Labeled PSI from Fully Homomorphic Encryption with Malicious Security.pdf

论文理解与样例流程

1. 主要流程

1.1 Full Labeled PSI protocol (sender's offline pre–processing)

Input: Receiver inputs set $Y \subset \{0,1\}^*$ of size N_Y; sender inputs set $X \subset \{0,1\}^*$ of size N_X. N_X, N_Y are public. κ and λ denote the computational and statistical security parameters, respectively.
Output: The receiver outputs $Y \cap X$; the sender outputs $\perp$.

1. **[Sender's OPRF]** The sender samples a key k for the [31] OPRF $F : \{0,1\}^* \to \{0,1\}^\kappa$. The sender updates its set to be $X' = \{H(F_k(x)) : x \in X\}$. Here H is a random oracle hash function with a range of $\sigma = \log_2(N_X N_Y) + \lambda$ bits which is sampled using a coin flipping protocol.
2. **[Hashing]** The parameter m is agreed upon such that cuckoo hashing N_Y balls into m bins succeed with probability $\geq 1 - 2^{-\lambda}$. Three random hash function $h_1, h_2, h_3 : \{0,1\}^\sigma \to [m]$ are agreed upon using coin flipping. The sender inserts all $x \in X'$ into the sets $\mathcal{B}[h_1(x)], \mathcal{B}[h_2(x)], \mathcal{B}[h_3(x)]$.
3. **[Choose FHE parameters]** The parties agree on parameters (n, q, t, d) for an IND-CPA secure FHE scheme. They choose t, d to be large enough so that $d \log_2 t \geq \sigma$.
4. **[Choose circuit depth parameters]** The parties agree on the split parameter $B < O(|Y|/m)$ and windowing parameter $w \in \{2^1, 2^2, \ldots, 2^{\log_2 B}\}$ as to minimize the overall cost.
5. **[Pre-Process X]**
   (a) **[Splitting]** For each set $\mathcal{B}[i]$, the sender splits it into α subsets of size at most B, denoted as $\mathcal{B}[i,1], \ldots, \mathcal{B}[i,\alpha]$.
   (b) **[Computing Coefficients]**
      i. **[Symmetric Polynomial]** For each set $\mathcal{B}[i,j]$, the sender computes the symmetric polynomial $S_{i,j}$ over $\mathbb{F}_{t^d}$ such that $S_{i,j}(x) = 0$ for $x \in \mathcal{B}[i,j]$.
      ii. **[Label Polynomial]** If the sender has labels associated with its set, then for each set $\mathcal{B}[i,j]$, the sender interpolates the polynomial $P_{i,j}$ over $\mathbb{F}_{t^d}$ such that $P_{i,j}(x) = \ell$ for $x \in \mathcal{B}[i,j]$ where ℓ is the label associated with x.
   (c) **[Batching]**
   View the polynomials $S_{i,j}$ as a matrix where i indexes the row. For each set of n/d rows (non-overlapping and contiguous), consider them as belonging to a single *batch*. For b-th batch and each j, take the k-th coefficient of the n/d polynomials, and batch them into one FHE plaintext polynomial $\overline{S}_{b,j,k}$.
   For Labeled PSI, perform the same batching on the label polynomials $P_{i,j}$ to form batched FHE plaintext polynomials $\overline{P}_{b,j}$.

Fig. 1. Full Labeled PSI protocol (sender's offline pre-processing).

1.2 Full Labeled PSI protocol continued (online phase)

7. **[Encrypt Y]**
   (a) **[Receiver's OPRF]** The receiver performs the interactive OPRF protocol of [31] using its set Y in a random order as private input. The sender uses the key k as its private input. The receiver learns $F_k(y)$ for $y \in Y$ and set $Y' = \{H(F_k(y)) : y \in Y\}$.
   (b) **[Cuckoo Hashing]** The receiver performs cuckoo hashing on the set Y' into a table $\mathcal{C}$ with m bins using h_1, h_2, h_3 has the hash functions.
   (c) **[Batching]** The receiver interprets $\mathcal{C}$ as a vector of length m with elements in $\mathbb{F}_{t^d}$. For the bth set of n/d (non-overlapping and contiguous) in $\mathcal{C}$, the receiver batches them into a FHE plaintext polynomial $\overline{Y}_b$.
   (d) **[Windowing]** For each $\overline{Y}_b$, the receiver computes the component-wise $i \cdot w^j$-th powers $\overline{Y}_b^{i \cdot w^j}$, for $1 \leq i \leq w-1$ and $0 \leq j \leq \lfloor \log_w(B) \rfloor$.
   (e) **[Encrypt]** The receiver uses $\mathsf{FHE.Encrypt}$ to encrypt each power $\overline{Y}^{i \cdot 2^j}$ and forwards the ciphertexts $c_{i,j}$ to the sender.
8. **[Intersect]** For the bth batch,
   (a) **[Homomorphically compute encryptions of all powers]** The sender receives the collection of ciphertexts $\{c_{i,j}\}$ and homomorphically computes a vector $\mathbf{c} = (c_0, \ldots, c_B)$, such that c_k is a homomorphic ciphertext encrypting $\overline{Y}_b^k$.
   (b) **[Homomorphically evaluate the dot product]** For each $\overline{S}_{b,1}, ..., \overline{S}_{b,\alpha}$, the sender homomorphically evaluates

$$z_{b,j} = \sum_{k=0}^{B} c_k \cdot \overline{S}_{b,j,k}$$

   and optionally performs modulus switching on the ciphertexts $z_{b,j}$ to reduce their sizes. All $z_{b,j}$ are sent back to the receiver. If Labeled PSI is desired, repeat the same operation for $\overline{P}$ and denote the returned ciphertexts $q_{b,j}$.
9. **[Decrypt and get result]** For the b-th batch, the receiver decrypts the ciphertexts $z_{b,1}, ..., z_{b,\alpha}$ to obtain $r_{b,1}, ..., r_{b,\alpha}$, which are interpreted as vectors of n/d elements in $\mathbb{F}_{t^d}$.
   Let $r_1^*, ..., r_\alpha^*$ be vectors of m elements in $\mathbb{F}_{t^d}$ obtained by concatenating $r_j^* = r_{1,j}^* || ... || r_{md/n,j}^*$. For all $y' \in Y'$, output the corresponding $y \in Y$ if

$$\exists j : r_j^*[i] = 0,$$

   where i is the index of the bin that y' occupies in $\mathcal{C}$.
   If Labeled PSI is desired, perform the same decryption and concatenation process on the $q_{b,j}$ ciphertexts to obtain the m element vectors $\ell_1^*, ..., \ell_\alpha^*$. For each $r_j^*[i] = 0$ above, output the label of the corresponding y to be $\ell_j^*[i]$.

Fig. 2. Full Labeled PSI protocol continued (online phase).

1.3 Native Theory

We now review the protocol of [12] in detail. Following the architecture of [43], their protocol instructs the receiver to construct a cuckoo hash table of its set Y. Specifically, the receiver will use three hash functions h_1, h_2, h_3, and a vector $B_R[0], \ldots, B_R[m]$ of $O(|Y|)$ bins. For each $y \in Y$, the receiver will place y in bin $B_R[h_i(y)]$ for some i such that all bins contain at most one item. The sender will perform a different hashing strategy. For all $x \in X$ and all $i \in \{1, 2, 3\}$, the sender places x in bin $B_S[h_i(x)]$. Note that each bin on the sender's side will contain $O(|X|/m)$ items with high proabibility when $|X| \gg m$. It then holds that the intersection of $X \cap Y$ is equal to the union of all bin-wise intersections. That is,

$$X \cap Y = \bigcup_j B_R[j] \cap B_S[j] = \bigcup_j \{y_j\} \cap B_S[j]$$

where y_j is the sole item in bin $B_R[j]$ (or a special sentinel value in the case that $B_R[j]$ is empty). The protocol then specifies a method for computing $\{y\} \cap B_S[j]$ using FHE. The receiver first sends an encryption of y, denoted as $[\![y]\!]$, to the sender who locally computes

$$[\![z]\!] := r \prod_{x \in B_S[j]} ([\![y]\!] - x)$$

2. 详细流程解析

2.1 符号定义

- X is the sender's set; Y is the receiver's set. We assume $|X| \gg |Y|$.
- σ is the length of items in X and Y.
- ℓ is the length of labels in Labeled PSI.
- n is the ring dimension in our FHE scheme (a power of 2); q is the ciphertext modulus; t is the plaintext modulus [22,21].
- d is the degree of the extension field in the SIMD encoding.
- m is the cuckoo hash table size.
- α is the number of partitions we use to split the sender's set X in the PSI protocol (following [12]).
- $[i, j]$ denotes the set $\{i, i+1, ..., j\}$, and $[j]$ is shorthand for the case $i = 1$.

2.1 本例参数定义

m	3
n	3
d	1
|Y|	3
|X|	4

2.2 经过cuckoo hash后两方的hash table

Receiver data：Y [1, 2, 3]

idx[1:m]	item
1	1
2	2
3	3

Sender data: X [1, 3, 4, 5] , 3个hash functions

idx[1:m]	item	item	item	item
1	5	1	3	4
2	3	4	1	5
3	5	1	4	3

说明： 从例子中，可以看出X与Y的交集是[1， 3]

2.4. 流程理解

2.4.1 Sender offline pre–processing: [Pre – Processing X]

(a) **[Splitting]** For each set $\mathcal{B}[i]$, the sender splits it into α subsets of size at most B, denoted as $\mathcal{B}[i,1],\ldots,\mathcal{B}[i,\alpha]$.

> 令 α=2， B=2，即分裂成α个子集，每个子集最多有B=2个item, 得到下图所示

idx[1:m]	α=1		α=2	
1	5	1	3	4
2	3	5	1	4
3	3	1	4	5

(b) **[Computing Coefficients]**

i. **[Symmetric Polynomial]** For each set $\mathcal{B}[i,j]$, the sender computes the symmetric polynomial $S_{i,j}$ over $\mathbb{F}_{t^d}$ such that $S_{i,j}(x) = 0$ for $x \in \mathcal{B}[i,j]$.

idx[1:m]	α=1	α=2
1	$S_{11} = [5, -6, 1]$	$S_{12} = [12, -7, 1]$
2	$S_{21} = [15, -8, 1]$	$S_{22} = [4, -5, 1]$
3	$S_{31} = [3, -4, 1]$	$S_{32} = [20, -9, 1]$

> $S_{ij} = aX^B + bX^{B-1} + \ldots + 1 = [a, b, \ldots, 1]$
>
> 例：$S_{11} = (x-5)(x-1) = 5-6x+x^2 = [5, -6, 1] = [axb+bx]$

(c) **[Batching]**

View the polynomials $S_{i,j}$ as a matrix where i indexes the row. For each set of n/d rows (non-overlapping and contiguous), consider them as belonging to a single *batch*. For b-th batch and each j, take the k-th coefficient of the n/d polynomials, and batch them into one FHE plaintext polynomial $\overline{S}_{b,j,k}$.

> 取n/d = 3, 那b的范围就是[1，m*d/n] = [1, 1]
>
> 例： $\overline{S}_{b,j,k}$, 取$j = 1, k = 0$, 则$\overline{S}_{b,1,0} = (S_{11}[0], S_{21}[0], S_{31}[0]) = (5, 15, 3)$

> 能过转换，可以得到以下矩阵视图

idx[1:m]	α=1	α=2
b=1	$S_{1,1,0}$ = (5, 15, 3)	$S_{1,2,0}$ = (12, 4, 20)
	$S_{1,1,1}$ = (−6, −8, −4)	$S_{1,2,1}$ = (−7, −5, −9)
	$S_{1,1,2}$ = (1, 1, 1)	$S_{1,2,2}$ = (1, 1, 1)

> 等同于以下矩阵
>
> $$S = \begin{bmatrix} (5,15,3) & (12,4,20) \\ (-6,-8,-4) & (-7,-5,-9) \\ (1,1,1) & (1,1,1) \end{bmatrix}$$

2.4.2 Receiver pre–processing

(c) **[Batching]** The receiver interprets $\mathcal{C}$ as a vector of length m with elements in $\mathbb{F}_{t^d}$. For the bth set of n/d (non-overlapping and contiguous) in $\mathcal{C}$, the receiver batches them into a FHE plaintext polynomial $\overline{Y}_b$.

(d) **[Windowing]** For each $\overline{Y}_b$, the receiver computes the component-wise $i \cdot w^j$-th powers $\overline{Y}_b^{i \cdot w^j}$, for $1 \le i \le w-1$ and $0 \le j \le \lfloor \log_w(B) \rfloor$.

(a) **[Homomorphically compute encryptions of all powers]** The sender receives the collection of ciphertexts $\{c_{i,j}\}$ and homomorphically computes a vector $\mathbf{c} = (c_0, \ldots, c_B)$, such that c_k is a homomorphic ciphertext encrypting $\overline{Y}_b^k$.

> 由于B=2，所以需要计算Y^0, Y^1, Y^2
>
> 通过计算，得到以下视图$\boldsymbol{c} = (c_0, ..., c_B)$

idx[1:m]	B=0, C_0	B=1, C_1	B=2, C_2
1	1^0	1^1	1^2
2	2^0	2^1	2^2
3	3^0	3^1	3^2

> 等同于以下矩阵

> c_0 c_1 c_2

$$C = \begin{bmatrix} 1,1,1 \\ 1,2,4 \\ 1,3,9 \end{bmatrix}$$

2.4.3 [Intersect] For the bth batch by Sender

(b) [**Homomorphically evaluate the dot product**] For each $\overline{S}_{b,1}, ..., \overline{S}_{b,\alpha}$, the sender homomorphically evaluates

$$z_{b,j} = \sum_{k=0}^{B} c_k \cdot \overline{S}_{b,j,k}$$

and optionally performs modulus switching on the ciphertexts $z_{b,j}$ to reduce their sizes. All $z_{b,j}$ are sent back to the receiver. If Labeled PSI is desired, repeat the same operation for $\overline{P}$ and denote the returned ciphertexts $q_{b,j}$.

*即计算$Z_{b,j}$ = C * S, 这里b=1, j = [1，2]*

$$z_{bj} = \begin{bmatrix} 1,1,1 \\ 1,2,4 \\ 1,3,9 \end{bmatrix} * \begin{bmatrix} (5,15,3) & (12,4,20) \\ (-6,-8,-4) & (-7,-5,-9) \\ (1,1,1) & (1,1,1) \end{bmatrix} = [z_{b1}, z_{b2}]$$

详细过程如下所示

$$z_{b1} = z_{b10} + z_{b11} + z_{b12} = (1*5, 1*15, 1*3) + (1*-6, 2*-8, 3*-4) + (1*1, 4*1, 9*1) = (0,4,0)$$

$$z_{b2} = z_{b20} + z_{b21} + z_{b22} = (1*12, 1*4, 1*20) + (1*-7, 2*-5, 3*-9) + (1*1, 4*1, 9*1) = (6,-2,2)$$

2.4.4 [Decrypt and get result]

[**Decrypt and get result**] For the b-th batch, the receiver decrypts the ciphertexts $z_{b,1}, ..., z_{b,\alpha}$ to obtain $r_{b,1}, ..., r_{b,\alpha}$, which are interpreted as vectors of n/d elements in $\mathbb{F}_{t^d}$.
Let $r_1^*, ..., r_\alpha^*$ be vectors of m elements in $\mathbb{F}_{t^d}$ obtained by concatenating $r_j^* = r_{1,j}^* || ... || r_{md/n,j}^*$. For all $y' \in Y'$, output the corresponding $y \in Y$ if

$$\exists j : r_j^*[i] = 0\,,$$

Decrypt $z_{b,j}$, obtain $r_{b,j}$

$r_{b,1}$ = [0, 4, 0], $r_{b,2}$ = [6, −2, 2]

由于b只有一个，且等于1，所以

$$r_1^* = r_{1,1}^* || ... || r_{md/n,1}^* = [0,4,0]$$

$$r_2^* = r_{1,2}^* || ... || r_{md/n,2}^* = [6,-2,2]$$

$$Y \cap X = [1,2,3] \cap [1,3,4,5] = r_1^* \quad or \quad r_2^* = [true, false, ture] or [false, false, false] = [1,0,1]$$

以上可得结果正确^_^

特别感谢邵航同学的大力指导

写的累死我了