An Introduction to Threshold PSI

Xinpeng Yang

July 17, 2023

Contents

What is threshold PSI

Multiparty PSI enables n parties to compute the intersection of their n private data sets, without revealing any additional information.

Threshold PSI is able to compute the elements that appear at least k times in n sets

Threshold PSI

There are n parties $P_1, \cdots, P_n$ where P_1 is the leader and $k \in [1, n-1]$ denotes the threshold.
Input: For each $i \in [n]$, P_i inputs a set X_i of size m.
Output: For each $x \in X_1$, let $q_x = |\{i : x \in X_i \text{ for } i \in \{2, \cdots, n\}\}|$, then, output $Y = \{x \in X_1 : q_x \geq k\}$ to P_1.

Simple approach

We can compute the result as follow

1. select subset $s \subseteq \{1, 2, \cdots, n\}$ and $|s| \geq k$
2. run multi-party PSI between X_j and get $X^s = \{x|x \in X_j, j \in s\}$
3. output $Y = \bigcap_{|s|>=k, s\subseteq[n]} X^s$

The computation cost is at least $C_n^k + C_n^{k+1} + \cdots + C_n^n$

inefficient and insecure !

Main challenges

Additional leakage

- Resist the collusion
- Can't leak which k parties have a same element
- Can't leak how many parties have a same element

Application

- Identifying High-Risk Individuals in the Spread of Disease
- Share ride
- Anonymous Voting and Consensus

Contents

Practical Multi-Party Private Set Intersection Protocols

TIFS 22

Preliminaries

Bloom Filter

A Bloom Filter, $BF = (BF[0], \cdots, BF[j], \cdots, BF[m-1])$ encodes a set S of length at most n into m bit string
chosen k hash function $h_i : \{0,1\}* \longrightarrow [0, 1, \cdots, m-1]$
for every $x \in S$, set $BF(h_i(x)) = 1$ where $i = 1, 2, \cdots, k$, the other slot is $\mathbf{0}$

Encrypted Bloom Filter

for $j \in 0, 1, \cdots, m-1$, $EBF[j] = Enc_{pk}(BF[j])$, where pk is a public key of a secret key sk

Preliminaries

Threshold Paillier PKE

- (t,n)-threshold version of the Paillier's scheme
- Additive Homomorphism
- At least t shares of decryption can reconstruct the plaintext

Preliminaries

SCP(Secure Comparison Protocol) Kerschbaum et al.

Given only their encrypted values Enc(x_0) and Enc(x_1) as input. The output is a single encrypted bit Enc(b) and the encryption scheme is additive homomorphic (here is Paillier PKE)

In their protocol, $\mathbb{Z}_p$ is represented by the upper half of the range [0, p - 1] as negative, that is $[\lceil \frac{p}{2} \rceil, p-1] \equiv [\lfloor -\frac{p}{2} \rfloor, -1]$

P_1 computes (a_1^1, a_2^1, a_3^1) = (Enc(1), Enc(0), Enc(c))
where

$$\text{Enc(c)} = (Enc(x_0)Enc(x_1))^{r_1} Enc(r_2) = Enc(r_1(x_0 - x_1) - r_2)$$

$$r_1 > r_2$$

Preliminaries

For every party $P_i, 2 \leq i \leq t$, selects $r_2 < r_1$ and flips a coin $b_i \in \{0, 1\}$, sends (a_1^i, a_2^i, a_3^i) to P_{i+1} where

$$\begin{aligned} a_1^i &= a_{1+b}^{i-1}\, Enc(0) \\ a_2^i &= a_{2-b}^{i-1}\, Enc(0) \\ a_3^i &= (a_3^{i-1})^{r_1} Enc(r_2) \end{aligned}$$

All parties $P_i, 2 \leq i \leq t$, jointly decrypt a_t^3 to decide the result.

If $a_t^3 < 0$ then $a_t^1 = \text{Enc}(1)$, that is $[\, x_0 \leq x_1] = 1$, else $a_t^1 = \text{Enc}(0)$.

Main method

Local EBFs generation

Each client $P_i, 1 \leq i \leq t-1$

1. Computes their Bloom filter of their private data set S_i , where $1 \leq i \leq t-1$
2. Computes their encrypted Bloom filter EBF_i by encrypting each element of $BFi[j]$ using pk
3. Forward their EBF_i to the server P_t

Main method

Set Intersection generation by the server

The server P_t:

- Computes k hash values of each element $y_j \in S_t$, and for each party P_i
  - Computes $C_d^{i,j} = EBF_i[h_d(y_j)]$ for $d \in \{1, 2, \cdots, k\}$
  - Computes $C^{i,j} = \mathrm{ReRand}(C_1^{i,j} +_H C_2^{i,j} +_H \cdots +_H C_k^{i,j})$
  - Run **SCP** to compare $C^{i,j}$ and Enc(k) and get the output Enc($\alpha^{i,j}$)
  - If Dec($C^{i,j}$)=k then $\alpha^{i,j}$ will be **1** else **0**

Main method

Set Intersection generation by the server

The server P_t:

- Computes $\mathrm{Enc}(\alpha^j) = \mathrm{ReRand}(\alpha^{1,j} +_H \alpha^{2,j} +_H \cdots +_H \alpha^{t-1,j})$
- Run **SCP** to compare $\mathrm{Enc}(\alpha^j)$ and $\mathrm{Enc}(\mathcal{T})$ and get the output $\mathrm{Enc}(\beta^j)$
- $\mathrm{Enc}(\beta^j) = \mathrm{ReRand}(\mathrm{Enc}(\beta^j))$
- Perform joint decryption of $\mathrm{Enc}(\beta^j)$
- If $\beta^j = 1$ and then adds y_j to Y
- Repeats for every $y_j \in S_t$

Analysis

Communication complexity

the set size is n and the threshold of Paillier PKE is l and the server needs to receive message from t parties and the size of bloom filter is $O(\lambda n)$

- $O(n \cdot \kappa \cdot l \cdot t)$ for server
- $O(n \cdot \kappa \cdot max(t, \lambda))$ for client
- $O(t)$ for communication rounds

however when $l = \frac{t}{2}$ the communication cost is not linear with the number of parties

Computation complexity

- $O(n \cdot t)$ for server
- $O(n \cdot max(t, \lambda))$ for client

Result

Evaluate the run time performance

INTEL CORE I7-1065G7 processor at 1.30GHz 8 cores 16GB

		$n = 2^2$	$n = 2^4$	$n = 2^6$
$t = 3$	$\ell = 1$	0.50 ± 0.02	1.92 ± 0.01	7.62 ± 0.02
	$\ell = 2$	0.57 ± 0.07	2.19 ± 0.28	8.73 ± 1.14
$t = 4$	$\ell = 2$	0.82 ± 0.00	3.28 ± 0.01	13.12 ± 0.02
	$\ell = 3$	0.91 ± 0.10	3.66 ± 0.39	14.60 ± 1.51
$t = 6$	$\ell = 3$	1.52 ± 0.03	5.83 ± 0.02	23.34 ± 0.03
	$\ell = 5$	1.75 ± 0.24	6.86 ± 1.06	27.47 ± 4.24
$t = 8$	$\ell = 4$	2.30 ± 0.01	9.17 ± 0.02	37.07 ± 0.51
	$\ell = 7$	2.81 ± 0.52	11.20 ± 2.08	44.94 ± 8.08

mean run time results in seconds for threshold PSI averaged over 10 runs

secure parameter $\kappa = 1024$

Contents

Efficient Linear Multiparty PSI and Extensions to Circuit/Quorum PSI

CCS 21

Preliminaries

Circuit-based PSI

The problem of circuit PSI was introduced in the 2 party setting and enables parties P_1 and P_2, with their private input sets X and Y, respectively, to compute $f(X \cap Y)$, where f is any symmetric function
This also applies to n parties

It allows to keep the intersection $X \cap Y$ secret from the parties while allowing to securely compute $f(X \cap Y)$

Applications: cardinality, set intersection sum and threshold cardinality/intersection

Preliminaries

Multiparty Functionalities

Functionality	Communication	Rounds
$\mathsf{RandomF}^{n,t}(\ell)$	$\lceil \frac{\ell}{n-t} \rceil n(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$ $< 2\ell(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$	1
$\mathsf{MultF}^{n,t}([a],[b])$ (amortized cost)	$2(\frac{2n}{n-t}+3)(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$ $< 14(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$	5
$\mathsf{Reveal}^{n,t}([a])$	$(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$	1
$\mathsf{ConvertShares}^{n,t}(\langle a \rangle)$ (amortized cost)	$2(\frac{n}{n-t}+1)(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$ $< 6(n-1)\lceil \log \lvert\mathbb{F}\rvert \rceil$	3

Secret Sharing Scheme:
(n,t) secret sharing for a as $[a]$ and additive secret sharing for a as $\langle a \rangle$

Preliminaries

Multiparty Functionalities

- $\mathsf{RandomF}^{n,t}(l)$: Generate $[r_1], [r_2], \cdots, [r_l]$ for uniform elements $r_1, r_2, \cdots, r_l$ in $\mathbb{F}$
- $\mathsf{MultF}^{n,t}([a],[b])$: Takes $[a],[b]$ for $a, b \in \mathbb{F}$ and output $[a \cdot b]$
- $\mathsf{Reveal}^{n,t}([a])$: Takes $[a]$ where $a \in \mathbb{F}$ and outputs a to P_1
- $\mathsf{ConvertShares}^{n,t}(\langle a \rangle)$: Takes $\langle a \rangle$ where $a \in \mathbb{F}$ and outputs $[a]$

Preliminaries

Weak Private Set Membership $\mathcal{F}_{wPSM}^{\beta,\sigma,N}$

P_1 and P_2 are the receiver and the sender respectively
Receiver P_1's Inputs: The queries $q_1, q_2, \cdots, q_\beta \in \{0,1\}^\sigma$
Sender P_2's Inputs: Sets $\{X_j\}$ $j \in \{1, 2, \cdots, \beta\}$ $X_j[i] \in \{0,1\}^\sigma$ and $\Sigma_j |X_j| = N$
Output:

- For each $j \in 1, 2, \cdots, \beta$, sample w_j uniformly from $\{0,1\}^\sigma$
- For each $j \in 1, 2, \cdots, \beta$, if $q_j \in X_j$, set $y_j = w_j$, else sample y_j uniformly from $\{0,1\}^\sigma$
- Return $\{y_j\}$ to P_1 and $\{w_j\}$ to P_2

$\mathcal{F}_{wPSM}^{\beta,\sigma,N}$ is similar in spirit to the batch oblivious programmable PRF

Preliminaries

Equality Test $\mathcal{F}_{EQ}^{\sigma}$

Input: parties P_1 and P_2 have $a, b \in \{0,1\}^{\sigma}$
Output: receive **boolean** shares of the bit $r_a \oplus r_b = 1$ if $a = b$ and $r_a \oplus r_b = 0$ otherwise, as the output

Boolean to Arithmetic Share Conversion $\mathcal{F}_{B2A}^{\mathbb{F}_p}$

Input: parties P_1 and P_2 boolean shares $\langle b \rangle_1^B$ and $\langle b \rangle_2^B$
Output: receive additive shares $\langle x \rangle_1^B$ and $\langle x \rangle_2^B$ respectively for $x = b$

Main method

Quorum PSI

Input: Each party P_i has input set $X_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$

Protocol

1. **Hashing**:
   P_1 does stash-less cuckoo hashing on X_1 using h_1, h_2, h_3 to generate Table_1.
   For $i \in \{2, 3 \cdots, n\}$ P_i does simple hashing of X_i using h_1, h_2, h_3 into $\mathrm{Table}_\mathrm{i}$

Main method

2. **Invoking $\mathcal{F}_{wPSM}^{\beta,\sigma,N}$ functionality**:
   For each $i \in \{2, 3, \cdots, n\}$, P_1 and P_i invoke the $\mathcal{F}_{wPSM}^{\beta,\sigma,N}$
   - P_i is the sender with inputs $\{\mathrm{Table_i[j]}\}$ and P_1 is the receiver with inputs $\{\mathrm{Table_1[j]}\}$ for $j \in \{1, 2, \cdots, \beta\}$
   - P_i receives the outputs $\{w_j\}$ and P_1 receives $\{y_j\}$ for $j \in \{1, 2, \cdots, \beta\}$

3. **Invoking the $\mathcal{F}_{EQ}^{\sigma}$ functionality**:
   For each $i \in \{2, \cdots, n\}$ and for each $j \in \{1, \cdots, \beta\}$, P_1 and P_i invoke the $\mathcal{F}_{EQ}^{\sigma}$ functionality as follows:
   - P_1 and P_i send their inputs y_{ij} and w_{ij} resp., and receive **Boolean** shares $\langle eq_{ij}\rangle_1^{\beta}$ and $\langle eq_{ij}\rangle_i^{\beta}$ resp., as outputs

Main method

4. **Invoking $\mathcal{F}_{B2A}^{\mathbb{F}_p}$ functionality**:
   For each $i \in \{2, \cdots, n\}$ and for each $j \in \{1, \cdots, \beta\}$, P_1 and P_i invoke the $\mathcal{F}_{B2A}^{\mathbb{F}_p}$ functionality as follows:
   - P_1 and P_i send their inputs eq_{ij} and eq_{ij} resp., and receive **Additive** shares $\langle f_{ij} \rangle_1$ and $\langle f_{ij} \rangle_i$ resp., as outputs

5. **Converting to (n,t) shares**:
   For each $j \in \{1, 2, \cdots \beta\}$,
   - P_1 computes $\langle a_j \rangle_1 = \Sigma_{i=2}^{n} \langle f_{ij} \rangle_1$ and for each $i \in \{2, \cdots n\}$, P_i sets $\langle a_j \rangle_i = \langle f_{ij} \rangle_1$
   - $P_1, \cdots, P_n$ compute $[a_j] \leftarrow \text{ConvertShares}^{n,t}(\langle a_j \rangle)$

Main method

Weak Comparison Protocol

Parameters:

There are n parties $P_1, \cdots, P_n$ with (n,t) shares $[a]$ and define polynomial ψ

$$\psi(x) = \begin{cases} x \cdot (x-1) \cdot (x-2) \cdots (x-(k-1)) & \text{if } k < \frac{n}{2} \\ (x-k) \cdot (x-(k+1)) \cdots (x-n) & \text{if } k \geq \frac{n}{2} \end{cases}$$

Input: Each P_i inputs its (n,t) shares $[a]_i$

Protocol:

- **Pre-Process**
  - $P_1, \cdots, P_n$ run: $[s_1], \cdots, [s_J] \leftarrow \mathrm{RandomF}^{n,t}(\mathrm{J})$

Main method

- **Evaluating the polynomial**
  - invoke $\mathrm{MultF}^{n,t}$ to compute all the required $[a^i]$ followed by scalar multiplications and additions to compute $[\psi(a)]$
  - For each $j \in \{1, \cdots, J\}$
    - $[v_j] \leftarrow \mathrm{MultF}^{n,t}([\psi(a), s_j])$
    - $v_j \leftarrow \mathrm{Reveal}^{n,t}([v_j])$

Output:
if $k < \frac{n}{2}$ and return P_1 **1** else **0**
if $k > \frac{n}{2}$ and return P_1 **0** else **1**
Other parties get no output

Analyze

Communication complexity

$O(nm\kappa(\lambda + \kappa \log n))$

Result

Evaluate the run time performance

A single machine with 64-core Intel Xeon 2.6GHz CPU and 256GB RAM

n	4			5			10			15		
m	2^{12}	2^{16}	2^{18}	2^{12}	2^{16}	2^{18}	2^{12}	2^{16}	2^{18}	2^{12}	2^{16}	2^{18}
Run-time LAN (s)	1.46	2.91	9.32	1.62	3.10	9.49	2.19	4.12	11.27	2.26	4.54	13.12
Run-time WAN (s)	7.10	13.74	34.04	6.98	15.44	39.34	7.88	23.08	74.02	8.14	31.28	108.36
Total Communication (MB)	16.98	209.86	874.23	24.64	290.68	1166.28	55.44	667.73	2627.01	86.24	1038.68	4086.45
Client Communication (MB)	5.66	69.95	291.41	6.16	72.67	291.57	6.16	74.19	291.9	6.16	74.19	291.89

Run-time in seconds and communication in MB for qPSI expect for Weak Comparison Protocol

Whole protocol:
$t = 7, m = 2^{16}$ and any $k \leq 14$ for 15 parties
5.49s and 37.85s in LAN and WAN setting respectively

Contents

End

Thanks for your listening.

End

Q&A