Fuzzy Private Set intersection from Fuzzy Mapping

Ying Gao[1,2], **Lin Qi**[1], Xiang Liu[1], Yuanchao Luo[1], Longxin Wang[1]

School of Cyber Security and Technology, Beihang University
Zhongguancun Laboratory

December 8th

Asiacrypt 2024

Contents

1. Background
2. Our Main Idea
3. Instantiation of Fuzzy Mapping
4. Implementation

Contents

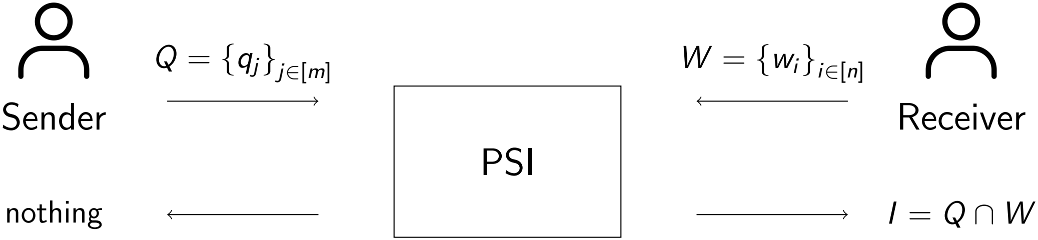
1 Background

2 Our Main Idea

3 Instantiation of Fuzzy Mapping

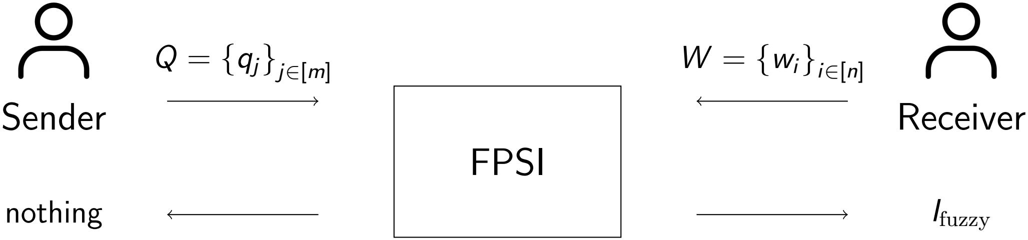
4 Implementation

Private Set intersection (PSI)



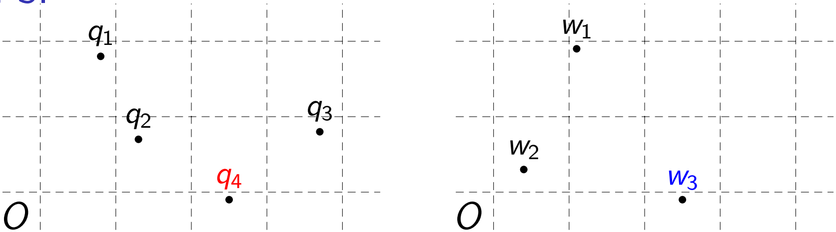
$$I = \{q_j : \exists\ i, \text{s.t. } w_i = q_j\}$$

Fuzzy Private Set intersection (FPSI)



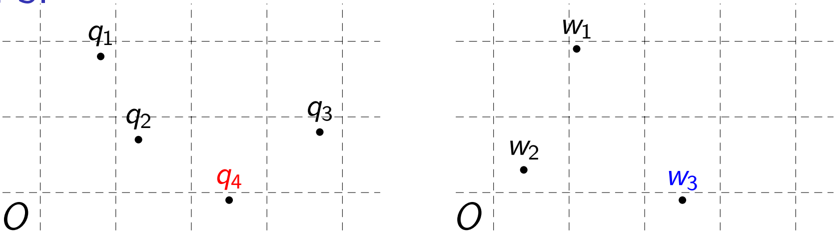
$$I_{\mathrm{fuzzy}} = \{q_j : \exists\ i, \text{s.t. } \mathsf{dist}\left(w_i, q_j\right) \le \delta\}$$

PSI and FPSI

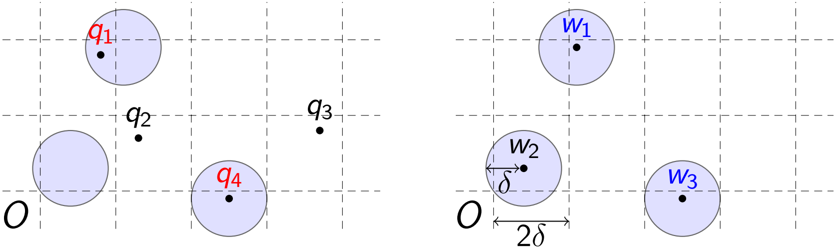


PSI: Receiver obtains $I = \{q_4\}$

PSI and FPSI



PSI: Receiver obtains $I = \{q_4\}$



FPSI: Receiver obtains $I_{\text{fuzzy}} = \{q_1, q_4\}$

Applications

- Searching on a database whose entries are not always accurate or full [FNP04]
- Building block for privacy-preserving biometric identification [UCK+21; CFR23; CLO24]
- Checking whether a user's password is similar to passwords that have been leaked online [GRS22; BP24]
- Illegal content detection [BP24]
- ...

Previous Work and Motivation

Previous works can be divided into two categories: FPSI for Hamming and $L_{\mathrm{p}\in[1,\infty]}$ distances.

1. Complexities of FPSI for Hamming distance have superlinear factors on input set sizes
   - Brutally traversing all pairs of inputs results in the $m \cdot n$ factor in complexities [FNP04; IW06; CH08; YSPW10; UCK+21; CFR23]
   - approximating I_{fuzzy} via multiple rounds of PSI results in the $\max\{m, n\} \log(\max\{m, n\})$ factor in complexities [CLO24]

2. Complexities of FPSI for $L_{\mathrm{p}\in[1,\infty]}$ distance have superlinear factors on input set sizes or dimension d
   - Spatial Hashing and Locality Sensitive Hashing result in the 2^d and $m \cdot n^\rho$ factors in complexities, respectively [GRS22; GRS23; BP24]

Previous Work and Motivation

Previous works can be divided into two categories: FPSI for Hamming and $L_{\mathrm{p}\in[1,\infty]}$ distances.

1. Complexities of FPSI for Hamming distance have superlinear factors on input set sizes
   - Brutally traversing all pairs of inputs results in the $m \cdot n$ factor in complexities [FNP04; IW06; CH08; YSPW10; UCK+21; CFR23]
   - approximating I_{fuzzy} via multiple rounds of PSI results in the $\max\{m, n\} \log(\max\{m, n\})$ factor in complexities [CLO24]

2. Complexities of FPSI for $L_{\mathrm{p}\in[1,\infty]}$ distance have superlinear factors on input set sizes or dimension d
   - Spatial Hashing and Locality Sensitive Hashing result in the 2^d and $m \cdot n^\rho$ factors in complexities, respectively [GRS22; GRS23; BP24]

Can we construct FPSI whose cost scales linearly with input set sizes and dimension?

Our Contributions

We focus on FPSI for Hamming and $L_{\mathrm{p}\in[1,\infty]}$ distances in semi-honest setting.

- Introduce a new primitive called Fuzzy Mapping (Fmap)
- Propose a new FPSI framework based on Fmap and Fuzzy Matching (FMatch)
- Construct FPSI for Hamming and $L_{\mathrm{p}\in[1,\infty]}$ distances with new Fmap instances
  - Costs of FPSI for Hamming distance scale linearly with input set sizes
  - Costs of FPSI for $L_{\mathrm{p}\in[1,\infty]}$ distance scale linearly with input set sizes, dimension and threashold δ
- Demonstrate the efficiency of our FPSI with an implementation

Contents

1. Background
2. Our Main Idea
3. Instantiation of Fuzzy Mapping
4. Implementation

Oblivious Key-Value Store

- Oblivious Key-Value Store (OKVS) enables encoding n key-value pairs such that an adversary can not reverse engineer the original input keys with the encoding result, when input keys $\{k_1, \cdots, k_n\}$ are distinct and values $\{v_1, \cdots, v_n\}$ are random.
- OKVS consists of Encode and Decode algorithms.
  - $D \leftarrow \text{Encode}(\{(k_1, v_1), \cdots, (k_n, v_n)\})$
  - $v \leftarrow \text{Decode}(D, k)$
  - If $k = k_i \in \{k_1, \cdots, k_n\}$, then $v = v_i$
- Recent OKVS constructions achieve output D of size $\mathcal{O}(n)$, encoding cost of $\mathcal{O}(n\lambda)$, decoding cost of $\mathcal{O}(\lambda)$, and Randomly Decoding.
  - Randomly Decoding: If $k \notin \{k_1, \cdots, k_n\}$, then $v = \text{rand}$

λ is the statistical security parameter.

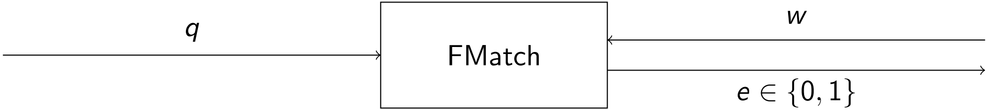
Additively Homomorphic Encryption

- An Additively Homomorphic Encryption (AHE) scheme is an encryption scheme that enables to compute an encryption of the sum of two messages by just performing operations on ciphertexts of these messages.
  - $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$
  - $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$
  - $m \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c)$
  - If $c' \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m')$ and $c'' \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m'')$, then it holds that

$$\mathsf{Dec}_{\mathsf{sk}}(c' \oplus_{\mathsf{pk}} c'') = m' + m''$$

- κ is the computational security parameter.

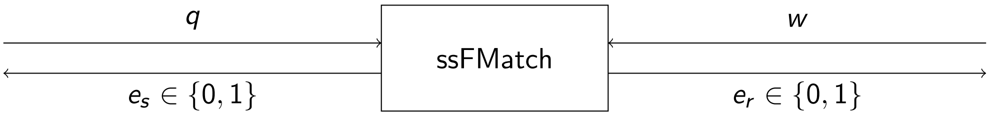
Fuzzy Matching



$$e = \begin{cases} 1, & \mathsf{dist}(w, q) \le \delta \\ 0, & \text{otherwise} \end{cases}$$

- A trivial construction of FPSI:
  - Invoke FMatch on all $m \cdot n$ pairs of inputs to indicate the result of FPSI
  - Receiver can obtain I_{fuzzy} via OT

Secret-Shared Fuzzy Matching

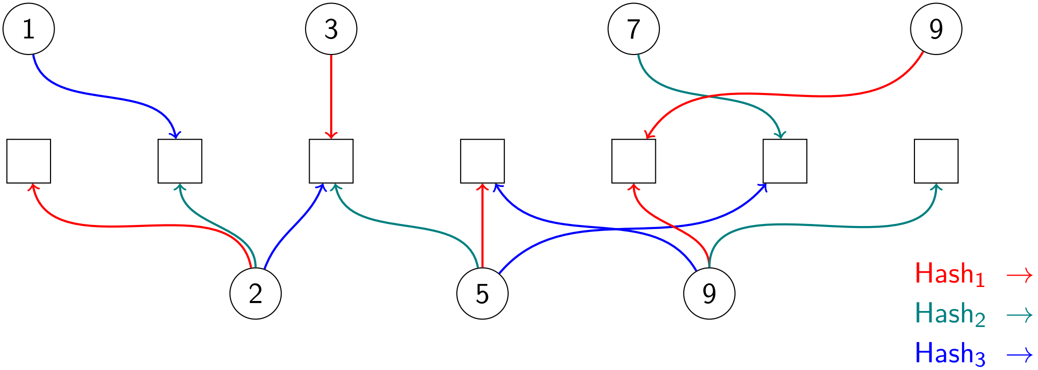


$$e_s \oplus e_r = \begin{cases} 1, & \mathrm{dist}\,(w,q) \le \delta \\ 0, & \text{otherwise} \end{cases}$$

- A trivial construction of FPSI:
  - Invoke FMatch on all $m \cdot n$ pairs of inputs to indicate the result of FPSI
  - Receiver can obtain I_{fuzzy} via OT

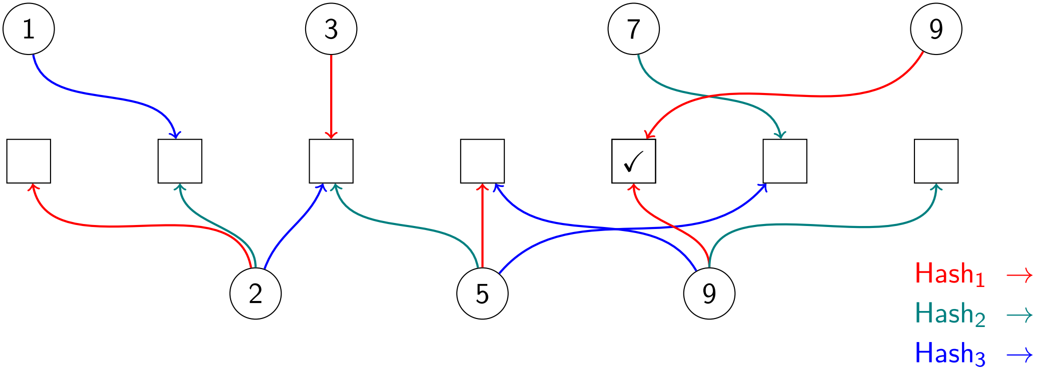
Mapping in PSI

- How does PSI avoid the $m \cdot n$ factor caused by comparing all pairs of inputs?
  - Using Cuckoo-Simple Hashing, each q_j is hashed to 1 bin and each w_i is hashed to 3 bins
  - Same elements will be hashed to a same bin
  - $Q \cap W$ can be computed by Sender and Receiver processing m and $3n$ bins, respectively



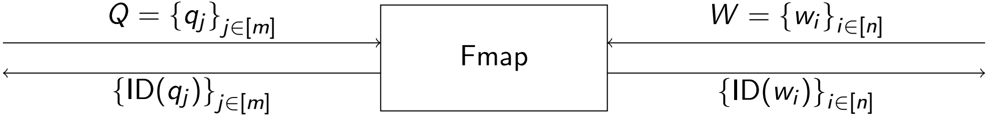
Mapping in PSI

- How does PSI avoid the $m \cdot n$ factor caused by comparing all pairs of inputs?
  - Using Cuckoo-Simple Hashing, each q_j is hashed to 1 bin and each w_i is hashed to 3 bins
  - Same elements will be hashed to a same bin
  - $Q \cap W$ can be computed by Sender and Receiver processing m and $3n$ bins, respectively



Fuzzy Mapping in FPSI

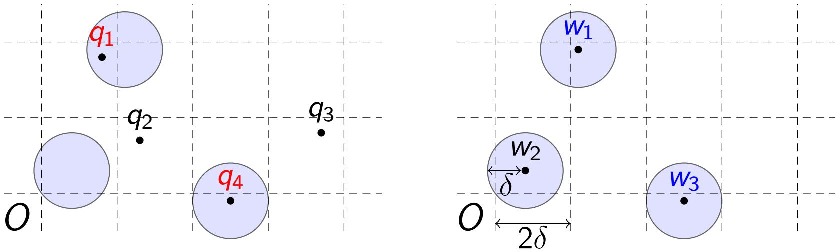
- Similarly, we define Fuzzy Mapping (Fmap) for FPSI to avoid the $m \cdot n$ factor.
  - Using Fmap, each q_j is mapped to $\text{rate}_{\mathcal{S}}$ identifiers and each w_i is mapped to $\text{rate}_{\mathcal{R}}$ identifiers
  - (Correctness) If $\text{dist}\,(w_i, q_j) \le \delta$, q_j and w_i will have a same identifier
  - I_{fuzzy} can be computed by Sender and Receiver processing $m \cdot \text{rate}_{\mathcal{S}}$ and $n \cdot \text{rate}_{\mathcal{R}}$ identifiers, respectively
  - (Security) Fmap should not leak one party's information to the other



$$|\text{ID}(q_j)| = \text{rate}_{\mathcal{S}}$$
$$|\text{ID}(w_i)| = \text{rate}_{\mathcal{R}}$$
$$\text{If dist}\,(w_i, q_j) \le \delta,\ \text{ID}(q_j) \cap \text{ID}(w_i) \ne \emptyset$$

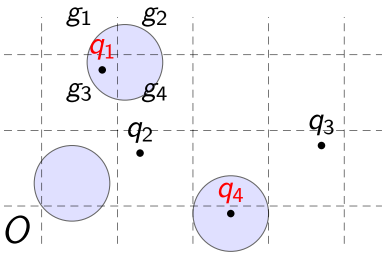
Existing Fmap Instances

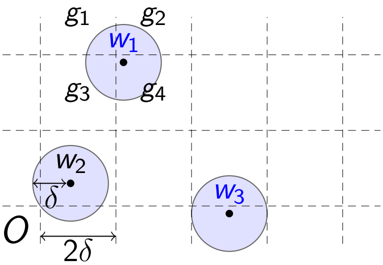
- **Naive Fmap**: Brutally traversing all pairs of inputs. [FNP04; IW06; CH08; YSPW10; UCK+21; CFR23]
  - $\mathrm{ID}(q_j) = \{1, 2, \cdots, n\}$, thus Sender have $m \cdot n$ identifiers
  - $\mathrm{ID}(w_i) = \{i\}$, thus Receiver have n identifiers
- **Spatial Hashing Fmap**: Spatial Hashing is an Fmap instance. [GRS22; GRS23; BP24]
  - The entire d-dimensional space is divided into several grids of sidelength of 2δ



Existing Fmap Instances

- **Spatial Hashing Fmap**: Spatial Hashing is an Fmap instance. [GRS22; GRS23; BP24]
  - The entire d-dimensional space is divided into several grids of sidelength 2δ
  - $\mathrm{ID}(q_j)$ is the grid including q_j, thus Sender have m identifiers
  - $\mathrm{ID}(w_i)$ are grids intersecting with ball w_i of radius δ, thus Receiver have $2^d \cdot n$ identifiers
    - ★ $\mathrm{ID}(q_1) = \{g_3\}\ldots$
    - ★ $\mathrm{ID}(w_1) = \{g_1, g_2, g_3, g_4\}\ldots$





Existing Fmap Instances

- **Naive Fmap**: Brutally traversing all pairs of inputs. [FNP04; IW06; CH08; YSPW10; UCK+21; CFR23]
  - $\mathrm{ID}(q_j) = \{1, 2, \cdots, n\}$, thus Sender have $m \cdot n$ identifiers
  - $\mathrm{ID}(w_i) = \{i\}$, thus Receiver have n identifiers
- **Spatial Hashing Fmap**: Spatial Hashing is an Fmap instance. [GRS22; GRS23; BP24]
  - $\mathrm{ID}(q_j)$ is the grid including q_j, thus Sender have m identifiers
  - $\mathrm{ID}(w_i)$ are grids intersecting with ball w_i of radius δ, thus Receiver have $2^d \cdot n$ identifiers
- **...**

Many FPSI protocols actually base on instances of Fmap. Complexity bottlenecks in these protocols are derived from the excessive expansion rates of their Fmap instances.

FPSI from Fmap

"Map and Reduce" Paradigm:

- **(Map) Map each input point to identifiers**
  Using Fmap, close points are mapped to a same identifier.
  False positives are allowed.
- **(Reduce) Reduce false positives to obtain result**
  Using OKVS, points have a same identifier form a pair.
  FMatch on these pair can reduce false positives.

FPSI from Fmap

"Map and Reduce" Paradigm:

- **(Map) Map each input point to identifiers**
  Using Fmap, close points are mapped to a same identifier.
  False positives are allowed.
- **(Reduce) Reduce false positives to obtain result**
  Using OKVS, points have a same identifier form a pair.
  FMatch on these pair can reduce false positives.

Note that Fmap for L_∞ is also the Fmap for $L_{\mathrm{p}\in[1,\infty]}$.

- For any points q and w, $L_\infty(w, q) \leq L_{\mathrm{p}\in[1,\infty]}(w, q)$
- $L_{\mathrm{p}\in[1,\infty]}(w, q) \leq \delta \Rightarrow L_\infty(w, q) \leq \delta$
- So Fmap for L_∞ can extract pairs that are close enough for $L_{\mathrm{p}\in[1,\infty]}$

FPSI from Fmap

$Q = \{q_j\}_{j\in[m]}$ → **Fmap** ← $W = \{w_i\}_{i\in[n]}$

$\{\mathrm{ID}(q_j)\}_{j\in[m]}$ ← **Fmap** → $\{\mathrm{ID}(w_i)\}_{i\in[n]}$

$$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}\left(1^\kappa\right)$$

$$\mathsf{List} \leftarrow \bigcup_{i\in[n]} \left\{ \left(\mathsf{id}_{\mathbf{w}_i}, \left(\mathsf{Enc}_{\mathsf{pk}}\left(w_i\right)\right)\right)\right\}_{\mathsf{id}_{\mathbf{w}_i} \in \mathrm{ID}(\mathbf{w}_i)}$$

$$E \leftarrow \mathsf{Encode}\left(\mathsf{List}\right)$$

$$\xleftarrow{\quad E, \mathsf{pk} \quad}$$

---

for each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$:

$$\text{samples } \mathsf{msk}_{j,\ell} = (\mathsf{msk}_{j,\ell,k})_{k\in[d]} \xleftarrow{R} \mathbb{U}^d$$

$$\mathsf{c}_{j,\ell} \leftarrow \left(\mathsf{Enc}_{\mathsf{pk}}\left(\mathsf{msk}_{j,\ell}\right) \oplus_{\mathsf{pk}} \mathsf{Decode}\left(E, \mathsf{id}_{q_j,\ell}\right)\right)$$

$$\color{red}{\mathsf{u}_{j,\ell}} \leftarrow \mathsf{msk}_{j,\ell} + q_j$$

$$\xrightarrow{\quad \mathsf{c}_{j,\ell} \quad}$$

$$\color{red}{\mathsf{v}_{j,\ell}} \leftarrow \mathsf{Dec}_{\mathsf{sk}}\left(c_{j,\ell}\right)$$

FPSI from Fmap

$$Q = \{q_j\}_{j\in[m]} \longrightarrow \boxed{\text{Fmap}} \longleftarrow W = \{w_i\}_{i\in[n]}$$

$$\{\mathsf{ID}(q_j)\}_{j\in[m]} \longleftarrow \quad \longrightarrow \{\mathsf{ID}(w_i)\}_{i\in[n]}$$

keys of OKVS encoding should be distinct
(Distinctiveness) $\mathsf{ID}(w_i) \cap \mathsf{ID}(w_j) = \emptyset$ for $i \neq j$

$$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\kappa)$$

$$\mathsf{List} \leftarrow \bigcup_{i\in[n]} \left\{ \left(\mathsf{id}_{\mathbf{w}_i}, (\mathsf{Enc}_{\mathsf{pk}}(w_i)) \right) \right\}_{\mathsf{id}_{\mathbf{w}_i} \in \mathsf{ID}(\mathbf{w}_i)}$$

$$E \leftarrow \mathsf{Encode}(\mathsf{List})$$

$$\xleftarrow{E,\ \mathsf{pk}}$$

---

for each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$:

samples $\mathsf{msk}_{j,\ell} = (\mathsf{msk}_{j,\ell,k})_{k\in[d]} \xleftarrow{\mathsf{R}} \mathbb{U}^d$

$$\mathsf{c}_{j,\ell} \leftarrow \left(\mathsf{Enc}_{\mathsf{pk}}(\mathsf{msk}_{j,\ell}) \oplus_{\mathsf{pk}} \mathsf{Decode}(E, \mathsf{id}_{q_j,\ell})\right)$$

$$\mathsf{u}_{j,\ell} \leftarrow \mathsf{msk}_{j,\ell} + q_j$$

$$\xrightarrow{\mathsf{c}_{j,\ell}}$$

$$\mathsf{v}_{j,\ell} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_{j,\ell})$$

FPSI from Fmap

for each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$:

$$\mathsf{u}_{j,\ell} \rightarrow \boxed{\text{ssFMatch}} \leftarrow \mathsf{v}_{j,\ell}$$

$$e_{s,j,\ell} \leftarrow \boxed{\text{ssFMatch}} \rightarrow e_{r,j,\ell}$$

for each $j \in [m]$:

$$(e_{s,j,1}, e_{s,j,2}, \cdots, e_{s,j,\mathsf{rate}_{\mathcal{S}}}) \rightarrow \boxed{\text{PEqT}} \leftarrow (e_{r,j,1}, e_{r,j,2}, \cdots, e_{r,j,\mathsf{rate}_{\mathcal{S}}})$$

$$\boxed{\text{PEqT}} \rightarrow e_j$$

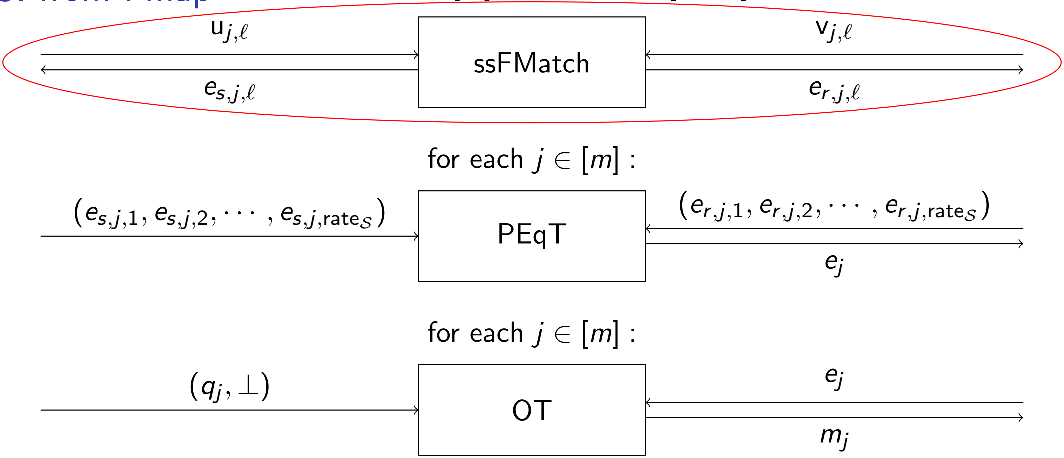
for each $j \in [m]$:

$$(q_j, \perp) \rightarrow \boxed{\text{OT}} \leftarrow e_j$$

$$\boxed{\text{OT}} \rightarrow m_j$$

FPSI from Fmap

for each $j \in [m]$ and each $\ell \in [\text{rate}_{\mathcal{S}}]$:

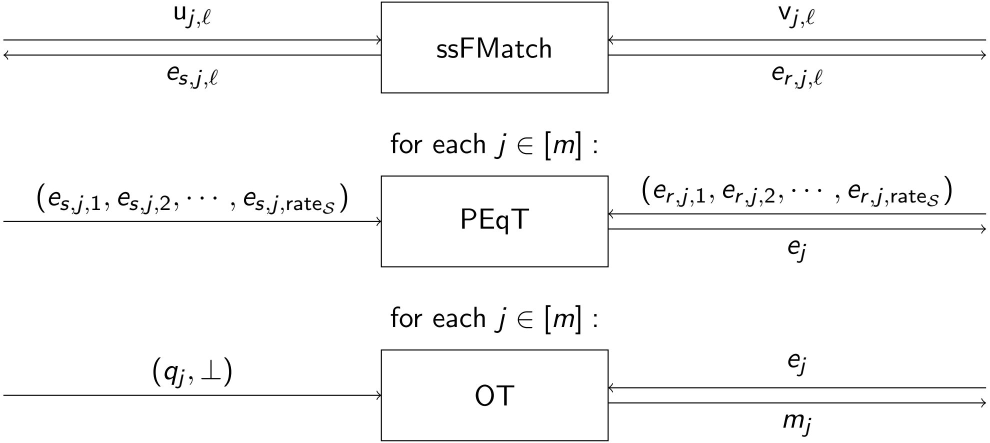


translation invariance:

$\text{dist}(q_j + \text{msk}_j, w_i + \text{msk}_j) = \text{dist}(q_j, w_i)$

FPSI from Fmap

for each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$:



If $\mathsf{rate}_{\mathcal{S}}$ and $\mathsf{rate}_{\mathcal{R}}$ are not related to m and n,
FPSI's cost scales linearly with input set sizes

Contents

1. Background
2. Our Main Idea
3. Instantiation of Fuzzy Mapping
4. Implementation

Instance of Fmap for Hamming Distance

- To construct an efficient FPSI, all we need is an Fmap with small $\mathsf{rate}_{\mathcal{S}}$ and $\mathsf{rate}_{\mathcal{R}}$.

Instance of Fmap for Hamming Distance

- To construct an efficient FPSI, all we need is an Fmap with small $\text{rate}_{\mathcal{S}}$ and $\text{rate}_{\mathcal{R}}$.
- For Hamming distance, we assume that each Receiver's point has $\delta + 1$ unique components (R. UniqC).
- In other words, for each Receiver's point w_i, there exists at least $\delta + 1$ dimensions such that on each of them w_i's component is different from $w_{i' \neq i}$'s components.

Instance of Fmap for Hamming Distance

- To construct an efficient FPSI, all we need is an Fmap with small $\text{rate}_{\mathcal{S}}$ and $\text{rate}_{\mathcal{R}}$.
- For Hamming distance, we assume that each Receiver's point has $\delta + 1$ unique components (R. UniqC).
- **UniqC Fmap** for Hamming distance.
  - UniqC Fmap maps q_j to all of its d components, thus $\text{rate}_{\mathcal{S}} = d$
  - UniqC Fmap maps w_i to $\delta + 1$ unique components, thus $\text{rate}_{\mathcal{R}} = \delta + 1$
- (Correctness) If $\text{Ham}(q_j, w_i) \leq \delta$, q_j and w_i have at most δ different components. Therefore, $\text{ID}(q_j) \cap \text{ID}(w_i) \neq \emptyset$.
- (Security) Security property is self-evident because UniqC Fmap has no interaction.
- (Distinctiveness) R. UniqC assumption guarantees that different Receiver's points have different identifiers.

Instance of Fmap for Hamming Distance

- To construct an efficient FPSI, all we need is an Fmap with small $\text{rate}_{\mathcal{S}}$ and $\text{rate}_{\mathcal{R}}$.
- For Hamming distance, we assume that each Receiver's point has $\delta + 1$ unique components (R. UniqC).
- **UniqC Fmap** for Hamming distance.
  - UniqC Fmap maps q_j to all of its d components, thus $\text{rate}_{\mathcal{S}} = d$
  - UniqC Fmap maps w_i to $\delta + 1$ unique components, thus $\text{rate}_{\mathcal{R}} = \delta + 1$
- (Correctness) If $\text{Ham}(q_j, w_i) \leq \delta$, q_j and w_i have at most δ different components. Therefore, $\text{ID}(q_j) \cap \text{ID}(w_i) \neq \emptyset$.
- (Security) Security property is self-evident because UniqC Fmap has no interaction.
- (Distinctiveness) R. UniqC assumption guarantees that different Receiver's points have different identifiers.
- Obviously, in a high dimensional space, R. UniqC assumption holds with high probability for a uniformly distributed set of points.

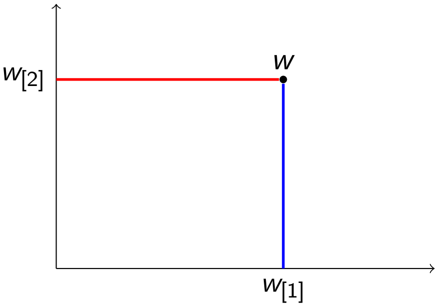
Instance of Fmap for L_∞ Distance

- As mentioned before, Fmap for L_∞ is also the Fmap for $L_{\mathrm{p}\in[1,\infty]}$
- Thus, to construct FPSI for $L_{\mathrm{p}\in[1,\infty]}$ distance, we only need an Fmap for L_∞ distance
- In fact, we report an instance of Fmap for L_∞ distance with $\text{rate}_{\mathcal{S}} = \text{rate}_{\mathcal{R}} = 1$
- Our Fmap with optimal expansion rate brings great efficiency to our FPSI

Instance of Fmap for L_∞ Distance

- We assign random values to points on each of Receiver's d axes.
- The assignment of point w in Receiver's coordinate system $\mathrm{Seed}_{r,w}$ is the sum of its d components' assignment in this coordinate system.

$$\mathrm{Seed}_{r,w} = \sum_{k \in [d]} \mathrm{r}_{r,w_{[k]}}$$

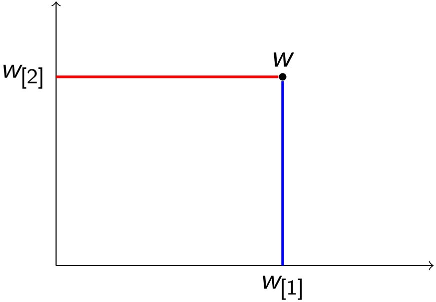


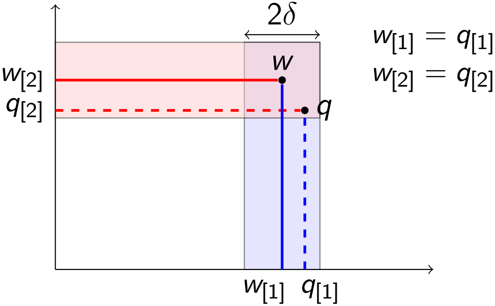
Instance of Fmap for L_∞ Distance

- If the assignment of Receiver's d axes satisfies:

$$\forall k \in [d], \forall \ell \in [-\delta, -\delta], \mathsf{r}_{r,w_{[k]}+\ell} = \mathsf{r}_{r,w_{[k]}}$$

- Then, if $L_\infty(w, q) \le \delta$, we have $\mathsf{Seed}_{r,w} = \mathsf{Seed}_{r,q}$
- Symmetrically, $\mathsf{Seed}_{s,w} = \mathsf{Seed}_{s,q}$ holds





Instance of Fmap for L_∞ Distance

- Using AHE and OKVS, it is easy to inform Sender assignments of its points in Receiver's coordinate system

$$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}\left(1^\kappa\right)$$

$$\mathsf{List} \leftarrow \bigcup_{i\in[n],k\in[d]} \left\{ \left(k \| (w_{i[k]} + \ell), \mathsf{Enc}_{\mathsf{pk}}\left(\mathsf{r}_{r,w_{i[k]}}\right) \right) \right\}_{\ell\in[-\delta,\delta]}$$

$$E \leftarrow \mathsf{Encode}\left(\mathsf{List}\right)$$

$$\xleftarrow{\quad E, \mathsf{pk} \quad}$$

for each $j \in [m]$:

samples $\mathsf{msk}_j \xleftarrow{\mathsf{R}} \mathcal{P}$

$$\mathsf{c}_j \leftarrow \mathsf{Enc}_{\mathsf{pk}}\left(\mathsf{msk}_j\right) \oplus_{\mathsf{pk}} \left(\bigoplus_{\mathsf{pk}\, k\in[d]} \mathsf{Decode}\left(E, k \| q_{j[k]}\right) \right)$$

$$\xrightarrow{\quad \mathsf{c}_j \quad}$$

$$p_j \leftarrow \mathsf{Dec}_{\mathsf{sk}}\left(c_j\right)$$

$$\xleftarrow{\quad p_j \quad}$$

$$\mathsf{Seed}_{r,q_j} \leftarrow p_j - \mathsf{msk}_j$$

Instance of Fmap for L_∞ Distance

- Using AHE and OKVS, it is easy to inform Sender assignments of its points in Receiver's coordinate system
- But we should not use Seed_{r,q_j} as q_j's identifier
  - For Sender's points q_j and q_j', if $\mathsf{Seed}_{r,q_j} = \mathsf{Seed}_{r,q_j'}$, Sender can infer that there is a Receiver's point nearby. Such information leakage undermines security
- We choose to avoid this with symmetric operations and DH-like subprotocol in our Fmap
- The identifier of a point is DDH value of the sum of its assignments in Sender's and Receiver's coordinate system
  - For Sender, identifier of q_j is $\mathsf{id}_{q_j} = (\mathsf{Seed}_{r,q_j} + \mathsf{Seed}_{s,q_j})^{\mathsf{sk}_{\mathsf{DH},\mathcal{R}} \cdot \mathsf{sk}_{\mathsf{DH},\mathcal{S}}}$
  - For Sender, identifier of w_i is $\mathsf{id}_{w_i} = (\mathsf{Seed}_{r,w_i} + \mathsf{Seed}_{s,w_i})^{\mathsf{sk}_{\mathsf{DH},\mathcal{R}} \cdot \mathsf{sk}_{\mathsf{DH},\mathcal{S}}}$
  - Here, $\mathsf{sk}_{\mathsf{DH},\mathcal{S}}$ and $\mathsf{sk}_{\mathsf{DH},\mathcal{R}}$ are Sender's and Receiver's private keys, respectively
- (Correctness) If $L_\infty(w_i, q_j) \le \delta$, we have $\mathsf{Seed}_{r,w_i} = \mathsf{Seed}_{r,q_j}$ and $\mathsf{Seed}_{s,w_i} = \mathsf{Seed}_{s,q_j}$. So
$$\mathsf{id}_{q_j} = \mathsf{id}_{w_i}$$
- (Distinctiveness) We assume that Seeds of different points are different

Contents

1 Background

2 Our Main Idea

3 Instantiation of Fuzzy Mapping

4 Implementation

Experiment Results for Hamming Distance

- Experiments are conducted in LAN setting, and we omit all the offline costs

Table: The comparison of SOTA and our FPSI protocol for Hamming distance in running time (s) and communication cost (MB), where dimension $d = 128$, and threshold $\delta = 4$.

Set Size $m = n$	Protocol	Cost	
		Comm.	Comp.
256	[CLO24]	465.68	38.7
	Ours	**91.889**	**5.18**
1024	[CLO24]	1779.3	147.85
	Ours	**367.53**	**19.428**
4096	[CLO24]	6870	569.9
	Ours	**1470**	**76.00**

Experiment Results for $L_{p\in[1,\infty]}$ Distance

Table: The comparison of SOTA and our FPSI protocol for L_2 distance in running time (s) and communication cost (MB).

$m=n$	Protocol	(d,δ) (2,30) Comm.	Comp.	(6,30) Comm.	Comp.	(10,30) Comm.	Comp.
2^4	[BP24]	**0.957**	3.082	25.19	74.80	660.4	2046
	Ours	1.339	**0.820**	**3.960**	**1.783**	**6.581**	**2.801**
2^8	[BP24]	**15.31**	45.34	403.1	1246	$>10^4$	$>10^4$
	Ours	21.42	**8.825**	**63.35**	**23.18**	**106.6**	**38.97**
2^{12}	[BP24]	**244.9**	742.6	>6000	$>10^4$	$>10^5$	$>10^5$
	Ours	346.8	**142.3**	**1026**	**402.7**	**1706**	**657.2**
2^{16}	[BP24]	**3919**	12017	$>10^4$	$>10^5$	$>10^6$	$>10^6$
	Ours	5549	**2366**	**16419**	**6539**	**27289**	**10953**

Table: The comparison of SOTA and our FPSI protocol for L_∞ distance in running time (s) and communication cost (MB).

$m=n$	Protocol	(d,δ) (2,30) Comm.	Comp.	(6,30) Comm.	Comp.	(10,30) Comm.	Comp.
2^4	[BP24]	**0.517**	1.891	24.75	73.61	660.0	2042
	Ours	1.340	**0.696**	**3.994**	**1.727**	**6.648**	**2.501**
2^8	[BP24]	**8.266**	25.10	396.0	1225	$>10^4$	$>10^4$
	Ours	21.44	**7.930**	**63.90**	**22.28**	**106.4**	**36.99**
2^{12}	[BP24]	**132.3**	420.8	>6000	$>10^4$	$>10^5$	$>10^5$
	Ours	343.0	**128.9**	**1022**	**391.4**	**1702**	**644.1**
2^{16}	[BP24]	**2116**	6796	$>10^4$	$>10^5$	$>10^6$	$>10^6$
	Ours	5488	**2218**	**16358**	**6366**	**27228**	**10779**

- Our protocol performs better in almost every situation
- The larger the set sizes and dimension, the greater our advantage

References

[BP24] Aron van Baarsen and Sihang Pu. "Fuzzy Private Set Intersection with Large Hyperballs". In: *Advances in Cryptology – EUROCRYPT 2024*. Ed. by Marc Joye and Gregor Leander. Cham: Springer Nature Switzerland, 2024, pp. 340–369.

[CFR23] Anrin Chakraborti, Giulia Fanti, and Michael K. Reiter. "Distance-Aware Private Set Intersection". In: *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, 2023, pp. 319–336.

[CH08] Lukasz Chmielewski and Jaap-Henk Hoepman. "Fuzzy Private Matching (Extended Abstract)". In: *2008 Third International Conference on Availability, Reliability and Security*. 2008, pp. 327–334.

[CLO24] Wutichai Chongchitmate, Steve Lu, and Rafail Ostrovsky. *Approximate PSI with Near-Linear Communication*. Cryptology ePrint Archive, Paper 2024/682. 2024. URL: `https://eprint.iacr.org/2024/682`.

[FNP04] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. "Efficient Private Matching and Set Intersection". In: *Advances in Cryptology - EUROCRYPT 2004*. Ed. by Christian Cachin and Jan L. Camenisch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–19.

[GRS22] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. "Structure-Aware Private Set Intersection, with Applications to Fuzzy Matching". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 323–352.

[GRS23] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. "Malicious Secure, Structure-Aware Private Set Intersection". In: *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 577–610.

[IW06] Piotr Indyk and David Woodruff. "Polylogarithmic Private Approximations and Efficient Matching". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 245–264.

[UCK+21] Erkam Uzun et al. "Fuzzy Labeled Private Set Intersection with Applications to Private Real-Time Biometric Search". In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2021, pp. 911–928.

[YSPW10] Qingsong Ye, Ron Steinfeld, Josef Pieprzyk, and Huaxiong Wang. "Efficient Fuzzy Matching and Intersection on Private Datasets". In: *Information, Security and Cryptology – ICISC 2009*. Ed. by Donghoon Lee and Seokhie Hong. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 211–228.

Thanks for your attention!