An Efficient Toolkit for Computing Third-party Private Set Intersection

Kai Chen[1,2] Yongqiang Li[1,2] Mingsheng Wang[1,2]

[1]Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

December, 2024

Contents

Background

Definition of Third-party Private Set Intersection

Waiting for a set X of size N_X from party P_1 and Y of size N_Y from party P_2, the protocol give intersection $I = X \cap Y$ to party P_3.

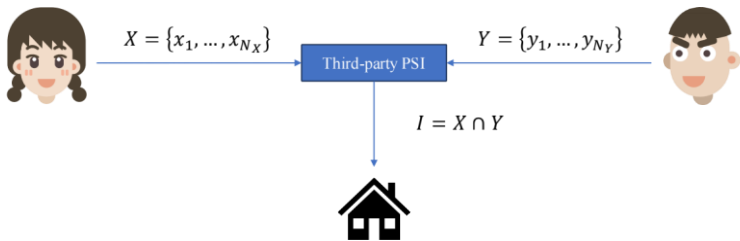


Figure 1: Third-party Private Set Intersection

Motivation

Scenario: pandemic-related disease outbreak

Public health authorities need to rapidly identify potential asymptomatic carriers.

- Public health authority acts as the third-party entity.
- Premises maintaining visitor records act as the participating entities.

Third-PSI allows public health authority to access the data of who were present at specific location during particular time, fulfilling mandate while preserving privacy

Previous Work

Protocol	Comm.	Comp.	Rounds
Commutative Cipher Based PSI	$O(n)$	$O(n)$	4
Key Agreement Based PSI	$O(n^{1.5+o(1)})$	$O(n^{2.5+o(1)})$	3
Key Agreement Based PSI Mod$_1$	$O(n)$	$O(n^{1.5+o(1)})$	3
Key Agreement Based PSI Mod$_2$	$O(n^{1+\delta})$[1]	$O(n^{1+\epsilon})$[2]	3
Our HE Based PSI	$O(n)$	$O(n)$	2
Our OPRF Based PSI	$O(n)$	$O(n)$	2

[1] $\delta > 0$ denotes the security parameter associated with the KA protocol
[2] $0 < \epsilon < 1$ denotes any positive constant

Table 1: Comparisons of communication and computation costs of third-party PSI protocols

Preliminaries

Definition of Bloom Filters

A Bloom Filter is a data structure to represent data and perform membership test.

- Initially represented by a bit vector of length B, with all bits set to 0.
- Employ k hash functions, denoted as $h_i : \{0, 1\}^* \to \{1, \ldots, B\}$ for $i \in [k]$.
- Insert an item $x \in X$:
  - Evaluate $h_1(x), \ldots, h_k(x)$ and bits at the corresponding indices are set from 0 from 1.
- Verify an item $x \in X$:
  - If $\{\mathrm{BF}[h_i(x)] = 1\}_{i \in [k]}$, then x is represented in the BF.

Preliminaries

Definition of Encrypted Bloom Filters

- An Encrypted Bloom Filter has B entries where each entry is defined as

$$\mathrm{EBF}[i] = \mathrm{Enc}_{pk}(\mathrm{BF}[i])$$

Definition of Inverted Bloom Filters

- An Inverted Bloom Filter has B entries where each entry is defined as

$$\mathrm{IBF}[i] = \begin{cases} 1 & \text{if } \mathrm{BF}[i] = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we could denote EIBF as an Encrypted, Invrted Bloom Filter as before.

Preliminaries

Definition of Homomorphic Encryption

An Additively Homomorphic Encryption satisfies the following properties:

- Let $+_{\mathrm{H}}$ denote the homomorphic addition, then $\mathrm{Dec}_{sk}(\tilde{x} +_{\mathrm{H}} \tilde{y}) = x + y$, where $\tilde{x} = \mathrm{Enc}_{pk}(x)$ and $\tilde{y} = \mathrm{Enc}_{pk}(y)$.
- Let r denote a scalar, then $\mathrm{Dec}_{sk}(\tilde{x} \cdot r) = x \cdot r$.

Definition of Oblivious Pseudorandom Function

An Oblivious Pseudorandom Function is a widely used protocol where the receiver selects a random key k and the sender takes x as input and obtains $F_k(x)$.

Preliminaries

Definition of Cuckoo Hashing

Cuckoo hashing employs γ hash functions $h_1, \ldots, h_\gamma$ to map n items into $b = \epsilon n$ bins, along with an auxiliary stash.

- Insert an item $x \in X$: Choose any empty bin of $B_{h_1(x)}, \ldots, B_{h_\gamma(x)}$. If all bins are occupied, a bin $B_{h_i(x)}$ is randomly chosen among the γ bins, and the prior item y in $B_{h_i(x)}$ is relocated to a new bin $B_{h_k(x)}$, where $k \neq i$.

Definition of Simple Hashing

Simple hashing uses γ hash functions $h_1, \ldots, h_\gamma : \{0,1\}^* \to [b]$ to map n items to b bins $B_1, \ldots, B_b$:

$$\Pr[\exists \text{ bin with } \geq \rho \text{ items}] \leq b \left[\sum_{i=\rho}^{n} \binom{n}{i} \cdot \left(\frac{1}{b}\right)^i \cdot \left(1 - \frac{1}{b}\right)^{n-i} \right]$$

HE-based Third-party Private Set Intersection

$X = \{x_1, \ldots, x_n\}$

EIBF_{1,pk_3}

EIBF_{1,pk_3}

$Y = \{y_1, \ldots, y_n\}$

$\{c_i = c_1^i +_H \cdots +_H c_k^i\}_{i\in[n]}$

$\{r_i \leftarrow \mathbb{Z}_N\}_{i\in[n]}$

$\{p_i = r_i \cdot c_i +_H \mathrm{Enc}_{pk_3}(y_i)\}_{i\in[n]}$

$\{(p_i, c_i)\}_{i\in[n]}$

$I = \mathrm{Dec}_{sk_3}(p_i) \mid \mathrm{Dec}_{sk_3}(c_i) = 0$

HE-based Third-party Private Set Intersection-Cardinality

$X = \{x_1, \ldots, x_n\}$

$EIBF_{1,Pk_3}$

$EIBF_{1,Pk_3}$

$Y = \{y_1, \ldots, y_n\}$

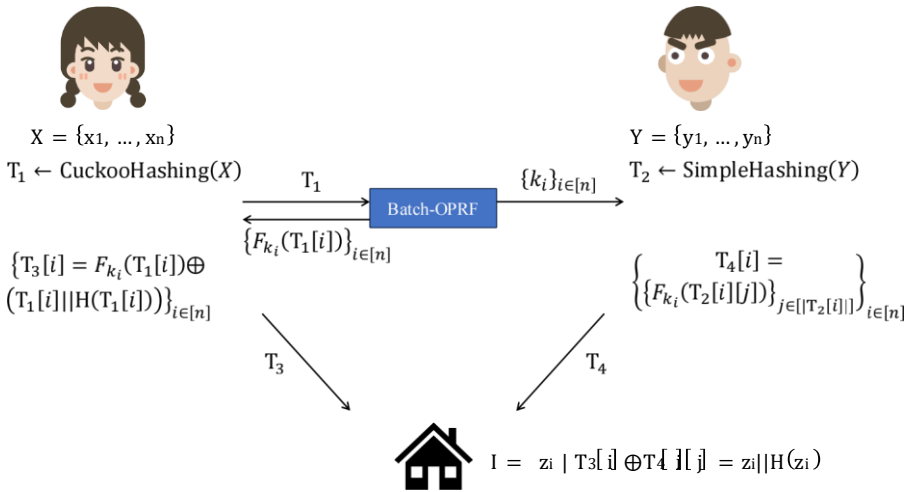
$\{c_i = c_1^i +_H \cdots +_H c_k^i\}_{i \in [n]}$

$\{c_i\}_{i \in [n]}$

$d = 0$

$\{\text{If } Dec_{sk_3}(c_i) = 0 \colon d = d + 1\}_{i \in [n]}$

OPRF-based Third-party Private Set Intersection



OPRF-based Third-party Private Set Intersection-Cardinality

$X = \{x_1, \ldots, x_n\}$

$T_1 \leftarrow \text{CuckooHashing}(X)$

$Y = \{y_1, \ldots, y_n\}$

$T_2 \leftarrow \text{SimpleHashing}(Y)$

T_1

$\{k_i\}_{i\in[n]}$

Batch-OPRF

$\{F_{k_i}(T_1[i])\}_{i\in[n]}$

$\{T_3[i] = F_{k_i}(T_1[i])\}_{i\in[n]}$

$\left\{T_4[i] = \left\{F_{k_i}(T_2[i][j])\right\}_{j\in[|T_2[i]|]}\right\}_{i\in[n]}$

T_3

T_4

$d = 0$

$\{\text{If } T_3[i] = T_4[i][j]\colon\ d = d + 1\}_{i\in[n]}$

Results of Third-party Private Set Intersection

Protocol		2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
$\prod^{thir_d,HE}_{PSI}$	P_1	0.3471706	1.41192	5.623202	22.16618	254.714	-
	P_2	0.115061	0.460324	1.84586	7.28272	29.197153	-
	P_3	0.011339	0.036721	0.131207	0.519475	2.01733	-
	Sum	0.462232	1.87224	7.46906	29.4489	283.911	-
$\prod^{thir_d,OPRF}_{PSI}$	P_1	0.080859	0.170165	0.532038	2.01944	8.08216	33.7933
	P_2	0.09861	0.210888	0.640433	2.77528	20.3685	85.8857
	P_3	0.000921	0.001833	0.006039	0.022515	0.08662	0.344163
	Sum	**0.099531**	**0.212721**	**0.646472**	**2.797795**	**20.45512**	**86.229863**

Table 2: Comparisons of total runtime (in seconds) and respective runtime (in seconds)

Results of Third-party Private Set Intersection-Cardinality

Protocol		2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
$\Pi^{third,HE}_{PSI\text{-}card}$	P_1	0.342722	1.43299	5.66005	22.2455	254.925	-
	P_2	0.0964466	0.387186	1.613198	6.380176	25.572383	-
	P_3	0.0080178	0.0197356	0.0751962	0.2737504	1.0439745	-
	Sum	0.439169	1.82018	7.27325	28.6257	280.497	-
$\Pi^{third,OPRF}_{PSI\text{-}card}$	P_1	0.080666	0.171368	0.53547	2.03603	8.12223	33.8245
	P_2	0.098735	0.210805	0.64066	2.77604	20.4886	84.9362
	P_3	0.00132	0.003163	0.012608	0.041021	0.153384	0.588081
	Sum	**0.100055**	**0.213968**	**0.653268**	**2.817061**	**20.641984**	**85.524281**

Table 3: Comparisons of total runtime (in seconds) and respective runtime (in seconds)

Thanks!

chenkai1621@iie.ac.cn