalanced setting: the client possesses a smaller set, and the server holds a considerably larger set.



UCPSI Application

U(C)PSI has several applications, e.g., password leakage detection, image abuse detection, UPSI with associated data.

The Apple PSI System

Abhishek Bhowmick, Apple Inc.
Dan Boneh, Stanford University
Steve Myers, Apple Inc.
Kunal Talwar, Apple Inc.
Karl Tarbe, Apple Inc.

July 29, 2021

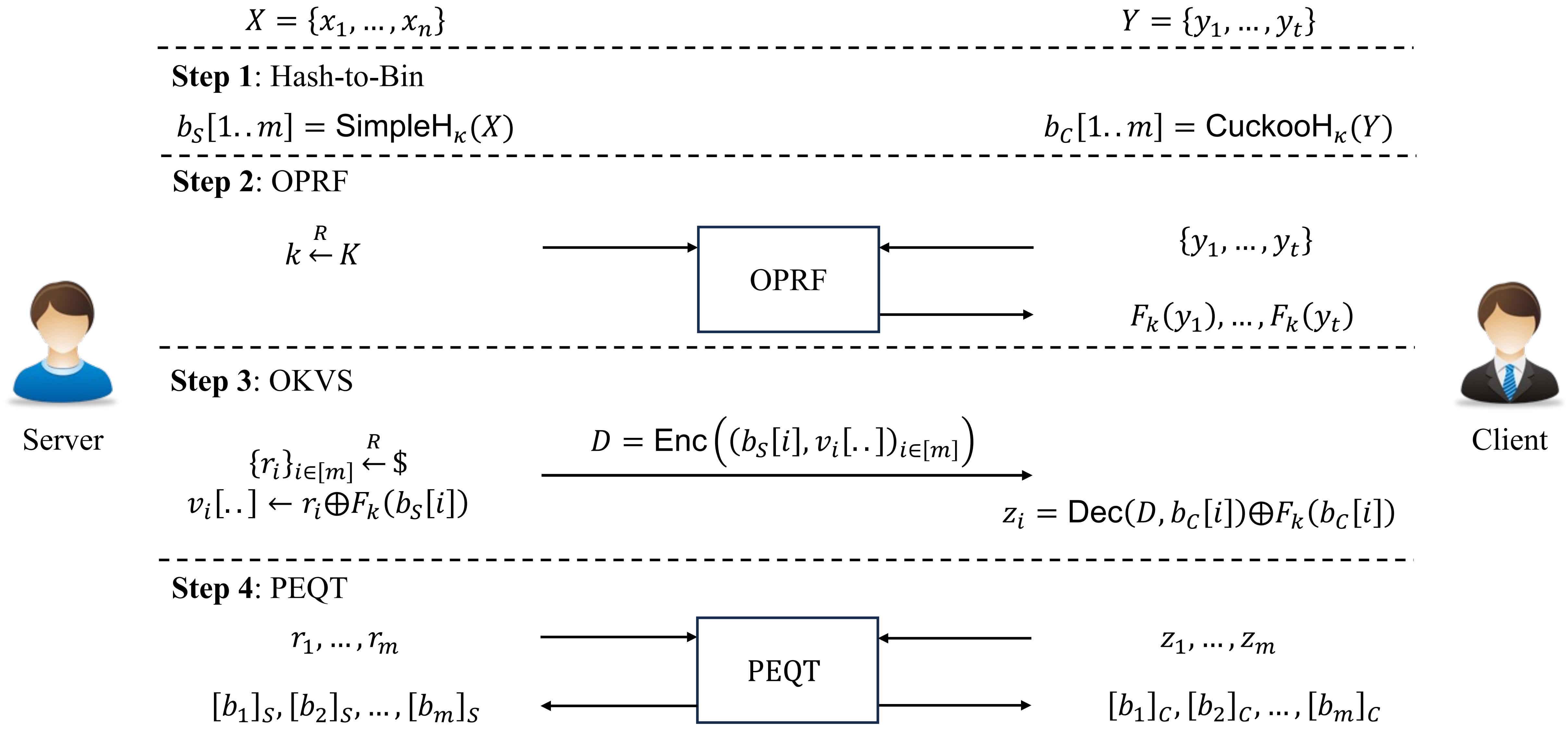
(Abstract) This document describes the constraints that drove the design of the Apple private set intersection (PSI) protocol. Apple PSI makes use of a variant of PSI we call private set intersection with associated data (PSI-AD), and an extension called threshold private set intersection with associated data (tPSI-AD). We describe a protocol that satisfies the constraints, and analyze its security. The context and motivation for the Apple PSI system are described on the main project site.

Remark: In the unbalanced setting ($t \ll n$), we must ensure that $m = O(t)$, otherwise the communication cost must be at least $O(n)$.

Outline

02 | Existing CPSI Framework

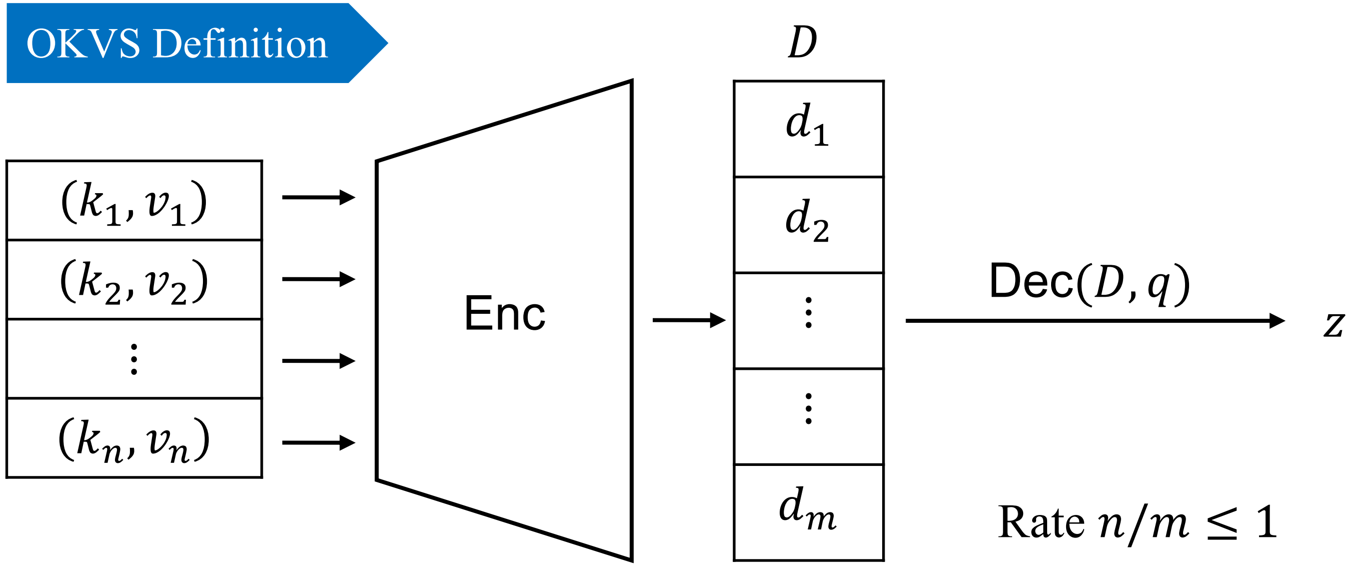
2.1 Existing CPSI Framework Overview



[PSTY19] B. Pinkas, T. Schneider, O. Tkachenko, A. Yanai. Efficient circuit-based PSI with linear communication. EUROCRYPT 2019, pp. 122-153.

2.2 Oblivious Key-Value Store (OKVS)

OKVS Definition



$$\begin{pmatrix} - & h(k_1) & - \\ - & h(k_2) & - \\ \vdots & \ddots & \vdots \\ - & h(k_n) & - \end{pmatrix} \times \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_m \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

$h(\cdot)$ is a mapping function.

OKVS Properties

- **Correctness**: For any $L = \{(k_1, v_1), \ldots, (k_n, v_n)\}$, if $D = \text{Enc}(L)$, then for any $q = k_i$, $\text{Dec}(D, k_i) = v_i$.
- **Obliviousness**: For any $(k_1^0, \ldots, k_n^0) \neq (k_1^1, \ldots, k_n^1)$, if $v_1, \ldots, v_n \xleftarrow{R} \$$, then

$$\text{Enc}\left((k_1^0, v_1), \ldots, (k_n^0, v_n)\right) \approx_s \text{Enc}\left((k_1^1, v_1), \ldots, (k_n^1, v_n)\right)$$

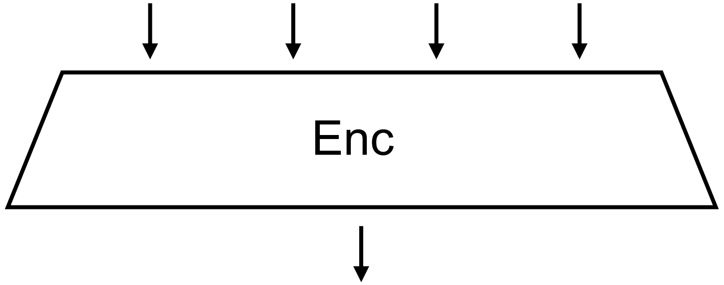
- **Binary** (optional): $h(k) \in \{0,1\}^m$. This work focuses on binary OKVS.

[GPR+23] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, A. Yanai. Oblivious key-value stores and amplification for private set intersection. CRYPTO 2021, pp. 395-425.

2.2 Oblivious Key-Value Store (OKVS)

OKVS Example: Polynomial

$\{(k_1, v_1), (k_2, v_2), \dots, (k_n, v_n)\}$



$D = s_1 + s_2 x + s_3 x^2 + \cdots + s_n x^{n-1}$

$$\begin{pmatrix} 1 & k_1 & k_1^2 & \cdots & k_1^{n-1} \\ 1 & k_2 & k_2^2 & \dots & k_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & k_n & k_n^2 & \cdots & k_n^{n-1} \end{pmatrix} \times \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

$$h(k) = (1, k, k^2, \dots, k^{n-1})$$

Rate: $n/m = 1$ (Optimal)

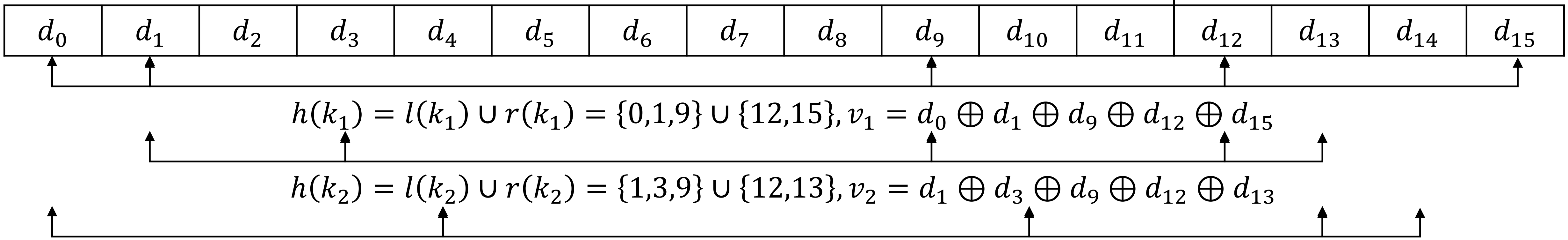
Require: $k_i, v_i \in \mathbb{F}$

Enc: $O(n \log n^2)$

Dec: $O(n \log n^2)$

OKVS Example: GCT

$m_l = \epsilon n$ | $m_r = O(\log n)$



$h(k_1) = l(k_1) \cup r(k_1) = \{0,1,9\} \cup \{12,15\}, v_1 = d_0 \oplus d_1 \oplus d_9 \oplus d_{12} \oplus d_{15}$

$h(k_2) = l(k_2) \cup r(k_2) = \{1,3,9\} \cup \{12,13\}, v_2 = d_1 \oplus d_3 \oplus d_9 \oplus d_{12} \oplus d_{13}$

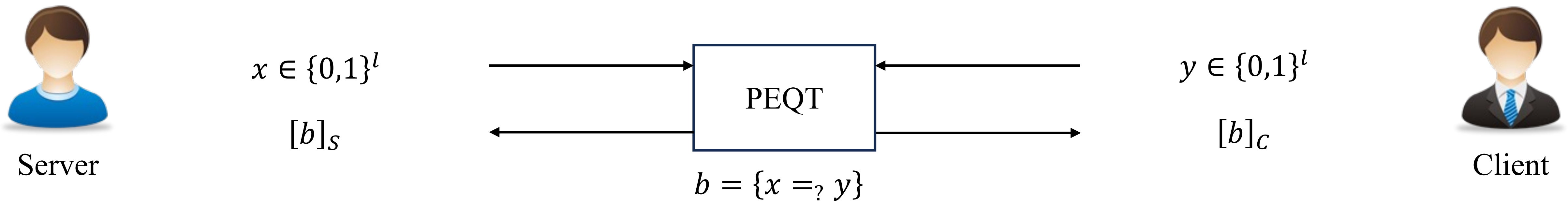
$h(k_3) = l(k_3) \cup r(k_3) = \{0,4,10\} \cup \{13,14\}, v_3 = d_0 \oplus d_4 \oplus d_{10} \oplus d_{13} \oplus d_{14}$

[PRTY20] B. Pinkas, M. Rosulek, N. Trieu, A. Yanai. PSI from PaXoS: fast, malicious private set intersection. EUROCRYPT 2020, pp. 739-767.

2.3 Private Equality Test (PEQT)

PEQT Functionality

PEQT enables two parties to privately compare two values and obtain the shares of whether the values are equal or not.



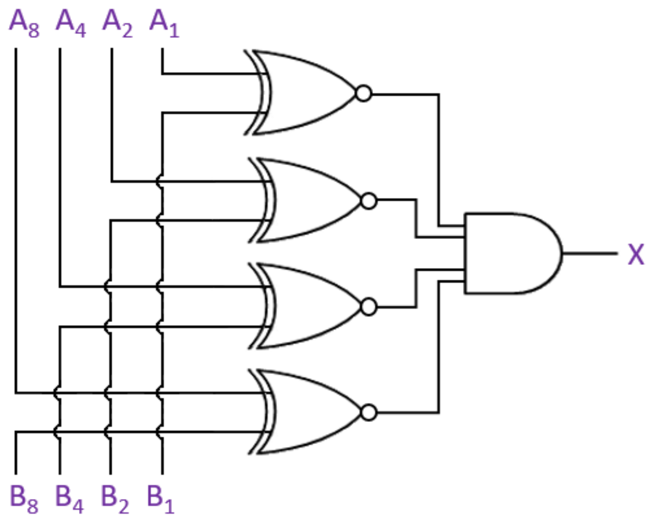
PEQT Researches

Trivial solution: use general 2PC to implement the equality test functionality.

Improved solution: XOR and NOT are free; batch AND to make it $\log l$ rounds.

Advanced solution: use 1-out-of-N OT and general 2PC to reduce communication.

- The idea comes from the protocol for Millionaires' in CrypTFlow2 [RRK+20].
- It is introduced in PEQT and in Private Membership Test (PMT) [CGS22].



[RRK+20] D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, R. Sharma. CrypTFlow2: Practical 2-party secure inference. CCS 2020, pp. 325-342.
[CGS22] N. Chandran, D. Gupta, A. Shah. Circuit-PSI with linear complexity via relaxed batch OPPRF. PETS 2022.

Outline

03 | OKVR and UCPSI Construction

3.1 Communication Bottleneck of Existing UCPSI

Server

Client

$X = \{x_1, \dots, x_n\}$ $\qquad$ $Y = \{y_1, \dots, y_t\}$

Step 1: Hash-to-Bin

$b_S[1..m] = \mathsf{SimpleH}_\kappa(Y)$ $\qquad$ $b_C[1..m] = \mathsf{CuckooH}_\kappa(Y)$

Step 2: OPRF

$k \overset{R}{\leftarrow} K$ → DH-based OPRF ← $\{y_1, \dots, y_t\}$

DH-based OPRF → $F_k(y_1), \dots, F_k(y_t)$

Step 3: OKVS

$\{r_i\}_{i\in[m]} \overset{R}{\leftarrow} \$$

$v_i[..] \leftarrow r_i \oplus F_k(b_S[i])$

$D = \mathsf{Enc}\left((b_S[i], v_i[..])_{i\in[m]}\right)$ →

$O(n)$ communication!

$z_i = \mathsf{Dec}(D, b_C[i]) \oplus F_k(b_C[i])$

Step 3: PEQT

$r_1, \dots, r_m$ → PEQT ← $z_1, \dots, z_m$

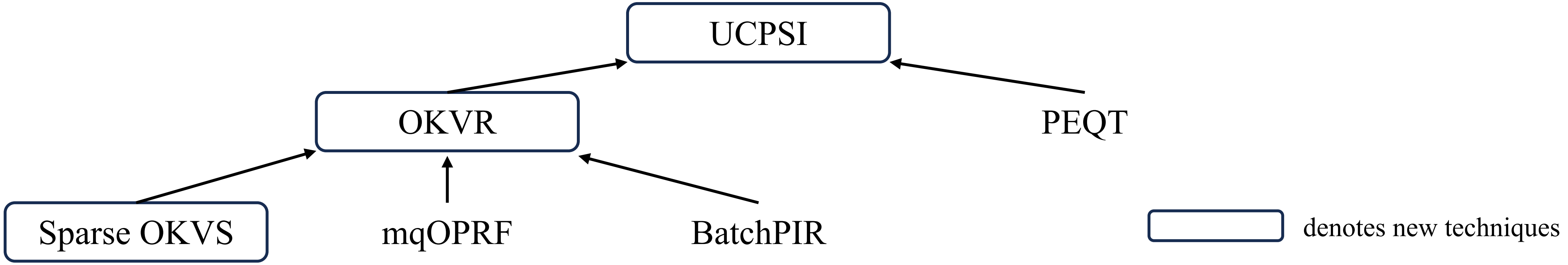
$[b_1]_S, [b_2]_S, \dots, [b_m]_S$ ← PEQT → $[b_1]_C, [b_2]_C, \dots, [b_m]_C$

[PSTY19] B. Pinkas, T. Schneider, O. Tkachenko, A. Yanai. Efficient circuit-based PSI with linear communication. EUROCRYPT 2019, pp. 122-153.

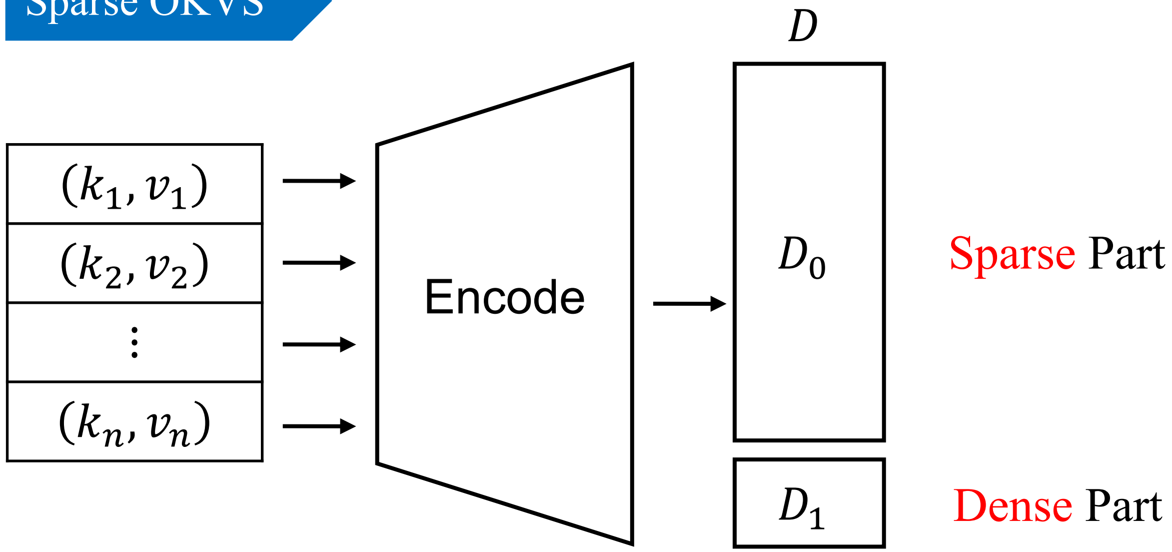
3.2 New Framework for UCPSI

New Framework

Our UCPSI is constructed from Oblivious Key-Value Retrieval (OKVR). The core of OKVR is Sparse OKVS.



Sparse OKVS



Sparse properties

- $|D_0| \gg |D_1|$, e.g., $|D_0| = O(n), |D_1| = O(\log n)$
- $l(k) \in \{0,1\}^{|D_0|}$ is sparse with constant weight α.
- $r(k) \in \{0,1\}^{|D_1|}$ is dense with random weight.

Instantiations

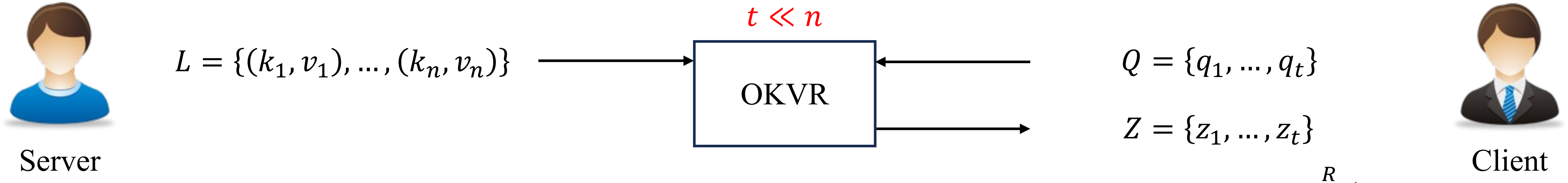
- Garbled Cuckoo Table [GPR+23] is a Sparse OKVS with $|D_0| = O(n), |D_1| = O(\log n)$, and $\alpha \in \{2,3\}$.

[GPR+23] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, A. Yanai. Oblivious key-value stores and amplification for private set intersection. CRYPTO 2021, pp. 395-425.

3.3 Oblivious Key-Value Retrieval (OKVR)

OKVR Functionality

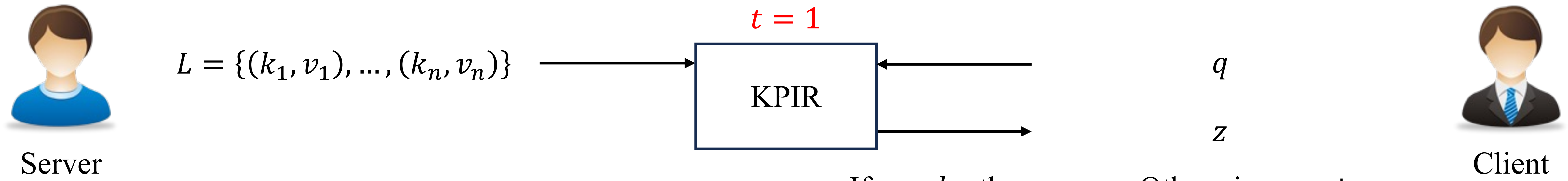
OKVR enables the client to obliviously retrieve values from the server's key-value pairs for all keys queried by the client.



If $q_i = k_j$, then $z_i = v_j$. Otherwise, $z_i \overset{R}{\leftarrow} \$$

Keyword PIR

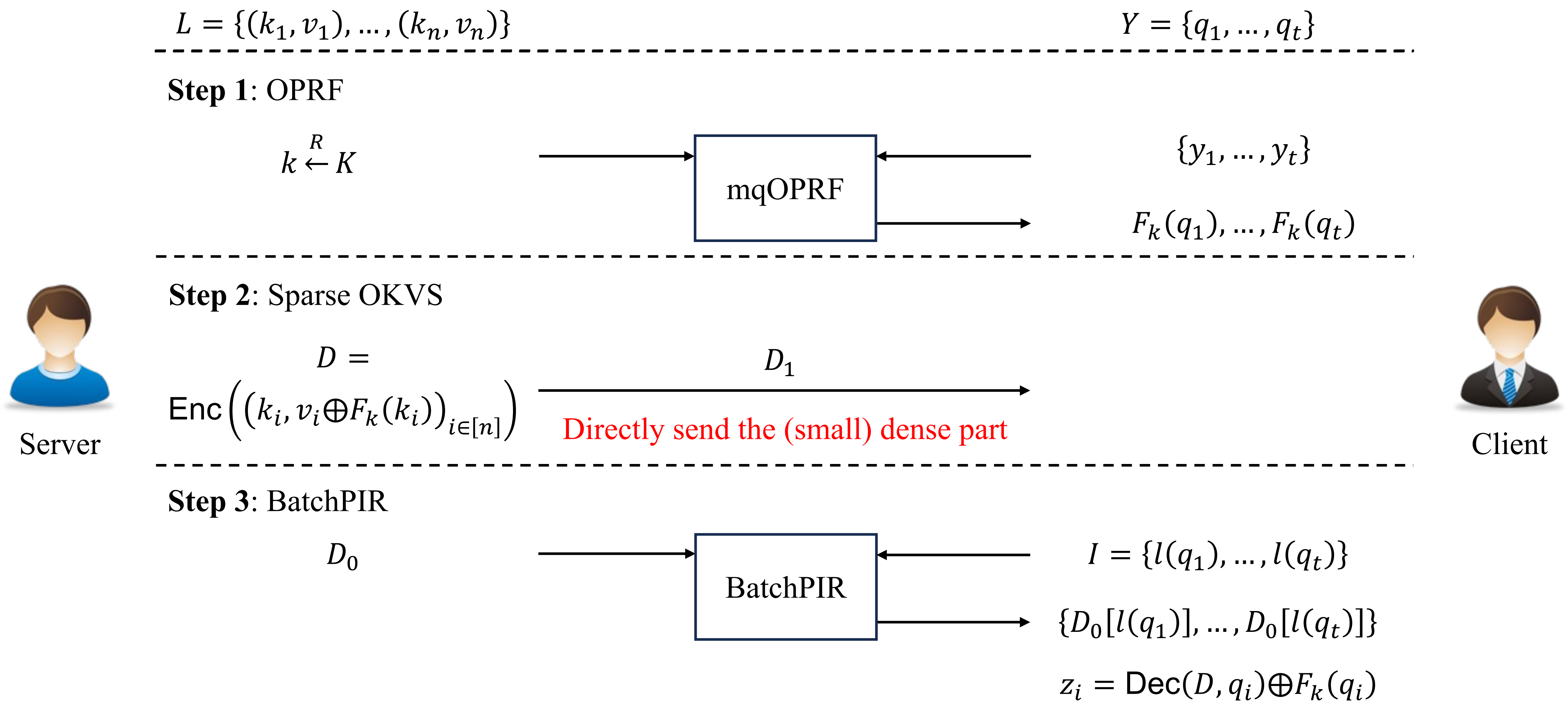
KPIR enables the client to obliviously retrieve a value from the server's key-value pairs and know if the server has the key.



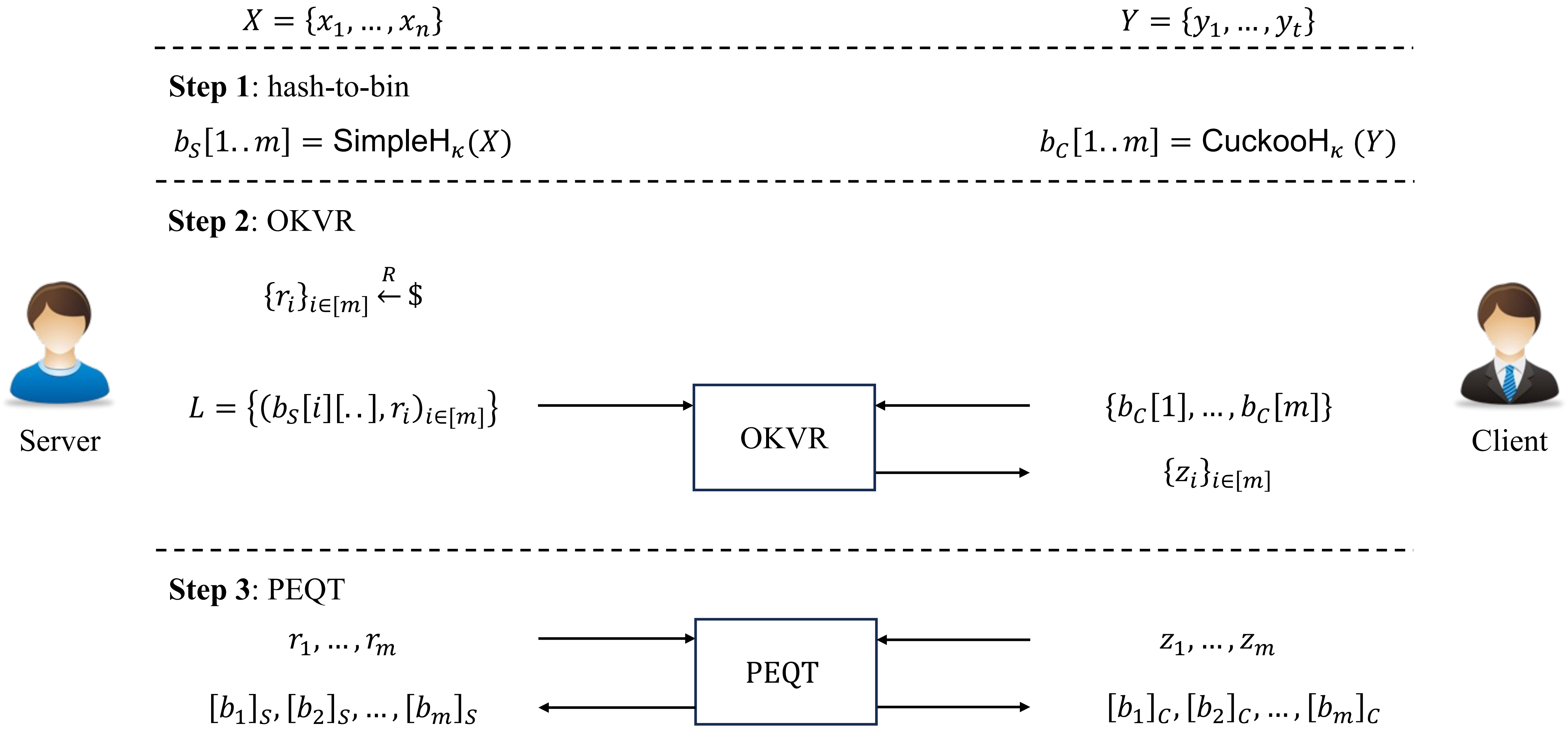
If $q = k_j$, then $z = v_j$. Otherwise, $z = \perp$

OKVR is for batch queries, protects server privacy, and client knows a random value if the query is not in the key-value pair.

3.4 OKVR Construction



3.4 UCPSI from OKVR

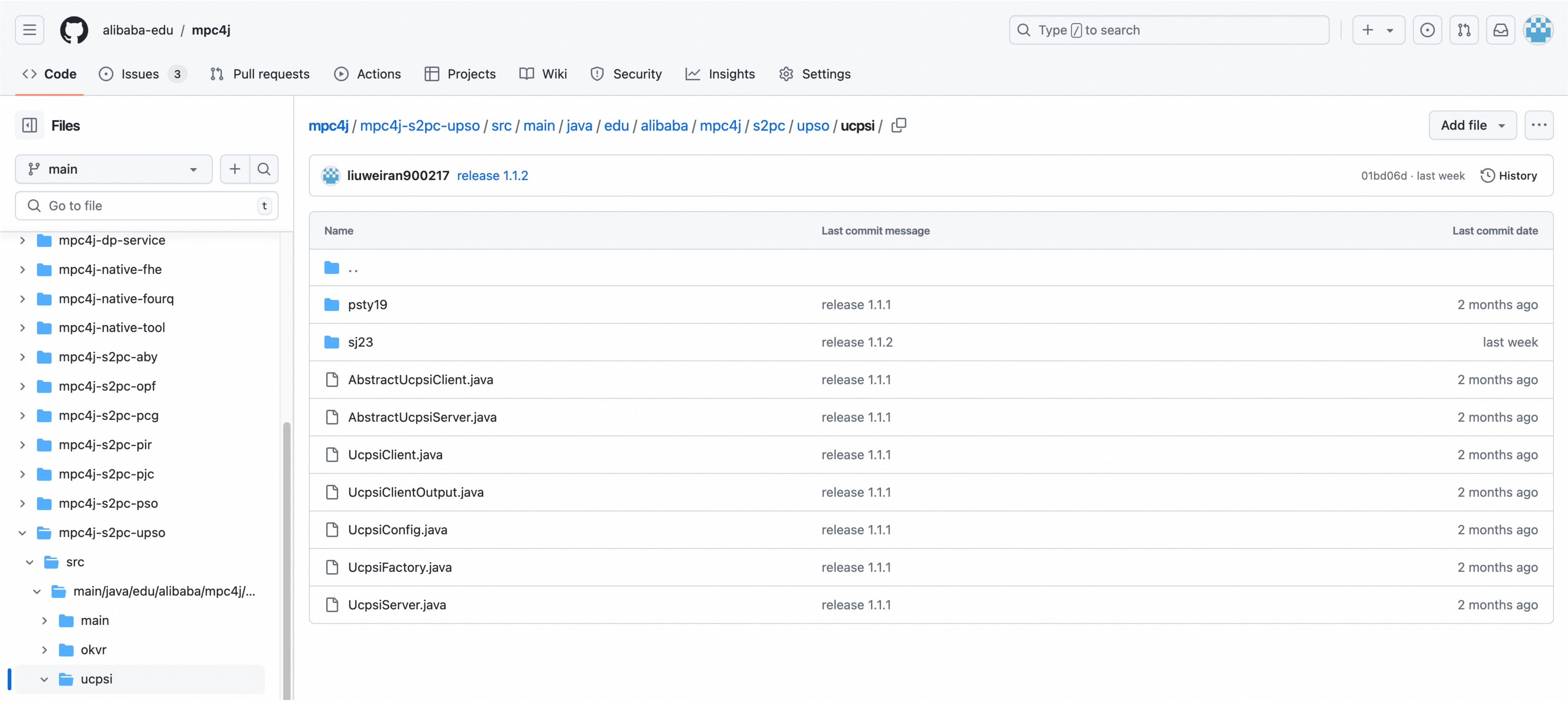


Outline

04 | Implementation and Performance

5.1 Implementation

We provide a unified (U)CPSI implementation, including known (U)CPSI protocols, with optimizations like Silent OT.



Remark: The source code is available at https://github.com/alibaba-edu/mpc4j (current version 1.1.2).

5.2 Performance

The best result   The second-best result

Parameter		Protocol	Communication (MB)		Time LAN (s)		Time WAN (s)		Time Mobile (s)	
n	*t*		Setup	Online	Setup	Online	Setup	Online	Setup	Online
2^{20}	2^4	PSTY19	0.308	32.859	0.100	7.020	2.224	18.035	5.162	356.548
		CGS22	0.317	30.978	0.117	10.866	2.581	21.736	5.377	340.301
		RS21	1.882	33.087	0.137	7.941	3.386	19.778	21.067	358.152
		SJ23-C1	1.994	5.938	7.996	4.232	10.380	8.567	18.581	58.258
		SJ23-C2	3.958	18.608	2.750	2.252	7.599	9.669	40.451	164.546
		UCPSI-2H	18.172	1.872	82.666	3.655	87.271	7.310	236.127	23.341
		UCPSI-3H	18.172	2.457	58.875	1.937	64.045	5.610	214.787	26.816
	2^8	PSTY19	0.405	33.057	0.077	6.965	2.188	17.816	5.787	358.192
		CGS22	0.415	31.196	0.075	11.674	2.575	22.412	6.235	343.088
		RS21	1.980	33.278	0.134	8.028	3.386	19.117	21.935	360.385
		SJ23-C1	1.994	5.938	8.265	3.843	9.812	8.752	18.284	57.751
		SJ23-C2	3.957	18.609	2.828	2.021	7.565	9.247	41.127	165.253
		UCPSI-2H	18.270	3.336	113.585	2.616	117.690	6.843	267.438	35.475
		UCPSI-3H	18.269	3.921	80.066	1.521	83.482	6.012	235.190	39.270
	2^{12}	PSTY19	0.687	34.028	0.175	8.221	2.350	18.510	8.380	365.269
		CGS22	0.697	32.537	0.110	12.426	2.678	23.398	8.767	353.945
		RS21	2.352	34.080	0.186	8.521	3.459	20.147	25.007	369.325
		SJ23-C1	4.854	9.669	6.393	11.932	8.955	17.296	41.982	96.145
		SJ23-C2	3.958	18.607	2.715	1.917	7.960	9.164	40.486	164.379
		UCPSI-2H	18.551	17.616	191.363	5.157	199.714	12.317	350.308	158.070
		UCPSI-3H	18.551	25.149	161.069	6.254	165.943	14.949	315.529	222.685

5.2 Performance

The best result   The second-best result

Parameter		Protocol	Communication (MB)		Time LAN (s)		Time WAN (s)		Time Mobile (s)	
n	*t*		Setup	Online	Setup	Online	Setup	Online	Setup	Online
2^{22}	2^4	PSTY19	0.308	130.148	0.090	31.202	2.237	62.721	5.067	1405.807
		CGS22	0.318	122.418	0.095	61.104	2.591	91.548	5.343	1352.874
		RS21	1.882	130.376	0.133	34.840	3.343	67.732	20.567	1407.752
		SJ23-C1	5.227	6.618	102.531	10.102	103.283	15.881	112.038	70.543
		SJ23-C2	3.958	42.674	11.998	3.728	16.224	15.202	44.846	369.253
		UCPSI-2H	18.172	2.856	287.176	7.330	295.251	11.241	443.559	36.117
		UCPSI-3H	18.172	2.668	323.034	7.627	316.533	11.544	483.251	35.354
	2^8	PSTY19	0.405	130.346	0.077	31.955	2.255	64.689	5.838	1407.101
		CGS22	0.415	122.636	0.080	64.442	2.670	93.478	6.361	1357.350
		RS21	1.980	130.567	0.318	35.146	3.331	64.838	21.478	1410.698
		SJ23-C1	5.226	6.618	103.195	9.663	101.296	15.717	111.341	70.589
		SJ23-C2	3.957	42.673	11.075	3.555	16.572	15.720	45.625	368.650
		UCPSI-2H	18.269	5.583	398.922	5.785	399.003	10.471	557.363	57.775
		UCPSI-3H	18.269	7.291	287.933	5.425	297.677	10.449	446.333	71.604
	2^{12}	PSTY19	0.687	131.322	0.281	35.768	2.355	66.032	8.363	1416.060
		CGS22	0.697	123.982	0.112	66.434	2.633	101.694	8.766	1369.970
		RS21	2.352	131.369	0.162	39.705	3.584	69.666	25.332	1422.936
		SJ23-C1	5.227	6.618	101.144	10.897	104.207	16.169	112.113	71.405
		SJ23-C2	3.957	42.675	11.639	3.464	16.575	15.108	44.739	368.319
		UCPSI-2H	18.551	33.130	623.735	13.086	631.618	20.526	795.836	295.674
		UCPSI-3H	18.551	25.361	672.840	12.123	674.243	19.188	843.413	230.016

Thanks for Your Attention!

Any Questions?

https://eprint.iacr.org/2023/1636

Weiran Liu
weiran.lwr@alibaba-inc.com