

# Efficient Fuzzy Private Set Intersection from Fuzzy Mapping

Ying Gao<sup>1,2</sup>(✉)<sup>[0000-0001-8992-651X]</sup>, Lin Qi<sup>1</sup><sup>[0009-0003-0489-9412]</sup>,  
Xiang Liu<sup>1</sup><sup>[0009-0000-8128-7872]</sup>, Yuanchao Luo<sup>1</sup><sup>[0009-0001-6541-690X]</sup>, and  
Longxin Wang<sup>1</sup><sup>[0009-0007-2277-7039]</sup>

<sup>1</sup> School of Cyber Science and Technology, Beihang University, Beijing, China  
{gaoying, 19373287, lx1234, 19373507, wx\_buaa}@buaa.edu.cn

<sup>2</sup> Zhongguancun Laboratory, Beijing, China

**Abstract.** Private set intersection (PSI) allows Sender holding a set  $X$  and Receiver holding a set  $Y$  to compute only the intersection  $X \cap Y$  for Receiver. We focus on a variant of PSI, called fuzzy PSI (FPSI), where Receiver only gets points in  $X$  that are at the distance not greater than a threshold from some points in  $Y$ .

Most current FPSI approaches first pick out pairs of points that are potentially close and then determine whether the distance of each selected pair is indeed small enough to yield FPSI result. Their complexity bottlenecks stem from the excessive number of point pairs selected by the first picking process. Regarding this process, we consider a more general notion, called fuzzy mapping (Fmap), which can map each point of two parties to a set of identifiers, with closely located points having a same identifier, which forms the selected point pairs.

We initiate the formal study on Fmap and show novel Fmap instances for Hamming and  $L_\infty$  distances to reduce the number of selected pairs. We demonstrate the powerful capability of Fmap with some superior properties in constructing FPSI variants and provide a generic construction from Fmap to FPSI.

Our new Fmap instances lead to the fastest semi-honest secure FPSI protocols in high-dimensional space to date, for both Hamming and general  $L_{p \in [1, \infty]}$  distances. For Hamming distance, our protocol is the first one that achieves strictly linear complexity with input sizes. For  $L_{p \in [1, \infty]}$  distance, our protocol is the first one that achieves linear complexity with input sizes, dimension, and threshold.

**Keywords:** Fuzzy private set intersection · Fuzzy mapping · Multi-query fuzzy reverse private membership test

## 1 Introduction

Private set intersection (PSI) enables two parties, each with a private set, to compute the intersection of their sets without revealing any information more than the intersection itself. Since its high practical value in threat detection, private contact discovery, sample alignment, and other scenarios, numerous PSI

protocols [24, 32, 33] have been designed in last decades. And recent PSI protocols have achieved extremely high efficiency [32]. Facing various complex practical needs, there is also a growing interest in works on variants of PSI, including labeled PSI (LPSI) [4, 8, 12], which outputs labels associated with elements in intersection to Receiver; PSI cardinality (PSI-card) [13, 23, 36], which only reveals intersection cardinality to Receiver; threshold PSI [2, 5, 19, 20, 25], which outputs the intersection if and only if its size is within a given range.

This work focuses on a variant of PSI, fuzzy PSI (FPSI). The input of FPSI consists of  $m$  points from Sender in a  $d$ -dimensional space and  $n$  points from Receiver in the same space. And FPSI’s output only informs Receiver those Sender’s points that have the distance (e.g., Hamming distance,  $L_2$  distance, etc.) with some Receiver’s points not greater than the threshold  $\delta$ , while nothing is revealed to Sender. FPSI has many potential applications in fields that involve fuzzy matching on datasets, such as privacy-preserving biometric search [34], illegal content detection [1], and vulnerable password detection [7]. For instance, the deployment of biometric systems in public places for searching for sensitive groups (such as fugitives) yields significant benefits to public safety. However, public concerns over privacy protection make it impractical to upload locally recognized biometric features to a cloud database for matching. Using PSI can solve the privacy issue, but since biometric features always contain some noises (e.g., due to environmental disturbance, algorithms’ randomness, etc.), conventional PSI cannot fulfill feature matching. In such cases, FPSI becomes indispensable.

Since the concept of fuzzy matching was introduced in PSI by Freedman, Nissim, and Pinkas [15] in 2004, there has been a long list of works related to FPSI [1, 7, 10, 11, 17, 18, 22, 34, 37]. The majority of them concern with FPSI for Hamming distance, and the few exceptions [17, 18, 22] only consider about one or two of  $L_1$ ,  $L_2$ , and  $L_\infty$  distances. Until 2024, Baarsen and Pu [1] make a breakthrough by presenting the first FPSI protocol supporting general  $L_p$  distance with  $p \in [1, \infty]$ . As the state-of-the-art FPSI for  $L_{p \in [1, \infty]}$  distance, the communication and computation costs of their protocols for high dimension scale linearly or quadratically with the dimension  $d$ . They also conduct research on some variants of FPSI, including labeled FPSI (LFPSI), fuzzy PSI-card (FPSI-card), and FPSI with Sender privacy (FPSI-SP). Regrettably, due to the super-linear factor in complexities, their protocols still have room for improvement.

## 1.1 Motivation

Current FPSI protocols might have expensive overheads for their super-linear factors in complexities.

First, all existing FPSI protocols for Hamming distance retain super-linear factors with input sizes in their complexities. Starting from [15], most FPSI protocols for Hamming distance employ the same idea: perform fuzzy matching over all  $m \cdot n$  pairs of inputs in order to select the final result. Existing works often focus on improving fuzzy matching protocol for Hamming distance, and rarely deal with  $m \cdot n$  factor introduced by this idea. The current best FPSI protocol for Hamming distance reducing this quadratic factor at the cost of

introducing assumption on inputs from both parties, only achieves near-linear complexity [11].

Second, the efficiency of existing FPSI protocols for  $L_{p \in [1, \infty]}$  distance is also not satisfactory. Using oblivious key-value store (OKVS) and decisional Diffie-Hellman (DDH) tuple, Baarsen and Pu [1] provide FPSI protocols with previous optimal complexities. However, most of their protocols are still troubled by super-linear complexity with dimension. Although their protocol based on locality-sensitive hashing (LSH) has communication and computation costs scaling linearly with dimension, its costs scale super-linearly with Receiver’s input size. Unfortunately, the prevalence of real databases with substantial dimension and size (such as facial feature databases) makes the previously mentioned flaws greatly hindering the applications of their protocols.

So, there exist two fascinating open questions:

- *Can we construct an FPSI protocol for Hamming distance with communication and computation complexities that are strictly linear with  $m$  and  $n$ <sup>3</sup>?*
- *Can we construct an FPSI protocol for  $L_{p \in [1, \infty]}$  distance of which costs scale linearly with anyone of  $m$ ,  $n$ ,  $d$ , and  $\delta$ ?*

## 1.2 Our Contribution

We provide affirmative answers to these two questions in the semi-honest setting. Our main contributions are summarized as below.

- **A New Cryptographic Primitive Called Fuzzy Mapping.** We introduce the abstraction of a new cryptographic primitive called fuzzy mapping (Fmap). We show that many FPSI protocols [1, 7, 10, 15, 17, 22, 34, 37] actually are based on instances of Fmap, and complexity bottlenecks in these protocols derive from the excessive expansion rates of their Fmap instances. Under some reasonable assumptions about inputs, we present a non-trivial Fmap instance for Hamming distance and an Fmap instance for  $L_\infty$  distance with expansion rate of 1.
- **FPSI for Hamming Distance of Which Costs Scale Strictly Linearly with  $m$  and  $n$ .** We provide a generic construction for FPSI from Fmap that does not introduce any additional assumptions about inputs. As an instance of it, we construct an FPSI protocol for Hamming distance using our Fmap instance for Hamming distance. Due to the employment of this non-trivial Fmap instance, communication and computation complexity of the new protocol achieve strict linearity with  $m$  and  $n$  for the first time.
- **FPSI for  $L_{p \in [1, \infty]}$  Distance of Which Costs Scale Linearly with Anyone of  $m$ ,  $n$ ,  $d$ , and  $\delta$ .** We show how to construct multi-query fuzzy reverse private membership test (mqFRPMT), the fuzzy version of multi-query reverse private membership test (mqRPMT), from Fmap without expansion on Sender’s set. Using mqFRPMT, we can easily obtain FPSI and its

<sup>3</sup> That is to say, as both  $m$  and  $n$  grow to be  $k$  times larger, communication and computation costs of the protocol increase to  $k$  times at most.

variants, including FPSI-card, LFPSI, and FPSI-SP. By instantiating with our Fmap instance for  $L_\infty$  distance with expansion rate of 1, we ultimately construct a new FPSI protocol for  $L_{p \in [1, \infty]}$  distance. Its costs scale linearly with any one of  $m$ ,  $n$ ,  $d$ , and  $\delta$ , which allows it to perform better than prior protocols.

- **Performance.** Our experimental results demonstrate that compared with the state-of-the-art protocols, our protocol achieves a  $4.6\times$  reduction in communication cost for Hamming distance when both parties input 128-bit binary strings and  $\delta$  is set to 4, and achieves a  $28 - 166\times$  speedup and  $6 - 40\times$  reduction in communication cost for  $L_{p \in \{1, 2, \infty\}}$  distance when  $d \geq 6$ .

### 1.3 Related Work

We review previous semi-honest secure FPSI protocols, which can be divided into two categories: FPSI for Hamming distance and FPSI for  $L_{p \in [1, \infty]}$  distance. A comparison of asymptotic complexities is given in Table 1.

**FPSI for Hamming Distance.** Freedman et al. [15] first propose the concept of FPSI and provide a protocol for Hamming distance based on polynomial interpolation and additively homomorphic encryption (AHE). Their protocol has been proved insecure by Chmielewski and Hoepman [10]. For a long time, subsequent works on FPSI mainly focus on Hamming distance. Ye et al. [37] design FPSI for Hamming distance with oblivious polynomial evaluation technique. Indyk and Woodruff [22] deal with FPSI for Hamming and  $L_2$  distances, but their protocols rely on AHE and costly garbled circuits. Uzun et al. [34] construct LFPSI for Hamming distance based on fully homomorphic encryption (FHE), another costly technique. Using vector oblivious linear evaluation, Chakraborti et al. [7] propose an efficient FPSI for Hamming distance of which cost is independent of  $d$ , but at the cost of a non-negligible false positive rate. In addition, they propose an efficient FPSI for  $L_1$  distance in one-dimensional space with the concept of prefix matching. These protocols always perform a brute-force search over all  $m \cdot n$  pairs of inputs from both parties, which results in an  $m \cdot n$  blowup in communication and computation complexities. In 2024, Chongchitmate et al. [11] propose the most efficient FPSI for Hamming distance to date. Their protocol reduces the  $m \cdot n$  blowup through approximating FPSI result via multiple rounds of PSI on sampled components of points. However, its complexities still fail to achieve strict linearity with  $m$  and  $n$ . Moreover, same with previous protocols in [7, 34], they only consider Hamming distance over  $\mathbb{F}_2$ .

**FPSI for  $L_{p \in [1, \infty]}$  Distance.** In 2022, Garimella et al. [17] construct the first FPSI protocols for  $L_1$  and  $L_\infty$  distance, which are considered as instances of structure-aware PSI in their opinion. For FPSI, their key innovation lies in the use of spatial hashing technique to decrease communication complexity. However, they do not discuss FPSI for general  $L_p$  distance and lack the improvement

**Table 1.** Asymptotic complexities of semi-honest secure FPSI protocols, where Sender holds  $m$  points and Receiver holds  $n$  points in a  $d$ -dimensional space.  $M$  is the larger one of  $m$  and  $n$ .  $\delta$  is the threshold of FPSI.  $B_1$  and  $B_2$  are parameters in FHE scheme.  $\rho \in (0, 1)$  is a parameter in LSH scheme. We ignore multiplicative factors of the computational security parameter  $\kappa$  and statistical security parameter  $\lambda$ . In computation cost, we ignore non-cryptographic operations, such as the binary search and polynomial interpolation in plaintext form.

Distance	Protocol	Assumption	Communication	Computation	
				Sender	Receiver
Hamming	[37]	–	$\mathcal{O}(d^2 mn)$	$\mathcal{O}(\text{poly}(d)mn)$	$\mathcal{O}(d^2 mn)$
	[34] <sup>⊖</sup>	FPR&FNR	$\mathcal{O}(B_1 dmn)$	$\mathcal{O}(B_2 dmn)$	$\mathcal{O}\left(\binom{d}{\delta} n\right)$
	[7] <sup>⊖</sup>	FNR	$\mathcal{O}(\delta^2 mn)$	$\mathcal{O}((d + \delta^2)mn)$	$\mathcal{O}((d + \delta)mn)$
	[11] <sup>⊖</sup>	R $\wedge$ S. cluster.	$\mathcal{O}(dM \log M)$	$\mathcal{O}(dM \log M)$	$\mathcal{O}(dM \log M)$
	Ours	R. UniqC	$\mathcal{O}(d^2 m + \delta dn)$	$\mathcal{O}(d^2 m)$	$\mathcal{O}(d^2 m + \delta dn)$
$L_\infty$	[17]	R. $l_{\min} > 2\delta$	$\mathcal{O}(m + (4 \log \delta)^d n)$	$\mathcal{O}((2 \log \delta)^d m)$	$\mathcal{O}((2\delta)^d n)$
	[1]	R. $l_{\min} > 2\delta$	$\mathcal{O}(2^d m + \delta dn)$	$\mathcal{O}(2^d dm)$	$\mathcal{O}(2^d m + \delta dn)$
		R. disj. proj.	$\mathcal{O}(m + (\delta d)^2 n)$	$\mathcal{O}(d^2 m)$	$\mathcal{O}(m + (\delta d)^2 n)$
	Ours	R $\wedge$ S. disj. proj.	$\mathcal{O}(\delta dm + \delta dn)$	$\mathcal{O}(\delta dm + n)$	$\mathcal{O}(m + \delta dn)$
$L_{\rho \in [1, \infty]}$	[1]	R. $l_{\min} > 2\delta \left(\frac{d^\rho}{\rho} + 1\right)$	$\mathcal{O}(\delta^\rho m + \delta 2^d dn)$	$\mathcal{O}((d + \delta^\rho) m)$	$\mathcal{O}(m + \delta 2^d dn)$
		R. $l_{\min} > \frac{1}{\rho} \delta$	$\mathcal{O}((\delta^\rho n^\rho \log n) m + \delta dn^{\rho+1})$	$\mathcal{O}(((d + \delta^\rho) n^\rho \log n) m)$	$\mathcal{O}((n^\rho \log n) m + \delta dn^{\rho+1})$
	Ours	R $\wedge$ S. disj. proj.	$\mathcal{O}((\delta d + \rho \log \delta) m + \delta dn)$	$\mathcal{O}((\delta d + \rho \log \delta) m + n)$	$\mathcal{O}(\rho \log \delta m + \delta dn)$

-  $\ominus$  means that this protocol only handles with Hamming distance on bit vectors.

- FPR (FNR) means that Receiver can tolerate a non-negligible false positive rate (false negative rate).

- R  $\wedge$  S. cluster. means that for both Sender's set and Receiver's set, the Hamming distance between any two points in the same set should be less than  $\delta$  or greater than  $\delta \log n$ .

- R. UniqC means that for each Receiver's point, there exists at least  $\delta + 1$  dimensions such that on each of them this point's component is different from others.

- R.  $l_{\min} > l_*$  means that the minimum distance between points of Receiver is greater than  $l_*$ .

- R. disj. proj. means that for each Receiver's point, there exists at least one dimension on which its component keeps a distance greater than  $2\delta$  from other Receiver's points.

- R  $\wedge$  S. disj. proj. means that the disj. proj. assumption should hold for both Sender's set and Receiver's set.

in computation cost. In 2024, Baarsen and Pu [1] propose the first FPSI protocol supporting general  $L_{\rho \in [1, \infty]}$  distance. They use spatial hashing or similar techniques for coarse filtration on all pairs of both inputs, and propose a novel fuzzy matching protocol based on OKVS and DDH tuple for refined filtering to complete FPSI. Additionally, they go further in protecting Sender privacy by proposing and constructing FPSI-SP. Although many techniques are employed to optimize complexity, complexities of their protocols still remain super-linear factors in  $n$  or  $d$ , which make their efficiency suffer greatly.

*Remark 1.* Note that recent protocols are always based on assumptions. It is necessary to introduce assumptions for making costs strictly linear with  $m$  and  $n$ . The motivation is to limit the number of point pairs that might successfully match in FPSI. If no restrictions are imposed, the number of point pairs that need to be checked is  $m \cdot n$ , which inevitably leads to an  $m \cdot n$  factor in complexities [11].

## 2 Overview of Our Techniques

In this section, we present a high-level technical overview of our work. And the ideal functionalities for FPSI and its several variants considered in our work are given in Fig.1.

<p>PARAMETERS: Sender <math>\mathcal{S}</math>, Receiver <math>\mathcal{R}</math>; Set size <math>m, n</math>; Dimension <math>d</math>; Distance function <math>\text{dist}(\cdot, \cdot)</math>, Distance threshold <math>\delta</math>; Leakage function <math>\text{leakage}(\cdot, \cdot)</math>; Label length <math>\sigma</math>.</p> <p>FUNCTIONALITY:</p> <ul style="list-style-type: none"> <li>– Wait an input <math>\mathbf{Q} \in \mathbb{U}^{d \times m}</math> from <math>\mathcal{S}</math>.</li> <li>– For LFPSI, wait another input <math>\text{Label}_{\mathbf{Q}} \in \{0, 1\}^{\sigma \times m}</math> from <math>\mathcal{S}</math>.</li> <li>– Wait an input <math>\mathbf{W} \in \mathbb{U}^{d \times n}</math> from <math>\mathcal{R}</math>.</li> <li>– Return <math>\text{leakage}(\mathbf{Q}, \mathbf{W})</math> to <math>\mathcal{R}</math>.</li> </ul> <p>LEAKAGE FUNCTIONS: <math>\text{leakage}(\mathbf{Q}, \mathbf{W})</math> is defined as:</p> <ul style="list-style-type: none"> <li>– <b>FPSI</b>: <math>\text{leakage}(\mathbf{Q}, \mathbf{W}) = \{\mathbf{q}_j \mid \exists i \in [n], \text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta\}</math>.</li> <li>– <b>LFPSI</b>: <math>\text{leakage}(\mathbf{Q}, \mathbf{W}) = \{\text{label}_j \mid \exists i \in [n], \text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta\}</math>, where <math>\text{label}_j</math> is the label associated with <math>\mathbf{q}_j</math>.</li> <li>– <b>FPSI-card</b>: <math>\text{leakage}(\mathbf{Q}, \mathbf{W}) = \sum_{j \in [m], i \in [n]} (\text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta)</math>.</li> <li>– <b>FPSI-SP</b>: <math>\text{leakage}(\mathbf{Q}, \mathbf{W}) = \{\mathbf{w}_i \mid \exists j \in [m], \text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta\}</math>.</li> </ul>
--

**Fig. 1.** Ideal Functionalities for FPSI and Its Variants:  $\mathcal{F}_{\text{FPSI}}$ ,  $\mathcal{F}_{\text{LFPSI}}$ ,  $\mathcal{F}_{\text{FPSI-card}}$ , and  $\mathcal{F}_{\text{FPSI-SP}}$

### 2.1 Challenge in Efficient FPSI

Most FPSI protocols [1, 7, 10, 15, 17, 22, 34, 37], including ours, are based on the same idea: perform FPSI using a batch of fuzzy matching, which can determine whether a Sender’s point  $\mathbf{q}_j$  and a Receiver’s point  $\mathbf{w}_i$  satisfy  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ , and return the result to Receiver. Therefore, there are two directions for improving FPSI protocols: one is the optimization of fuzzy matching protocol, which is the focus of most existing works [7, 10, 15, 22, 34, 37], and the other is the optimization of the process of reducing FPSI to fuzzy matching, which is the focus of this work.

The above FPSI paradigm can be decomposed into two phases: “coarse mapping” and “refined filtering” [1]. Coarse mapping is used to assign several identifiers to points of Sender and Receiver, and two points from Sender and Receiver respectively with a same identifier will form a pair<sup>4</sup>. Refined filtering is used to perform fuzzy matching on each pair obtained from coarse mapping to get the final result.

The main complexity bottlenecks of existing works stem from their coarse mapping methods. For example, the naive coarse mapping which brutally traverses all pairs of inputs from parties results in an unacceptable  $m \cdot n$  blowup in

<sup>4</sup> A point can appear in multiple pairs.

complexities. Besides, [17] uses spatial hashing technique to perform coarse mapping. In this coarse mapping, a Receiver’s point is mapped to  $\mathcal{O}(2^d)$  identifiers. This expansion is the source of the factor  $2^d$  in complexities.

The challenge in efficient FPSI protocols is to construct coarse mapping methods with minor expansion on input sizes to break bottlenecks.

## 2.2 Fuzzy Mapping

We abstract the coarse mapping into a new cryptographic primitive named fuzzy mapping (Fmap), with the complexity bottleneck being formalized as the expansion rate of Fmap. As Section 4.2 will demonstrate, almost all known FPSI protocols are constructed based on instances of Fmap. Thus, proposing non-trivial Fmap instances is the core task in this work.

The input of Fmap consists of  $m$  points  $(\mathbf{q}_j)_{j \in [m]} \in \mathbb{U}^{d \times m}$  from Sender and  $n$  points  $(\mathbf{w}_i)_{i \in [n]} \in \mathbb{U}^{d \times n}$  from Receiver. The output of Fmap consists of  $(\text{ID}(\mathbf{q}_j))_{j \in [m]}$  for Sender and  $(\text{ID}(\mathbf{w}_i))_{i \in [n]}$  for Receiver, where  $\text{ID}(\mathbf{q})$  and  $\text{ID}(\mathbf{w})$  are subsets of an identifier universe  $\mathcal{I}$ .

**Three Requirements.** For realizing the functionality of coarse mapping securely, Fmap for  $\text{dist}(\cdot, \cdot)$  of threshold  $\delta$  should satisfy the following requirements:

- $\text{ID}(\mathbf{q}_j)$  should intersect with  $\text{ID}(\mathbf{w}_i)$  when  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i)$  is not greater than  $\delta$ . Otherwise, coarse mapping would lose the pair  $(\mathbf{q}_j, \mathbf{w}_i)$ , leading to an incorrect FPSI result. Note that the existence of refined filtering allows Fmap to tolerate false positives<sup>5</sup>.
- The probability that there exist two distinct points  $\mathbf{w}_i$  and  $\mathbf{w}_{i'}$  in  $\mathbf{W}$  such that  $\text{ID}(\mathbf{w}_i)$  intersects with  $\text{ID}(\mathbf{w}_{i'})$  is negligible. Otherwise, an identifier might lead to point pairs involving multiple Receiver’s points, which could lead to incomplete executions of fuzzy matching in refined filtering<sup>6</sup>.
- For security, Fmap should not reveal any information about one party’s input to the other party. In other words, the view of Receiver invoking Fmap with Sender should be computationally indistinguishable from that with another Sender, and the same applies to Sender’s perspective.

**Expansion Rate.** We define the Sender’s expansion rate and Receiver’s expansion rate of Fmap as the ratio of the output size to the input size for Sender and Receiver respectively.

It is clear that the optimal expansion rate of Fmap is 1. We use unit Fmap (UFmap) to denote the Fmap with both expansion rates of 1, and unit Fmap for Sender (sUFmap) to denote the Fmap with Sender’s expansion rate of 1.

<sup>5</sup> That is to say, cases that  $\text{ID}(\mathbf{q}_j)$  intersects with  $\text{ID}(\mathbf{w}_i)$  and  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i)$  is greater than  $\delta$  are allowed.

<sup>6</sup> In order to hide the distribution of points, Receiver can only initiate fuzzy matching once for each identifier. If multiple fuzzy matchings are performed on an identifier, Sender can infer that there are multiple Receiver’s points nearby.

### 2.3 Non-trivial Fmap for Hamming Distance

There is no Fmap instance for Hamming distance except the naive one (i.e., brutally traversing all  $m \cdot n$  pairs of input points), which is the primary culprit for  $m \cdot n$  blowup in complexity. Therefore, we hope to find a non-trivial Fmap to improve FPSI for Hamming distance.

We assume that each Receiver’s point has at least  $\delta + 1$  unique components, and denote this assumption as Receiver’s unique components (R. UniqC) assumption. Typically,  $\delta \ll d$  holds for applications of FPSI [26, 27]. Thus, R. UniqC assumption is intuitively reasonable, and furthermore, we formally prove that it holds with overwhelming probability for uniformly random Receiver’s input in Section 5.1. R. UniqC assumption reflects real-world scenarios where legitimate texts or numbers vary significantly, and their errors are merely deviations of a few characters, while Receiver hopes to query several entries under such circumstances [11].

Under R. UniqC assumption, a non-trivial Fmap for Hamming distance, which we refer to as UniqC Fmap, can be constructed, where Receiver’s points are mapped to their unique components, while each Sender’s point is mapped to all of its  $d$  components. More details about UniqC Fmap are shown in Section 5.1.

### 2.4 UFmap for $L_\infty$ Distance

**UFmap for  $L_\infty$  Distance is Enough.** To overcome complexity bottlenecks in FPSI protocols for  $L_{p \in [1, \infty]}$  distance, we hanker for a UFmap for  $L_{p \in [1, \infty]}$  distance. Fortunately, benefiting from the facts that Fmap can tolerate false positives and that the  $L_\infty$  distance between any two points is always no greater than the  $L_{p \in [1, \infty]}$  distance, we can use Fmap for  $L_\infty$  distance in FPSI for general  $L_{p \in [1, \infty]}$  distance. Therefore, we will only discuss the construction of UFmap for  $L_\infty$  distance in the following paragraphs.

**A Toy Protocol from Spatial Additive Sharing.** Let us first consider a toy protocol in a simplified setting where Receiver chooses  $\text{seed}_{\mathbf{w}, \mathcal{R}}$  for  $\mathbf{w} \in \mathbf{W}$  as an assignment and Sender wants to choose  $\text{seed}_{\mathbf{q}, \mathcal{R}}$  as the assignment of  $\mathbf{q} \in \mathbf{Q}$  meeting  $\text{seed}_{\mathbf{q}, \mathcal{R}}$  equals to  $\text{seed}_{\mathbf{w}, \mathcal{R}}$  when  $L_\infty(\mathbf{q}, \mathbf{w})$  is not greater than  $\delta$ .

The rough idea termed as *spatial additive sharing* (SAS) is to share assignment  $\text{seed}_{\mathbf{w}, \mathcal{R}}$  of Receiver’s point  $\mathbf{w}$  via additive secret sharing across those positions close to its components on  $d$  dimensions as their assignments<sup>7</sup>, and then have Sender reconstruct the point’s assignment using shares from  $d$  dimensions.

Certainly, Receiver’s assignments at these positions (Receiver’s assigned coordinate system), should not be obtained in plaintext by Sender, or Sender will know which is the component of  $\mathbf{w}$  by comparing whether two adjacent positions were assigned the same shares, which violates security. Therefore, Receiver should use AHE to hide the assigned coordinate system.

<sup>7</sup> There are  $2\delta + 1$  positions centered around each component and their  $2\delta + 1$  assignments are the same secret share.



**Conversion from Toy Protocol to UFmap.** The toy protocol satisfies the first requirement of Fmap in simplified setting. For crossing the gap between it and UFmap, we should enhance the design to fully meet all three requirements.

By introducing the assumption from [1] that each Receiver’s point maintains a distance of more than  $2\delta$  on at least one dimension from the others, we can ensure that each Receiver’s point has at least a share that is independent of the others. Consequently, the second requirement is satisfied.

Moreover, Sender can perform exactly the same as Receiver, including assigning values to coordinate system and points. Thus, if the same assumption also holds for Sender’s input, each assignment of Sender’s point in own assigned coordinate system is also imported with at least one independently uniform random share. In order to prevent the final result from being used to deduce the assignment of one’s own point in the opponent’s assigned coordinate system, we additionally embed a Diffie-Hellman (DH) subprotocol.

In summary, a point’s ID from this Fmap contains only one element called *id*, which is the sum, protected by the DH keys of both parties, of assignments of the point in assigned coordinate systems of both parties. Building on the above idea, we construct a UFmap for  $L_\infty$  distance, which we call SAS Fmap. Since its expansion rate is 1, SAS Fmap is capable of circumventing complexity bottlenecks in FPSI protocols for  $L_{p \in [1, \infty]}$  distance.

## 2.5 Applications of Fmap

**mqFRPMT from sUFmap.** Chen et al. [9] demonstrate the powerful capabilities of mqRPMT as a central block in their private set operation (PSO) framework. An attractive idea is to use the fuzzy version of mqRPMT to provide a unified framework for FPSI and its variants. Thus, we propose multi-query fuzzy RPMT (mqFRPMT).

Roughly speaking, mqFRPMT is a two-party protocol between Sender holding  $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m)$  and Receiver holding  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n)$ . After invoking of mqFRPMT for distance  $\text{dist}(\cdot, \cdot)$  and threshold  $\delta$ , Receiver learns an indication bit vector  $\mathbf{e} = (e_1, \dots, e_m) \in \{0, 1\}^m$  such that  $e_i$  equals to 1 if and only if there exists a point  $\mathbf{w}_j \in \mathbf{W}$  meeting  $\text{dist}(\mathbf{q}_i, \mathbf{w}_j)$  is not greater than  $\delta$ , while Sender learns nothing.

We present a generic construction of mqFRPMT from sUFmap, OKVS, and fuzzy matching. Firstly, Receiver and Sender invoke sUFmap to get identifiers for their points. The first requirement of sUFmap guarantees that a Sender’s point and a Receiver’s point have a same identifier when they are close enough. For each Receiver’s point, Receiver generates keys with the point’s identifiers, and uses the message required to execute fuzzy matching with this point as value. Using these key-value pairs, Receiver encodes an OKVS and sends it to Sender. Sender decodes the OKVS using keys from identifiers of Sender’s points and continues to execute fuzzy matching, which will eliminate false positives in sUFmap result, ultimately allowing Receiver to obtain the result of mqFRPMT.

**FPSI from Fmap.** Consider a general Fmap that might not be an sUFmap. For each point of Sender, Receiver obtains multiple fuzzy matching results instead of one, and the number of 1 in these results may reveal additional information about this Sender’s point to Receiver, violating the security of mqFRPMT.

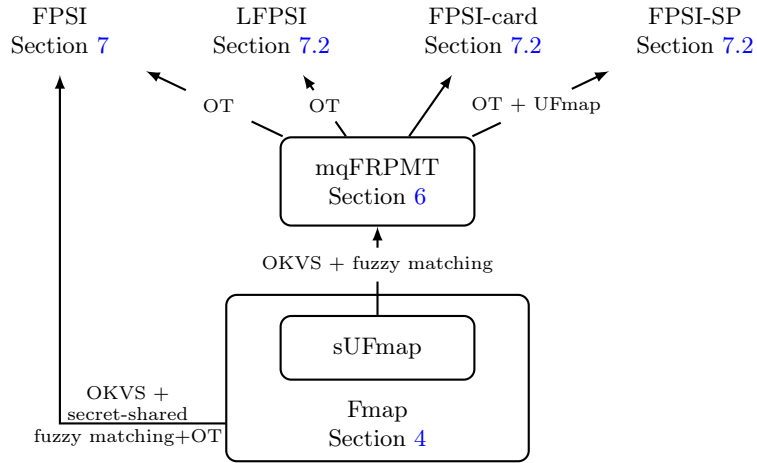
Using fuzzy matching that outputs secret shares can solve this problem. Details will be shown in Section 7.1.

## 2.6 Applications of mqFRPMT

With oblivious transfer (OT), we can derive FPSI, LFPSI, and FPSI-card from mqFRPMT by adopting the exact same approaches as that from mqRPMT to obtain PSI, LPSI, and PSI-card.

**Special Variant FPSI-SP from mqFRPMT and UFmap.** As an exception, FPSI-SP cannot be simply realized by replicating the framework of PSI because of its asymmetry. We observe that, Sender can obtain unique identifiers of Receiver’s points in FPSI-SP result from UFmap, although Sender does not know which identifiers. Therefore, if Sender uses the result of UFmap as points’ labels, Receiver can learn the corresponding identifiers of points in FPSI-SP result by invoking LFPSI with Sender. At last, Receiver can trace back to get the result of FPSI-SP with these identifiers.

Fig.2 gives a pictorial overview of our work.



**Fig. 2.** Summary of our work. The rectangles denote notions newly in this work.

### 3 Preliminaries

#### 3.1 Notation

We use  $\kappa$ ,  $\lambda$  to denote the computational and statistical security parameters respectively. We use  $[n]$  to denote the set  $\{1, 2, \dots, n\}$  and  $[n, m]$  to denote the set  $\{n, n+1, \dots, m\}$ . We assume that every set  $X$  of size  $|X|$  has a default order (e.g., lexicographical order), and represent it as  $X = (x_1, \dots, x_{|X|}) = (x_i)_{i \in [|X|]}$ .

We use  $\leftarrow$  to denote assignment and  $x \stackrel{R}{\leftarrow} X$  to denote sampling  $x$  uniformly at random from  $X$ . A function is negligible in  $\ell$ , written  $\text{negl}(\ell)$ , if it vanishes faster than the inverse of any polynomial in  $\ell$ .  $\mathbf{x} \parallel \mathbf{y}$  is the concatenation of two strings  $\mathbf{x}$  and  $\mathbf{y}$ . For a key-value pairs multiset  $\text{List}$ , we use  $\text{List}[k]$  to denote the value for key  $k$ .

For parameters in FPSI, we use  $d$  to denote the dimension of space,  $\delta$  to denote the threshold, and  $\text{dist}(\cdot, \cdot)$  to denote the distance function. We use  $\mathcal{H}$ ,  $L_\infty$ , and  $L_p$  distance as Hamming, infinite norm, and Minkowski distance, respectively. To simplify the statement, we use  $L_{p \in [1, \infty]}$  to represent the union of  $L_{p \in [1, \infty)}$  and  $L_\infty$ .

In FPSI, we use  $\mathcal{S}$  to denote Sender, who holds set  $\mathbf{Q} \in \mathbb{U}^{d \times m}$  of size  $m$ , and  $\mathcal{R}$  to denote Receiver, who holds set  $\mathbf{W} \in \mathbb{U}^{d \times n}$  of size  $n$ . Here we use  $2^u$  to denote the size of alphabet  $\mathbb{U}$ . We use  $\mathbf{q}_j$  and  $\mathbf{w}_i$  as points in  $\mathbf{Q}$  and  $\mathbf{W}$  respectively.  $q_{j,k}$  represents the component of point  $\mathbf{q}_j$  on dimension  $k$ , and  $w_{i,k}$  is analogous.  $\text{ball}_{\mathbf{w}_i}^{\text{dist}(\cdot, \cdot)}$  represents a  $d$ -dimensional ball with  $\mathbf{w}_i$  as center and  $\delta$  as radius. We use  $\mathcal{I}$  to denote the identifier universe.  $\text{ID}(\mathbf{q}_j), \text{ID}(\mathbf{w}_i) \subset \mathcal{I}$  are sets of  $\mathbf{q}_j$ 's identifiers and  $\mathbf{w}_i$ 's identifiers respectively.

Specifically, for any two points  $\mathbf{q}, \mathbf{w} \in \mathbb{U}^d$ , Hamming distance over  $\mathbb{U}$  is  $\mathcal{H}_{\mathbb{U}}(\mathbf{q}, \mathbf{w}) = \mathcal{H}(\mathbf{q}, \mathbf{w}) = \sum_{k=1}^d (q_k \neq w_k)$ , and Hamming distance over  $\mathbb{U}^P$  is  $\mathcal{H}_{\mathbb{U}^P}(\mathbf{q}, \mathbf{w}) = \sum_{k'=0}^{\frac{d}{P}-1} (\widetilde{q}_{k'} \neq \widetilde{w}_{k'})$ , where  $\widetilde{q}_{k'} = q_{k' \cdot P + 1} \| q_{k' \cdot P + 2} \| \dots \| q_{k' \cdot P + P}$  and  $\widetilde{w}_{k'} = w_{k' \cdot P + 1} \| w_{k' \cdot P + 2} \| \dots \| w_{k' \cdot P + P}$ .

#### 3.2 Additively Homomorphic Encryption

One of the main ingredients in our protocols is the additively homomorphic encryption (AHE) scheme. An AHE scheme is an encryption scheme for which it is possible to compute an encryption of the sum of two messages by just performing operations on ciphertexts of these messages. An AHE scheme with plaintext space  $\mathcal{P}$  and ciphertext space  $\mathcal{C}$  contains the following algorithms:

- $(\text{pk}, \text{sk}) \leftarrow \text{Gen}(1^\kappa)$ : takes the computational security parameter  $\kappa$  as input, then outputs the public key  $\text{pk}$  and private key  $\text{sk}$ .
- $c \leftarrow \text{Enc}_{\text{pk}}(m)$ : takes the public key  $\text{pk}$  and plaintext  $m \in \mathcal{P}$  as input, then outputs a ciphertext  $c \in \mathcal{C}$ .
- $m \leftarrow \text{Dec}_{\text{sk}}(c)$ : takes the private key  $\text{sk}$  and ciphertext  $c \in \mathcal{C}$  as input, then outputs a plaintext  $m \in \mathcal{P}$  if  $c \leftarrow \text{Enc}_{\text{pk}}(m)$ .

- $c \leftarrow c' \oplus_{\text{pk}} c''$ : takes the public key  $\text{pk}$  and two ciphertexts  $c', c'' \in \mathcal{C}$  as input, then outputs a ciphertext  $c \in \mathcal{C}$ . If  $c' \leftarrow \text{Enc}_{\text{pk}}(m')$  and  $c'' \leftarrow \text{Enc}_{\text{pk}}(m'')$ , then it holds that  $\text{Dec}_{\text{sk}}(c) = m' + m''$ .

We require AHE to satisfy IND-CPA security [6], which means that for any plaintext  $m \in \mathcal{P}$ , the distributions  $\{c \leftarrow \text{Enc}_{\text{pk}}(m)\}$  and  $\{c \xleftarrow{\mathcal{R}} \mathcal{C}\}$  are computationally indistinguishable.

ElGamal cryptosystem [14] is an AHE satisfying IND-CPA security under the decisional Diffie-Hellman (DDH) assumption. In our Fmap construction, we explicitly instantiate AHE by ElGamal’s cryptosystem to reduce the security of Fmap protocol to DDH assumption. Moreover, Paillier cryptosystem [29] and Goldwasser-Micali cryptosystem [21] are both AHE satisfying IND-CPA security.

### 3.3 Oblivious Key-Value Store

The oblivious key-value store (OKVS) is a data structure consisting of `Encode` and `Decode` algorithms that enables encoding  $n$  key-value pairs such that an adversary can not infer the original input keys with the encoding result, when the input values are random [16].

**Definition 1 (Oblivious Key-Value Store [16]).** *An oblivious key-value store (OKVS) is parameterized by a set  $\mathcal{K}$  of keys, a set  $\mathcal{V}$  of values, and a random value  $r \in \{0, 1\}^\kappa$ . It consists of two algorithms:*

- $D \leftarrow \text{Encode}(\{(k_1, v_1), \dots, (k_n, v_n)\}, r)$ : takes as input a set of  $\{(k_i, v_i)\}_{i \in [n]} \subset \mathcal{K} \times \mathcal{V}$  and outputs the object  $D \in \mathcal{V}^m$  (or, with statistically small probability, an error indicator  $\perp$ ).
- $v \leftarrow \text{Decode}(D, k, r)$ : takes as input an object  $D$ , a key  $k \in \mathcal{K}$ , and outputs a value  $v \in \mathcal{V}$ .

An OKVS must satisfy the following two properties:

- **Correctness.** For all  $A \subset \mathcal{K} \times \mathcal{V}$  with distinct keys:

$$(k, v) \in A \text{ and } \perp \neq D \leftarrow \text{Encode}(A, r) \Rightarrow \text{Decode}(D, k, r) = v$$

- **Obliviousness.** For all pairs of sets of  $n$  distinct keys  $\{k_1^0, \dots, k_n^0\} \subset \mathcal{K}$  and  $\{k_1^1, \dots, k_n^1\} \subset \mathcal{K}$ , if `Encode` does not output  $\perp$  for  $\{(k_1^0, v_1), \dots, (k_n^0, v_n)\}$  or  $\{(k_1^1, v_1), \dots, (k_n^1, v_n)\}$ , then  $\{D | v_i \xleftarrow{\mathcal{R}} \mathcal{V}, i \in [n], \text{Encode}(\{(k_1^0, v_1), \dots, (k_n^0, v_n)\}, r)\}$  is computationally indistinguishable to  $\{D | v_i \xleftarrow{\mathcal{R}} \mathcal{V}, i \in [n], \text{Encode}(\{(k_1^1, v_1), \dots, (k_n^1, v_n)\}, r)\}$ .

In addition, our Fmap and mqFRPMT protocols require independence property for OKVS, which means decoding a non-encoded key will yield a uniformly random result. Bienstoc et al. [3] prove that their RB-OKVS satisfies independence property<sup>8</sup>.

<sup>8</sup> They call this property “random decoding”.

**Definition 2 (Independence [17]).** *An OKVS satisfies independence property if, for any  $A = \{(k_1, v_1), \dots, (k_n, v_n)\}$  with distinct keys and  $k' \notin \{k_1, \dots, k_n\}$ , if Encode does not output  $\perp$  for  $A$ , the output of  $\text{Decode}(D, k', r)$  is statistically indistinguishable to a uniformly random element in  $\mathcal{V}$ , where  $D \leftarrow \text{Encode}(A, r)$ .*

For evaluating the efficiency of OKVS, there are typically three measures: rate, encoding cost, and decoding cost. The rate is the ratio between number  $n$  of input pairs and output size  $m$ . Recent OKVS constructions [3, 16, 32] achieve constant rate,  $\mathcal{O}(n\lambda)$  encoding cost, and  $\mathcal{O}(\lambda)$  decoding cost.

### 3.4 Oblivious Transfer

Oblivious transfer (OT) [31] is a foundational primitive used in various multi-party computation protocols.

The ideal functionality for 1-out-of-2 OT is given in Fig.3.

<p>PARAMETERS: Sender <math>\mathcal{S}</math>, Receiver <math>\mathcal{R}</math>; Message length <math>\ell</math>.</p> <p>FUNCTIONALITY:</p> <ul style="list-style-type: none"> <li>– Wait an input <math>(m_0, m_1) \in \{0, 1\}^{\ell \times 2}</math> from <math>\mathcal{S}</math>.</li> <li>– Wait an input <math>e \in \{0, 1\}</math> from <math>\mathcal{R}</math>.</li> <li>– Return <math>m_e</math> to <math>\mathcal{R}</math>.</li> </ul>
--

**Fig. 3.** Ideal Functionality for 1-out-of-2 OT  $\mathcal{F}_{\text{OT}}$

### 3.5 Fuzzy Matching

Fuzzy matching (FMatch) enables Sender and Receiver determine whether the Sender's point  $\mathbf{q}$  and the Receiver's point  $\mathbf{w}$  satisfy  $\text{dist}(\mathbf{q}, \mathbf{w}) \leq \delta$  [1]. Obviously, if let the protocol return the result by secret shares, we get secret-shared FMatch (ssFMatch). Their ideal functionalities are given in Fig.4.

<p>PARAMETERS: Sender <math>\mathcal{S}</math>, Receiver <math>\mathcal{R}</math>; Dimension <math>d</math>; Distance function <math>\text{dist}(\cdot, \cdot)</math>; Distance threshold <math>\delta</math>.</p> <p>FUNCTIONALITY:</p> <ul style="list-style-type: none"> <li>– Wait an input <math>\mathbf{q} \in \mathbb{U}^d</math> from <math>\mathcal{S}</math>.</li> <li>– Wait an input <math>\mathbf{w} \in \mathbb{U}^d</math> from <math>\mathcal{R}</math>.</li> <li>– Compute <math>e \in \{0, 1\}</math>, where <math>e = 1</math> if and only if <math>\text{dist}(\mathbf{q}, \mathbf{w}) \leq \delta</math>.</li> <li>– <b>FMatch:</b> Return <math>e</math> to <math>\mathcal{R}</math>.</li> <li>– <b>ssFMatch:</b> Return <math>e_{\mathcal{S}} \xleftarrow{\mathcal{R}} \{0, 1\}</math> to <math>\mathcal{S}</math>. Return <math>e_{\mathcal{R}} \leftarrow e \oplus e_{\mathcal{S}}</math> to <math>\mathcal{R}</math>.</li> </ul>
--

**Fig. 4.** Ideal Functionality for (Secret-Shared) Fuzzy Matching  $\mathcal{F}_{(\text{ss})\text{FMatch}}$

Our work is concerned with Hamming and  $L_{p \in [1, \infty]}$  distances. For Hamming distance, considering points of two parties as their Boolean shares in the case  $\mathbb{U} = \{0, 1\}$ , there is a trivial approach of (ss)FMatch that consists of OT-based conversion of Boolean sharing to Arithmetic sharing and (secret-shared) secure comparing [30]. This approach has  $\mathcal{O}(d)$  communication and computation costs. For  $L_{p \in [1, \infty]}$  distance, Baarsen and Pu provide constructions of FMatch in [1].

One of building blocks we use is a special case of FMatch, fuzzy matching for interval (IFmat), by which Sender with an interval and Receiver with a number can check whether this number belongs to the interval. Moreover, if  $\delta$  is set to 0, this special case of IFmat is private equality test (PEqT). Their ideal functionalities are given in Fig.5.

Using the idea of prefix matching, Chakraborti et al. [7] propose a semi-honest secure IFmat protocol achieving communication and computation complexities scaling logarithmically in the threshold. In all our constructions, we will use their protocol to instantiate IFmat and PEqT.

<p>PARAMETERS: Sender <math>\mathcal{S}</math>, Receiver <math>\mathcal{R}</math>; Threshold <math>\delta</math>.</p> <p>FUNCTIONALITY:</p> <ul style="list-style-type: none"> <li>- Wait an input <math>a \in \mathbb{Z}</math> from <math>\mathcal{S}</math>.</li> <li>- Wait an input <math>x \in \mathbb{Z}</math> from <math>\mathcal{R}</math>.</li> <li>- Return <math>e</math> to <math>\mathcal{R}</math>, where <math>e = 1</math> if and only if: <ul style="list-style-type: none"> <li><b>IFmat:</b> <math>x \in [a - \delta, a + \delta]</math></li> <li><b>PEqT:</b> <math>x = a</math></li> </ul> </li> </ul>
---

**Fig. 5.** Ideal Functionalities for IFmat  $\mathcal{F}_{\text{IFmat}}$  and PEqT  $\mathcal{F}_{\text{PEqT}}$

### 3.6 Security Model

In this work, we consider only the semi-honest model, where adversaries strictly follow the protocol specification but try to learn more than allowed by inspecting the protocol transcript. We use the standard semi-honest security definition for two-party computation [28] in this work.

Let  $\Pi$  be a two-party protocol for computing function  $f(X, Y)$  where party  $\mathcal{S}$  has input  $X$  and output  $f_{\mathcal{S}}(X, Y)$ ; party  $\mathcal{R}$  has input  $Y$  and output  $f_{\mathcal{R}}(X, Y)$ . Let  $\text{view}_{\mathcal{S}}^{\Pi}(X, Y)$  and  $\text{view}_{\mathcal{R}}^{\Pi}(X, Y)$  be the views of  $\mathcal{S}$  and  $\mathcal{R}$  in the protocol. Let  $\text{output}(X, Y)$  be the output of both parties in protocol  $\Pi$ .

**Definition 3 (Semi-Honest Security).** *A protocol  $\Pi$  is said to securely compute functionality  $f$  in the semi-honest model if for every probabilistic polynomial-time adversary  $\mathcal{A}$  there exists probabilistic polynomial-time simulators  $\text{Sim}_{\mathcal{S}}$  and  $\text{Sim}_{\mathcal{R}}$  such that for all inputs  $X$  and  $Y$ ,*

$$\begin{aligned} \{\text{view}_{\mathcal{S}}^{\Pi}(X, Y), \text{output}(X, Y)\} &\stackrel{c}{\approx} \{\text{Sim}_{\mathcal{S}}(X, f_{\mathcal{S}}(X, Y)), f(X, Y)\} \\ \{\text{view}_{\mathcal{R}}^{\Pi}(X, Y), \text{output}(X, Y)\} &\stackrel{c}{\approx} \{\text{Sim}_{\mathcal{R}}(Y, f_{\mathcal{R}}(X, Y)), f(X, Y)\} \end{aligned}$$

## 4 Fuzzy Mapping

In this section, we provide the formal definitions for fuzzy mapping (Fmap) and its expansion rate, and list existing instances of Fmap.

### 4.1 Definition of Fmap

As mentioned in Section 2.2, with Fmap, both parties can map each of their points to a set of identifiers. If a Sender's point and a Receiver's point are close enough, they will have a same identifier, and point pairs formed in this way will be further filtered by FMatch to obtain FPSI result.

The formal definition of Fmap is as follows.

**Definition 4 (Fuzzy Mapping).** *A two-party protocol  $\Pi$ , where Sender's input  $\mathbf{Q} = (\mathbf{q}_j)_{j \in [m]} \in \mathbb{U}^{d \times m}$  results in  $\text{ID}(\mathbf{Q}) = (\text{ID}(\mathbf{q}_j))_{j \in [m]}$  and Receiver's input  $\mathbf{W} = (\mathbf{w}_i)_{i \in [n]} \in \mathbb{U}^{d \times n}$  results in  $\text{ID}(\mathbf{W}) = (\text{ID}(\mathbf{w}_i))_{i \in [n]}$ <sup>9</sup>, is a semi-honest secure fuzzy mapping (Fmap) protocol  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$  of threshold  $\delta$  for  $\text{dist}(\cdot, \cdot)$ , if and only if  $\Pi$  satisfies:*

- **Correctness.** *For any two points  $\mathbf{q}_j \in \mathbf{Q}$  and  $\mathbf{w}_i \in \mathbf{W}$ :*

$$\text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta \implies \text{ID}(\mathbf{q}_j) \cap \text{ID}(\mathbf{w}_i) \neq \emptyset$$

- **Distinctiveness.** *For the output  $\text{ID}(\mathbf{W})$  of Receiver, the following equation holds:*

$$\Pr[\exists i, i' \in [n], \text{s.t. } (i \neq i') \wedge (\text{ID}(\mathbf{w}_i) \cap \text{ID}(\mathbf{w}_{i'}) \neq \emptyset)] = \text{negl}(\kappa)$$

- **Security.** *Considering corrupt semi-honest Sender, for any  $\mathbf{Q} \in \mathbb{U}^{d \times m}$  and any  $\mathbf{W}, \mathbf{W}' \in \mathbb{U}^{d \times n}$ , it holds that*

$$\text{view}_{\mathcal{S}}^{\Pi}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}) \stackrel{c}{\approx} \text{view}_{\mathcal{S}}^{\Pi}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}')$$

*Considering corrupt semi-honest Receiver, for any  $\mathbf{W} \in \mathbb{U}^{d \times n}$  and any  $\mathbf{Q}, \mathbf{Q}' \in \mathbb{U}^{d \times m}$ , it holds that*

$$\text{view}_{\mathcal{R}}^{\Pi}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}) \stackrel{c}{\approx} \text{view}_{\mathcal{R}}^{\Pi}(\kappa, \lambda; \mathbf{Q}', \mathbf{W})$$

To quantify the expansion of inputs, we define the expansion rate of Fmap.

**Definition 5 (Expansion Rate).** *The expansion rate of Fmap for Sender's input is*

$$\text{rate}_{\mathcal{S}} = \frac{1}{m} \sum_{j \in [m]} |\text{ID}(\mathbf{q}_j)|$$

<sup>9</sup>  $\text{ID}(\mathbf{q}_j), \text{ID}(\mathbf{w}_i) \subset \mathcal{I}$ ; for security reason, we default to  $|\text{ID}(\mathbf{q}_j)| = |\text{ID}(\mathbf{q}_{j'})|$  for different  $j, j' \in [m]$  and  $|\text{ID}(\mathbf{w}_i)| = |\text{ID}(\mathbf{w}_{i'})|$  for different  $i, i' \in [n]$ .

The expansion rate of Fmap for Receiver's input is

$$\text{rate}_{\mathcal{R}} = \frac{1}{n} \sum_{i \in [n]} |\text{ID}(\mathbf{w}_i)|$$

The expansion rate of Fmap is

$$\text{rate} = \max \{ \text{rate}_{\mathcal{S}}, \text{rate}_{\mathcal{R}} \}$$

**Definition 6 (Sender's Unit Fmap).** An Fmap is a Sender's unit Fmap (sUFmap) if and only if its expansion rate for Sender's input is 1.

**Definition 7 (Unit Fmap).** An Fmap is a unit Fmap (UFmap) if and only if its expansion rate is 1.

The efficiency of an Fmap instance is measured by:

- **Expansion rate:** Expansion rate of Fmap is positively related to complexities of FPSI based on it, thus we hope it to be as small as possible. Note that the optimal expansion rate is 1.
- **Communication complexity:** As a two-party protocol, Fmap's own efficiency is influenced by its communication complexity. Since many Fmap instances degenerate into two algorithms executed by Sender and Receiver respectively, they have no communication.
- **Computation complexity:** The computation complexity of Fmap is also a factor to consider, and it is clear that the lower bound of computation complexity is the size of output.

A crucial observation is that as long as the complexity of Fmap does not exceed that of the subsequent part, the asymptotic complexity of the entire FPSI will not be affected.

Therefore, a high-level intuition is that we can improve the overall efficiency of FPSI by reducing expansion rate of Fmap at the cost of a tolerable increase in complexity of Fmap.

**Lemma 1 (Reduction of Fmap).** If there are two distance functions  $\text{dist}(\cdot, \cdot)$  and  $\text{dist}'(\cdot, \cdot)$  such that  $\text{dist}(\mathbf{q}, \mathbf{w}) \leq \text{dist}'(\mathbf{q}, \mathbf{w})$  holds for any two points  $\mathbf{q}$  and  $\mathbf{w}$ , then Fmap protocol  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$  realizes Fmap for  $\text{dist}'(\cdot, \cdot)$ .

*Proof.* As Fmap for  $\text{dist}'(\cdot, \cdot)$ , the distinctiveness and security of  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$  are guaranteed by Definition 4 of Fmap for  $\text{dist}(\cdot, \cdot)$ .

Now consider the correctness of  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$  for  $\text{dist}'(\cdot, \cdot)$ . For any  $j \in [m]$  and  $i \in [n]$ , when  $\text{dist}'(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ . We have  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \text{dist}'(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ . Thus the correctness for  $\text{dist}'(\cdot, \cdot)$  is guaranteed by the correctness for  $\text{dist}(\cdot, \cdot)$ .

Therefore,  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$  realizes Fmap for  $\text{dist}'(\cdot, \cdot)$ .

**Corollary 1.** For any  $P \in \mathbb{N}^+$  and any two points  $\mathbf{q}, \mathbf{w} \in \mathbb{U}^d$ , we have  $\mathcal{H}_{\mathbb{U}^P}(\mathbf{q}, \mathbf{w}) \leq \mathcal{H}_{\mathbb{U}}(\mathbf{q}, \mathbf{w})$ . According to Lemma 1,  $\Pi_{\text{Fmap}}^{\mathcal{H}_{\mathbb{U}^P}}$  can be seen as  $\Pi_{\text{Fmap}}^{\mathcal{H}_{\mathbb{U}}}$ .

**Corollary 2.** For any two points  $\mathbf{q}, \mathbf{w} \in \mathbb{U}^d$ , we have  $L_{\infty}(\mathbf{q}, \mathbf{w}) \leq L_{p \in [1, \infty]}(\mathbf{q}, \mathbf{w})$ . According to Lemma 1,  $\Pi_{\text{Fmap}}^{L_{\infty}}$  can be seen as  $\Pi_{\text{Fmap}}^{L_{p \in [1, \infty]}}$ .



## 4.2 Existing Fmap Constructions

Table 2 lists existing constructions that fit to Definition 4 of Fmap. Many of existing FPSI protocols are constructed using Fmap instances listed in this table.

As can be seen in Table 2, all of previous Fmap instances have communication cost of zero and computation cost of theoretical lower bound but expansion rate of pretty big value, while our UniqC Fmap is the only non-trivial Fmap for Hamming distance and our SAS Fmap achieves the optimal expansion rate but has non-optimal complexities. This trade-off works well because complexity bottlenecks in previous FPSI protocols actually come from expansion rates; rather than costs of invoking Fmap.

**Table 2.** Comparison of Fmap instances, where  $m$  and  $n$  are set sizes of Sender’s and Receiver’s inputs, respectively.  $d$  is the space dimension.  $\rho \in (0, 1)$  is a parameter in LSH scheme. We ignore multiplicative factors of the computational security parameter  $\kappa$  and statistical security parameter  $\lambda$ . In computation cost, we ignore non-cryptographic operations, such as the binary search and polynomial interpolation in plaintext form.

Fmap	Distance	Assumption	rate <sub>S</sub>	rate <sub>R</sub>	Communication	Computation	
						Sender	Receiver
Naive [15]	Anyone	–	$n$	1	–	$\mathcal{O}(nm)$	$\mathcal{O}(n)$
Spatial Hashing [17]	$L_\infty$	R. $l_{min} > 2\delta$	1	$\mathcal{O}(2^d)$	–	$\mathcal{O}(m)$	$\mathcal{O}(2^d n)$
Separated Balls [1]	$L_\infty$	R. disj. proj.	$d$	$\mathcal{O}(\delta)$	–	$\mathcal{O}(dm)$	$\mathcal{O}(\delta n)$
LSH [1]	$L_{p \in [1, \infty)}$	R. $l_{min} > \frac{1}{\rho}\delta$	$\mathcal{O}(n^\rho \log n)$	$\mathcal{O}(n^\rho)$	–	$\mathcal{O}((n^\rho \log n)m)$	$\mathcal{O}(n^{\rho+1})$
Ours: UniqC	Hamming	R. UniqC	$d$	$\delta + 1$	–	$\mathcal{O}(dm)$	$\mathcal{O}(\delta n)$
Ours: SAS	$L_\infty$	R $\wedge$ S. disj. proj.	1	1	$\mathcal{O}(\delta dm + \delta dn)$	$\mathcal{O}(\delta dm + n)$	$\mathcal{O}(m + \delta dn)$

**Naive Fmap.** A straightforward approach of FPSI is to perform FMatch on all pairs of these two inputs to obtain results. This idea can be abstracted into a naive Fmap: for each Sender’s point  $\mathbf{q}_j$ ,  $\text{ID}(\mathbf{q}_j)$  is  $\{i\}_{i \in [n]}$ , thus rate<sub>S</sub> is  $\mathcal{O}(n)$ ; for each Receiver’s point  $\mathbf{w}_i$ ,  $\text{ID}(\mathbf{w}_i)$  is  $\{i\}$  where  $i$  is the index of this point, thus rate<sub>R</sub> is 1. It does not rely on any assumptions and can be used in FPSI for any distance function. Many existing FPSI protocols [7, 10, 15, 22, 34, 37] for Hamming distance adopt naive Fmap, which leads to the  $m \cdot n$  blowup in their complexities.

**Spatial Hashing.** Spatial hashing technique is proposed by [17]. The Fmap based on Spatial Hashing is abstracted from the scheme in Sec 4.3.2 of [17], in which the entire  $d$ -dimensional space is divided into several hypercubes of side-length  $\mathcal{O}(\delta)$ , referred to as “cells”. For each Sender’s point  $\mathbf{q}_j$ ,  $\text{ID}(\mathbf{q}_j)$  is a set of the cell containing  $\mathbf{q}_j$ , thus rate<sub>S</sub> is 1; for each Receiver’s point  $\mathbf{w}_i$ ,  $\text{ID}(\mathbf{w}_i)$  is a set of cells that intersect with  $\text{ball}_{\mathbf{w}_i}^{L_\infty}$ , thus rate<sub>R</sub> is  $\mathcal{O}(2^d)$ . To satisfy distinctiveness, this Fmap assumes that Receiver’s points are far enough apart from each other.

**Separated Balls.** This Fmap from FPSI protocol for  $L_\infty$  distance in [1] relies on the assumption that for each Receiver's point  $\mathbf{w}_i$ , there exists at least one dimension on which  $\text{ball}_{\mathbf{w}_i}^{L_\infty}$ 's projection is far enough apart from the others. For each Sender's point  $\mathbf{q}_j$ ,  $\text{ID}(\mathbf{q}_j)$  is  $\{q_{j,k}\}_{k \in [d]}$ , thus  $\text{rate}_S$  is  $\mathcal{O}(d)$ ; for each Receiver's point  $\mathbf{w}_i$ , let us use  $k_i$  to denote the dimension on which  $\mathbf{w}_{i,k_i}$  is far enough apart from the others, then  $\text{ID}(\mathbf{w}_i)$  is  $\{k_i \parallel (w_{i,k_i} + t)\}_{t \in [-\delta, \delta]}$ , thus  $\text{rate}_R$  is  $\mathcal{O}(\delta)$ .

**LSH.** LSH is introduced for FPSI in [1], and it leads to another Fmap. The Fmap based on LSH constructs  $L$  hash tables of capacity  $T$ . According to the definition of LSH, when the distance between two points is close enough, there exists a hash table in which these two points are hashed to the same bucket. In [1], LSH-based protocol chooses  $L = \mathcal{O}(n^\rho)$  and  $T = \mathcal{O}(\log n)$ . In this Fmap, for each Sender's point, Sender traverses  $T$  buckets in each table, thus  $\text{rate}_S$  is  $\mathcal{O}(n^\rho \log n)$ ; Receiver hashes each Receiver's point to one bucket in each table, thus  $\text{rate}_R$  is  $\mathcal{O}(n^\rho)$ . The distinctiveness of LSH-based Fmap is guaranteed by collision resistance property of LSH.

**Our Fmap.** Details of our Fmap will be given later.

## 5 New Fmap Constructions

In this section, we present new semi-honest secure Fmap constructions for Hamming and  $L_\infty$  distances, which are the infrastructure for subsequent protocols.

### 5.1 UniqC Fmap for Hamming Distance

We present a construction of semi-honest secure Fmap for Hamming distance, which is denoted by UniqC Fmap. Similar to existing Fmap constructions, UniqC Fmap consists of two algorithms for Sender and Receiver respectively due to the absence of interaction. For each Receiver's point, Receiver chooses its  $\delta+1$  unique components as its ID, while Sender selects all  $d$  components of a point as its ID. The formal description of UniqC Fmap is shown in Fig.6.

**Definition 8 (Unique Set).** *For a point  $\mathbf{w} \in \mathbf{W}$  in a  $d$ -dimensional space,  $\mathbf{w}$  has a unique component  $w_k$ , if and only if its component on dimension  $k$  is different from that of any other point in  $\mathbf{W}$ . A point  $\mathbf{w}$  is unique, if and only if  $\mathbf{w}$  has at least  $\delta+1$  unique components. A set  $\mathbf{W}$  is unique, if and only if all points in  $\mathbf{W}$  are unique.*

**Lemma 2 (Uniform Distribution).** *In a  $d$ -dimensional space, if points in set  $\mathbf{W}$  are uniformly distributed, then the probability that  $\mathbf{W}$  is unique is  $1 - \text{negl}(d)$ <sup>10</sup>.*

<sup>10</sup> Here we default that the size of alphabet  $\mathbb{U}$  is greater than  $n$ . When  $|\mathbb{U}| = 2^u \leq n$ , we can pack  $P$  components as one super-component such that  $|\mathbb{U}^P| = 2^{uP} > n$ , thus

<p>PARAMETERS:</p> <ul style="list-style-type: none"> <li>– Sender <math>\mathcal{S}</math> and Receiver <math>\mathcal{R}</math>.</li> </ul> <p>INPUT OF <math>\mathcal{S}</math>: <math>\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m) \in \mathbb{U}^{d \times m}</math>.</p> <p>INPUT OF <math>\mathcal{R}</math>: <math>\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n) \in \mathbb{U}^{d \times n}</math>.</p> <p><u><math>\mathcal{S}</math>'s UniqC Fmap (<math>\mathbf{Q}</math>):</u></p> <ol style="list-style-type: none"> <li>1. For each <math>j \in [m]</math>, <math>\mathcal{S}</math> computes <math>\text{ID}(\mathbf{q}_j) \leftarrow \{k \mid q_{j,k}\}_{k \in [d]}</math>.</li> <li>2. Return <math>\text{ID}(\mathbf{Q}) = (\text{ID}(\mathbf{q}_1), \dots, \text{ID}(\mathbf{q}_m))</math>.</li> </ol> <p><u><math>\mathcal{R}</math>'s UniqC Fmap (<math>\mathbf{W}</math>):</u></p> <ol style="list-style-type: none"> <li>1. For each <math>i \in [n]</math>, <math>\mathcal{R}</math> computes <math>\text{ID}(\mathbf{w}_i) \leftarrow \left\{ u_k^i \mid w_{i,u_k^i} \right\}_{k \in [\delta+1]}</math>, where <math>u_k^i \in [d]</math> is a dimension such that <math>w_{i,u_k^i} \neq w_{i',u_k^i}</math> holds for any <math>i' \in [n] \setminus \{i\}</math>.</li> <li>2. Return <math>\text{ID}(\mathbf{W}) = (\text{ID}(\mathbf{w}_1), \dots, \text{ID}(\mathbf{w}_n))</math>.</li> </ol>
---

**Fig. 6.** Fmap Protocol for Hamming Distance:  $\Pi_{\text{UniqC Fmap}}^{\mathcal{H}}$

*Proof.* For each  $\mathbf{w}_i \in \mathbf{W}$  and each dimension  $k \in [d]$ , we have

$$\Pr[w_{i,k} \text{ is not a unique component}] \leq \frac{n-1}{2^u}$$

Hence, the probability that  $\mathbf{w}_i$  has exactly  $\delta$  unique components is not greater than  $\binom{d}{\delta} \left(\frac{n-1}{2^u}\right)^{d-\delta}$ . By a union bound, it holds that

$$\Pr[\mathbf{w}_i \text{ is not unique}] \leq \delta \binom{d}{\delta} \left(\frac{n-1}{2^u}\right)^{d-\delta} \leq d^{\delta+1} \left(\frac{n-1}{2^u}\right)^{d-\delta} \triangleq f(d)$$

We default that  $2^u > n-1$  and thus  $f(d)$  is  $\text{negl}(d)$ .

$$\Pr[\mathbf{W} \text{ is unique}] = (1 - \Pr[\mathbf{w}_i \text{ is not unique}])^n \geq (1 - f(d))^n$$

which is  $1 - \text{negl}(d)$ .

*Remark 2.* Considering 400-dimensional bio-bit-vectors and  $\delta = 7$  in [35], we pack 16 bits into a super-component (i.e.,  $d$  and  $u$  are updated to 25 and 16 respectively), and we choose statistical security parameter  $\lambda = 40$ .

Then, when  $n < 2^{11}$ , we have

$$\Pr[\mathbf{W} \text{ is unique}] \geq \left(1 - d^{\delta+1} \left(\frac{n-1}{2^u}\right)^{d-\delta}\right)^n \geq 1 - 2^{-\lambda}$$

For UniqC Fmap, we introduce the Receiver's unique components (R. UniqC) assumption:

---

R. UniqC assumption for  $\mathcal{H}_{\text{UP}}$  holds and  $\Pi_{\text{UniqC Fmap}}^{\mathcal{H}_{\text{UP}}}$  works. According to Corollary 1, we can use  $\Pi_{\text{UniqC Fmap}}^{\mathcal{H}_{\text{UP}}}$  as  $\Pi_{\text{UniqC Fmap}}^{\mathcal{H}_{\text{U}}}$ .

*Each Receiver's point has unique components on at least  $\delta + 1$  dimensions.*

If Receiver's points are uniformly distributed, then according to Lemma 2, R. UniqC assumption holds in high-dimensional case with overwhelming probability. Thus, it is acceptable to base our construction on it. Now, we prove the protocol in Fig.6 is a semi-honest secure Fmap for Hamming distance.

**Theorem 1 (Correctness).** *The protocol presented in Fig.6 satisfies the correctness defined in Definition 4 for Hamming distance if R. UniqC assumption holds.*

*Proof.* Since R. UniqC assumption holds, the protocol presented in Fig.6 can be executed completely.

For  $\mathbf{q}_j \in \mathbf{Q}$  and  $\mathbf{w}_i \in \mathbf{W}$ , if  $\mathcal{H}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ , then  $\mathbf{q}_j$  has the same component with  $\mathbf{w}_i$  on at least  $d - \delta$  dimensions.  $\mathbf{w}_i$  has  $\delta + 1$  unique components, so one of  $\mathbf{w}_i$ 's unique components is also  $\mathbf{q}_j$ 's component. Hence, we have  $\text{ID}(\mathbf{q}_j) \cap \text{ID}(\mathbf{w}_i) \neq \emptyset$ .

**Theorem 2 (Distinctiveness).** *The protocol presented in Fig.6 satisfies the distinctiveness defined in Definition 4 if R. UniqC assumption holds.*

*Proof.* Since R. UniqC assumption holds, the protocol presented in Fig.6 can be executed completely.

The distinctiveness comes from Definition 8 of unique component.

**Theorem 3 (Security).** *The protocol presented in Fig.6 satisfies the security defined in Definition 4 if R. UniqC assumption holds.*

*Proof.* Since R. UniqC assumption holds, the protocol presented in Fig.6 can be executed completely.

Since UniqC Fmap does degenerate into two algorithms without interaction from a two-party protocol, outputs received by both parties are independent of each other's inputs. Thus, the security property is self-evident.

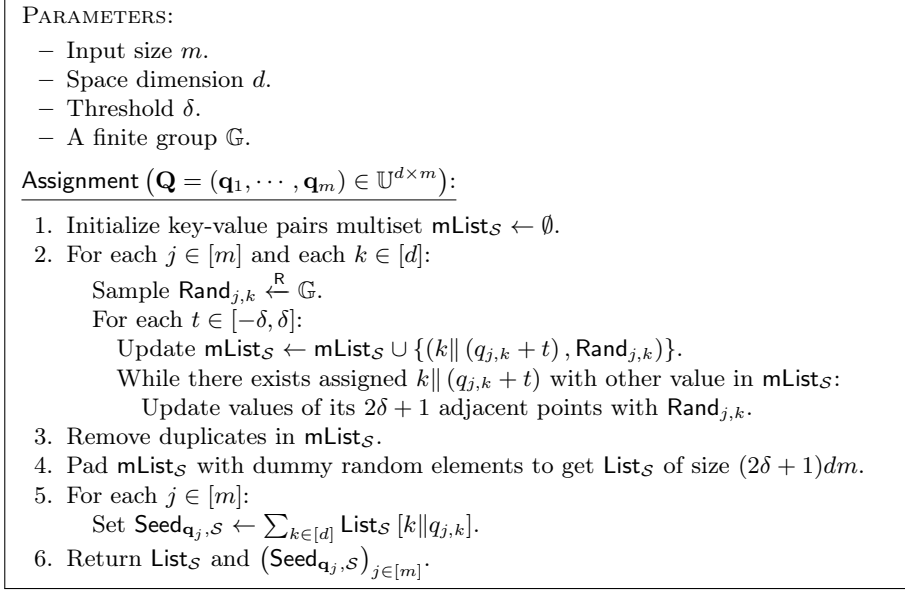
## 5.2 SAS Fmap for $L_\infty$ Distance

We present a construction of semi-honest secure UFmap for  $L_\infty$  distance, which is denoted by SAS Fmap. We use  $\text{id}_{\mathbf{q}_j}$  and  $\text{id}_{\mathbf{w}_i}$  to represent the only element in  $\text{ID}(\mathbf{q}_j)$  and  $\text{ID}(\mathbf{w}_i)$  respectively.

As described in Section 2.4, for each point in input sets, SAS Fmap generates the sum, protected by DH keys of two parties, of this point's assignments in assigned coordinate systems of two parties as its identifier.

We first deal with the assignment process with spatial additive sharing (SAS), and then utilize the assignment algorithm to construct SAS Fmap.

**Assignment Algorithm from SAS.** SAS treats a point's assignment as the sum of its components' assignments on  $d$  dimensions, thereby converting the processing of a point in  $d$ -dimensional space into the processing of  $d$  points in 1-dimensional axes. And SAS ensures that the assignment of each component of each point is also the assignment of the  $2\delta + 1$  positions centered around this component on the corresponding dimension. Our assignment algorithm is described formally in Fig.7.



**Fig. 7.** Assignment Algorithm: Assignment( $\cdot$ )

**UFmap Based on SAS.** We assume that input sets of both parties have good distribution in a high-dimensional space. Thus, we propose a semi-honest secure UFmap based on SAS for  $L_\infty$  distance.

Intuitively, both parties first use Assignment algorithm to attain their assigned coordinate systems (i.e., assigned axes of  $d$  dimensions). They encode their assigned coordinate systems into OKVS in the form of ElGamal ciphertexts and send OKVS to each other. By leveraging the homomorphism of ElGamal, both of them can obtain ciphertexts of their own points assigned in the other's coordinate system. Finally, through a masked DH subprotocol, they can securely acquire their own Fmap output. The formal description of SAS Fmap is in Fig.8.

**Definition 9 (Separated Set).** For two points  $\mathbf{q}$  and  $\mathbf{q}'$  in a  $d$ -dimensional space,  $\mathbf{q}$  collides with  $\mathbf{q}'$  on dimension  $k$  if and only if the distance between their components on dimension  $k$  is not greater than  $2\delta$ ; otherwise,  $\mathbf{q}$  is separated

PARAMETERS:

- Sender  $\mathcal{S}$  and Receiver  $\mathcal{R}$ .
- A finite group  $\mathbb{G}$  of prime order  $p$  and  $\mathbb{F}_p$ .
- EC-ElGamal scheme  $\mathcal{E} = (\text{Gen}^{\text{EIG}}, \text{Enc}^{\text{EIG}}, \text{Dec}^{\text{EIG}}, \oplus^{\text{EIG}})$  with plaintext space  $\mathbb{G}$ .
- An OKVS scheme  $(\text{Encode}, \text{Decode})$  and its random value  $r$ .

INPUT OF  $\mathcal{S}$ :  $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m) \in \mathbb{U}^{d \times m}$ .

INPUT OF  $\mathcal{R}$ :  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n) \in \mathbb{U}^{d \times n}$ .

PROTOCOL:

**Phase 1. Assignment and Summation:**

1.  $\mathcal{S}$  computes  $(\text{List}_{\mathcal{S}}, (\text{Seed}_{\mathbf{q}_j, \mathcal{S}})_{j \in [m]}) \leftarrow \text{Assignment}(\mathbf{Q})$ .
2.  $\mathcal{S}$  generates  $(\text{sk}_{\text{EIG}, \mathcal{S}}, \text{pk}_{\text{EIG}, \mathcal{S}}) \leftarrow \text{Gen}^{\text{EIG}}(1^\kappa)$ .  $\mathcal{S}$  samples  $m$  masks  $\{\text{mask}_{\mathcal{S}, j} \stackrel{\mathcal{R}}{\leftarrow} \mathbb{F}_p\}_{j \in [m]}$  and computes their inverses  $\{\text{mask}_{\mathcal{S}, j}^{-1}\}_{j \in [m]}$ .
3.  $\mathcal{S}$  initializes set  $\widetilde{\text{List}}_{\mathcal{S}} \leftarrow \emptyset$ . For each  $(\text{key}, \text{List}_{\mathcal{S}}[\text{key}]) \in \text{List}_{\mathcal{S}}$ ,  $\mathcal{S}$  updates

$$\widetilde{\text{List}}_{\mathcal{S}} \leftarrow \widetilde{\text{List}}_{\mathcal{S}} \cup \left\{ \left( \text{key}, \text{Enc}_{\text{pk}_{\text{EIG}, \mathcal{S}}}^{\text{EIG}}(\text{List}_{\mathcal{S}}[\text{key}]) \right) \right\}$$

4.  $\mathcal{S}$  encodes  $E_{\mathcal{S}} \leftarrow \text{Encode}(\widetilde{\text{List}}_{\mathcal{S}}, r)$ , and sends  $E_{\mathcal{S}}, \text{pk}_{\text{EIG}, \mathcal{S}}$  to  $\mathcal{R}$ .
5. Symmetrically,  $\mathcal{R}$  sends  $E_{\mathcal{R}} \leftarrow \text{Encode}(\widetilde{\text{List}}_{\mathcal{R}}, r)$ ,  $\text{pk}_{\text{EIG}, \mathcal{R}}$  to  $\mathcal{S}$ .
6. For each  $j \in [m]$ ,  $\mathcal{S}$  computes

$$\text{sum}_{\mathbf{q}_j}^{\text{pk}_{\text{EIG}, \mathcal{R}}} \leftarrow \bigoplus_{k \in [d]}^{\text{EIG}} \text{Decode}(E_{\mathcal{R}}, k \| q_{j,k}, r)$$

$$\text{cipher}_{\mathbf{q}_j}^{\text{pk}_{\text{EIG}, \mathcal{R}}} \leftarrow \text{mask}_{\mathcal{S}, j} \times \left( \left( \text{Enc}_{\text{pk}_{\text{EIG}, \mathcal{R}}}^{\text{EIG}}(\text{Seed}_{\mathbf{q}_j, \mathcal{S}}) \right) \oplus_{\text{pk}_{\text{EIG}, \mathcal{R}}}^{\text{EIG}} \text{sum}_{\mathbf{q}_j}^{\text{pk}_{\text{EIG}, \mathcal{R}}} \right)$$

**Phase 2. Decryption and Exchange:**

7.  $\mathcal{S}$  sends  $m$  ElGamal ciphertexts  $(\text{cipher}_{\mathbf{q}_j}^{\text{pk}_{\text{EIG}, \mathcal{R}}})_{j \in [m]}$  to  $\mathcal{R}$ .
8. Symmetrically,  $\mathcal{R}$  sends  $n$  ElGamal ciphertexts  $(\text{cipher}_{\mathbf{w}_i}^{\text{pk}_{\text{EIG}, \mathcal{S}}})_{i \in [n]}$  to  $\mathcal{S}$ .
9.  $\mathcal{S}$  samples  $\text{sk}_{\text{DH}, \mathcal{S}} \stackrel{\mathcal{R}}{\leftarrow} \mathbb{F}_p$ . For each  $i \in [n]$ ,  $\mathcal{S}$  computes

$$\text{Seed}_{\text{mk}_{\mathcal{R}, i}}^{\mathbf{w}_i} \leftarrow \text{Dec}_{\text{sk}_{\text{EIG}, \mathcal{S}}}^{\text{EIG}}(\text{cipher}_{\mathbf{w}_i}^{\text{pk}_{\text{EIG}, \mathcal{S}}})$$

$$\text{Seed}_{\text{mk}_{\mathcal{R}, i, \mathcal{S}}}^{\mathbf{w}_i} \leftarrow \text{sk}_{\text{DH}, \mathcal{S}} \cdot \text{Seed}_{\text{mk}_{\mathcal{R}, i}}^{\mathbf{w}_i}$$

10.  $\mathcal{S}$  sends  $(\text{Seed}_{\text{mk}_{\mathcal{R}, i, \mathcal{S}}}^{\mathbf{w}_i})_{i \in [n]}$  to  $\mathcal{R}$ . Symmetrically,  $\mathcal{R}$  sends  $(\text{Seed}_{\text{mk}_{\mathcal{S}, j, \mathcal{R}}}^{\mathbf{q}_j})_{j \in [m]}$ .
11. For each  $i \in [m]$ ,  $\mathcal{S}$  computes  $(\text{Seed}_{\text{mk}_{\mathcal{S}, j, \mathcal{R}, \mathcal{S}}}^{\mathbf{q}_j}) \leftarrow \text{sk}_{\text{DH}, \mathcal{S}} \cdot (\text{Seed}_{\text{mk}_{\mathcal{S}, j, \mathcal{R}}}^{\mathbf{q}_j})$ , and un.masks  $(\text{Seed}_{\text{mk}_{\mathcal{S}, j, \mathcal{R}, \mathcal{S}}}^{\mathbf{q}_j})$  to get  $\text{id}_{\mathbf{q}_j} \leftarrow \text{mask}_{\mathcal{S}, j}^{-1} \cdot (\text{Seed}_{\text{mk}_{\mathcal{S}, j, \mathcal{R}, \mathcal{S}}}^{\mathbf{q}_j})$ .
12. Symmetrically,  $\mathcal{R}$  gets  $(\text{id}_{\mathbf{w}_i})_{i \in [n]}$ .

**Fig. 8.** UFmap Protocol for  $L_\infty$  Distance Without Expansion:  $\Pi_{\text{SAS Fmap}}^{L_\infty}$

from  $\mathbf{q}'$  on dimension  $k$ . A set  $\mathbf{Q}$  is separated, if and only if, for each point in  $\mathbf{Q}$ , there exists a dimension such that this point is separated from any other points in  $\mathbf{Q}$  on it.

**Lemma 3 (Uniform Distribution [1]).** *In a  $d$ -dimensional space, if points in set  $\mathbf{Q}$  are uniformly distributed, then the probability that  $\mathbf{Q}$  is separated is  $1 - \text{negl}(d)$ .*

For SAS Fmap, we introduce the  $R \wedge S$ . disj. proj. assumption:

*Each Sender's or Receiver's point is separated from other points in the same set on at least one dimension.*

If points of inputs are uniformly distributed, then according to Lemma 3, this assumption holds in high-dimensional case with overwhelming probability, thus it is acceptable to base our construction on this assumption.

It is self-evident that the expansion rate of our protocol in Fig.8 is 1. Now, we prove our protocol is a semi-honest secure Fmap for  $L_\infty$  distance.

**Theorem 4 (Correctness).** *The protocol presented in Fig.8 satisfies the correctness defined in Definition 4 for  $L_\infty$  distance.*

*Proof.* For  $\mathbf{q}_j \in \mathbf{Q}$  and  $\mathbf{w}_i \in \mathbf{W}$ , if  $L_\infty(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ , then for  $k \in [d]$ ,  $|q_{j,k} - w_{i,k}| \leq \delta$  always holds, thus these  $q_{j,k}$  are all assigned in  $\text{List}_{\mathcal{R}}$ . According to correctness of OKVS,  $\mathcal{S}$  gets ElGamal ciphertexts of  $\text{List}_{\mathcal{R}}[k||q_{j,k}]$  by decoding.

Since Assignment algorithm ensures that the  $2\delta + 1$  points on dimension  $k$  centered at  $w_{i,k}$  all have the same assignment, it comes that  $\text{List}_{\mathcal{R}}[k||q_{j,k}] = \text{List}_{\mathcal{R}}[k||w_{i,k}]$  for  $k \in [d]$ . Therefore,  $\text{sum}_{\mathbf{q}_j}^{\text{pk}_{\text{EG}, \mathcal{R}}}$  is the ElGamal ciphertext of  $\text{Seed}_{\mathbf{w}_i, \mathcal{R}}$ . Then, it is clear that

$$\begin{aligned} \text{Seed}_{\text{mk}_{\mathcal{S}, j}, \mathcal{R}}^{\mathbf{q}_j} &= \text{sk}_{\text{DH}, \mathcal{R}} \cdot \text{mask}_{\mathcal{S}, j} \cdot (\text{Seed}_{\mathbf{q}_i, \mathcal{S}} + \text{Seed}_{\mathbf{w}_j, \mathcal{R}}) \\ \implies \text{id}_{\mathbf{q}_j} &= \text{sk}_{\text{DH}, \mathcal{S}} \cdot \text{sk}_{\text{DH}, \mathcal{R}} \cdot (\text{Seed}_{\mathbf{q}_i, \mathcal{S}} + \text{Seed}_{\mathbf{w}_j, \mathcal{R}}) \end{aligned}$$

Similarly, we can find that

$$\text{id}_{\mathbf{w}_i} = \text{sk}_{\text{DH}, \mathcal{R}} \cdot \text{sk}_{\text{DH}, \mathcal{S}} \cdot (\text{Seed}_{\mathbf{w}_j, \mathcal{R}} + \text{Seed}_{\mathbf{q}_i, \mathcal{S}})$$

Hence,  $\text{id}_{\mathbf{q}_j}$  equals  $\text{id}_{\mathbf{w}_i}$  when  $L_\infty(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$  holds.

**Theorem 5 (Distinctiveness).** *The protocol presented in Fig.8 satisfies the distinctiveness defined in Definition 4 if  $R \wedge S$ . disj. proj. assumption holds.*

*Proof.* First consider the side of  $\mathcal{S}$ . Since  $R \wedge S$ . disj. proj. assumption holds, for each  $j \in [m]$ , let us assume that  $\mathbf{q}_j$  is separated from the others in  $\mathbf{Q}$  on dimension  $k_j$ . With reference to proof of Theorem 4, we have

$$\text{id}_{\mathbf{q}_j} \triangleq \text{sk}_{\text{DH}, \mathcal{S}} \cdot \text{sk}_{\text{DH}, \mathcal{R}} \cdot (\text{List}_{\mathcal{S}}[k_j||q_{j,k_j}] + \Delta_j)$$

Under  $R \wedge S$ . disj. proj. assumption, in Assignment algorithm, the  $2\delta + 1$  points centered at  $q_{j,k_j}$  on dimension  $k_j$  do not cover any other assigned points

nor be covered by any other assigned points. Thus,  $\text{List}_{\mathcal{S}} [k_j \| q_{j,k_j}]$  is a uniformly random value independent of any other assignments. So, the probability that  $\text{id}_{\mathbf{q}_j}$  equals some  $\text{id}_{\mathbf{q}_{j'}}$  is  $\frac{m-1}{|\mathbb{G}|}$ . Hence, it holds that

$$\Pr[\exists j, j' \in [m], \text{s.t. } (j \neq j') \wedge (\text{ID}(\mathbf{q}_j) \cap \text{ID}(\mathbf{q}_{j'}) \neq \emptyset)] \leq \frac{m^2}{|\mathbb{G}|} = \text{negl}(\kappa)$$

Symmetrically, the same discussion for  $\mathcal{R}$  will complete the proof.

**Theorem 6 (Randomness).** *In the protocol presented in Fig.8,  $(\text{id}_{\mathbf{q}_j})_{j \in [m]}$  is computationally indistinguishable from  $\mathbb{R}_{\text{id}} \xleftarrow{\mathbb{R}} \mathbb{G}^m$ , and  $(\text{id}_{\mathbf{w}_i})_{i \in [n]}$  is computationally indistinguishable from  $\mathbb{R}_{\text{id}} \xleftarrow{\mathbb{R}} \mathbb{G}^n$ , if the DDH assumption holds.*

*Proof.* First consider the side of  $\mathcal{S}$ . With proof of Theorem 4, we have

$$\text{id}_{\mathbf{q}_j} = \text{sk}_{\text{DH},\mathcal{S}} \cdot (\text{sk}_{\text{DH},\mathcal{R}} \cdot \text{Seed}_{\mathbf{q}_j,\mathcal{S}}) + \text{sk}_{\text{DH},\mathcal{S}} \cdot \text{sk}_{\text{DH},\mathcal{R}} \cdot \text{Seed}_{\mathbf{q}_j,\mathcal{R}}$$

According to DDH assumption,  $(\text{sk}_{\text{DH},\mathcal{R}} \cdot \text{Seed}_{\mathbf{q}_j,\mathcal{S}})_{j \in [m]}$  is computationally indistinguishable from uniformly random vector in  $\mathbb{G}^m$ .

Since the assignment of  $\mathcal{S}$ 's coordinate system and that of  $\mathcal{R}$ 's coordinate system are independent,  $\text{Seed}_{\mathbf{q}_j,\mathcal{R}}$  is independent of  $\text{Seed}_{\mathbf{q}_j,\mathcal{S}}$ . In conclusion,  $(\text{id}_{\mathbf{q}_j})_{j \in [m]}$  is computationally indistinguishable from  $\mathbb{R}_{\text{id}} \xleftarrow{\mathbb{R}} \mathbb{G}^m$ .

Symmetrically, the same discussion for  $\mathcal{R}$  will complete the proof.

**Theorem 7 (Security).** *The protocol presented in Fig.8 satisfies the security defined in Definition 4 if the DDH assumption holds and  $\mathbb{R} \wedge \mathbb{S}$ . disj. proj. assumption holds.*

*Proof.* First consider the side of  $\mathcal{S}$ . We exhibit simulator  $\text{Sim}_{\text{Fmap}}^{\mathbb{S}}(\mathbf{Q}, \text{id}(\mathbf{Q})_{\mathbf{W}})$  for simulating corrupt  $\mathcal{S}$  where  $\text{id}(\mathbf{Q})_{\mathbf{W}}$  is the output of  $\mathcal{S}$  holding  $\mathbf{Q}$  who invokes the protocol presented in Fig.8 with  $\mathcal{R}$  holding  $\mathbf{W}$ . We first argue the indistinguishability of the produced transcript from the real execution. And then complete the security proof by combining Theorem 6.

$\text{Sim}_{\text{Fmap}}^{\mathbb{S}}$  simulating the view of corrupt semi-honest Sender executes as follows:

1.  $\text{Sim}_{\text{Fmap}}^{\mathbb{S}}$  generates  $(\overline{\text{sk}}_{\text{EIG},\mathcal{S}}, \overline{\text{pk}}_{\text{EIG},\mathcal{S}}) \leftarrow \text{Gen}^{\text{EIG}}(1^\kappa)$ , samples  $\left\{ \overline{\text{mask}}_{\mathcal{S},j} \xleftarrow{\mathbb{R}} \mathbb{F}_p \right\}_{j \in [m]}$ , computes  $\left\{ \overline{\text{mask}}_{\mathcal{S},j}^{-1} \right\}_{j \in [m]}$ , and appends them to the view.
2.  $\text{Sim}_{\text{Fmap}}^{\mathbb{S}}$  encodes OKVS  $\overline{E}_{\mathcal{R}}$  with  $(2\delta+1)dn$  dummy key-value pairs, generates  $(\overline{\text{sk}}_{\text{EIG},\mathcal{R}}, \overline{\text{pk}}_{\text{EIG},\mathcal{R}}) \leftarrow \text{Gen}^{\text{EIG}}(1^\kappa)$ , and appends  $\overline{E}_{\mathcal{R}}, \overline{\text{pk}}_{\text{EIG},\mathcal{R}}$  to the view.
3.  $\text{Sim}_{\text{Fmap}}^{\mathbb{S}}$  samples  $(\overline{\text{Seed}}_i \xleftarrow{\mathbb{R}} \mathbb{G})_{i \in [n]}$ , computes  $(\text{cipher}_i^{\overline{\text{pk}}_{\text{EIG},\mathcal{S}}} \leftarrow \text{Enc}_{\overline{\text{pk}}_{\text{EIG},\mathcal{S}}}^{\text{EIG}}(\overline{\text{Seed}}_i))_{i \in [n]}$ , and appends  $(\text{cipher}_i^{\overline{\text{pk}}_{\text{EIG},\mathcal{S}}})_{i \in [n]}$  to the view.
4.  $\text{Sim}_{\text{Fmap}}^{\mathbb{S}}$  samples  $\overline{\text{sk}}_{\text{DH},\mathcal{S}} \xleftarrow{\mathbb{R}} \mathbb{F}_p$  and appends it to the view.



5.  $\text{Sim}_{\text{Fmap}}^{\mathcal{S}}$  computes  $\left(\overline{\text{Seed}_{\text{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}} \leftarrow \overline{\text{mask}_{\mathcal{S},j}} \cdot \overline{\text{sk}_{\text{DH},\mathcal{S}}}^{-1} \cdot \text{id}(\mathbf{q}_j)_{\mathbf{W}}\right)_{j \in [m]}$  and appends them to the view.

Now we show that the view output by  $\text{Sim}_{\text{Fmap}}^{\mathcal{S}}$  is indistinguishable from the real one via a hybrid argument. We define four hybrid transcripts  $T_0, T_1, T_2, T_3$ , where  $T_0$  is the real view of  $\mathcal{S}$ , and  $T_3$  is the output of  $\text{Sim}_{\text{Fmap}}^{\mathcal{S}}$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.8. Let  $T_0$  denote  $\mathcal{S}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that OKVS  $E_{\mathcal{R}}$  and  $\text{pk}_{\text{ElG},\mathcal{R}}$  are replaced by  $\overline{E_{\mathcal{R}}}$  and  $\overline{\text{pk}_{\text{ElG},\mathcal{R}}}$ . The values for encoding  $E_{\mathcal{R}}$  are  $(2\delta+1)dn$  ciphertexts encrypted with  $\text{pk}_{\text{ElG},\mathcal{R}}$ , which are computationally indistinguishable from uniformly random ElGamal ciphertexts by DDH assumption. Combining the obliviousness of OKVS,  $E_{\mathcal{R}}$  and  $\overline{E_{\mathcal{R}}}$  are computationally indistinguishable. Hence, we have  $T_1 \stackrel{c}{\approx} T_0$ .
- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that  $(\text{sk}_{\text{ElG},\mathcal{S}}, \text{pk}_{\text{ElG},\mathcal{S}})$  and  $\left(\text{cipher}_{\mathbf{w}_i}^{\text{pk}_{\text{ElG},\mathcal{S}}}\right)_{i \in [n]}$  are replaced by  $(\overline{\text{sk}_{\text{ElG},\mathcal{S}}}, \overline{\text{pk}_{\text{ElG},\mathcal{S}}})$  and  $\left(\text{cipher}_i^{\overline{\text{pk}_{\text{ElG},\mathcal{S}}}}\right)_{i \in [n]}$ . Since each  $\text{Seed}_{\text{mk}_{\mathcal{R},i}^{\mathbf{w}_i}}$  is masked by uniformly random  $\text{mask}_{\mathcal{R},i}$  from  $\mathcal{R}$ ,  $\left(\text{Seed}_{\text{mk}_{\mathcal{R},i}^{\mathbf{w}_i}}\right)_{i \in [n]}$  are statistically indistinguishable from  $\left(\overline{\text{Seed}_i} \leftarrow^{\text{R}} \mathbb{G}\right)_{i \in [n]}$ . Therefore,  $\left(\text{cipher}_{\mathbf{w}_i}^{\text{pk}_{\text{ElG},\mathcal{S}}}\right)_{i \in [n]}$  and  $\left(\text{cipher}_i^{\overline{\text{pk}_{\text{ElG},\mathcal{S}}}}\right)_{i \in [n]}$  are statistically indistinguishable, which means  $T_2 \stackrel{s}{\approx} T_1$ .
- **Hyb<sub>3</sub>**. Let  $T_3$  be the same as  $T_2$ , except that  $\{\text{mask}_{\mathcal{S},j}\}_{j \in [m]}$ ,  $\{\text{mask}_{\mathcal{S},j}^{-1}\}_{j \in [m]}$ ,  $\text{sk}_{\text{DH},\mathcal{S}}$ , and  $\left(\text{Seed}_{\text{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}\right)_{j \in [m]}$  are replaced by  $\{\overline{\text{mask}_{\mathcal{S},j}}\}_{j \in [m]}$ ,  $\{\overline{\text{mask}_{\mathcal{S},j}^{-1}}\}_{j \in [m]}$ ,  $\overline{\text{sk}_{\text{DH},\mathcal{S}}}$ , and  $\left(\overline{\text{Seed}_{\text{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}}\right)_{j \in [m]}$ . It is clear that masks  $\{\text{mask}_{\mathcal{S},j}\}_{j \in [m]}$  and  $\{\overline{\text{mask}_{\mathcal{S},j}}\}_{j \in [m]}$  are distributed identically;  $\text{sk}_{\text{DH},\mathcal{S}}$  and  $\overline{\text{sk}_{\text{DH},\mathcal{S}}}$  are distributed identically. Hence,  $\left(\text{Seed}_{\text{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}\right)_{j \in [m]}$  and  $\left(\overline{\text{Seed}_{\text{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}}\right)_{j \in [m]}$  are statistically indistinguishable. Thus  $T_3 \stackrel{s}{\approx} T_2$  holds.

From the argument above, it holds that

$$\text{view}_{\mathcal{S}}^{\text{ISAS Fmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}) \stackrel{c}{\approx} \text{Sim}_{\text{Fmap}}^{\mathcal{S}}(\mathbf{Q}, \text{id}(\mathbf{Q})_{\mathbf{W}})$$

In addition, according to Theorem 6, we have

$$\text{Sim}_{\text{Fmap}}^{\mathcal{S}}(\mathbf{Q}, \text{id}(\mathbf{Q})_{\mathbf{W}}) \stackrel{c}{\approx} \text{Sim}_{\text{Fmap}}^{\mathcal{S}}\left(\mathbf{Q}, \text{R}_{\text{id}} \leftarrow^{\text{R}} \mathbb{G}^m\right)$$

Therefore, it comes that

$$\text{view}_{\mathcal{S}}^{\text{ISAS Fmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}) \stackrel{c}{\approx} \text{Sim}_{\text{Fmap}}^{\mathcal{S}}\left(\mathbf{Q}, \text{R}_{\text{id}} \leftarrow^{\text{R}} \mathbb{G}^m\right) \stackrel{c}{\approx} \text{view}_{\mathcal{S}}^{\text{ISAS Fmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W}')$$

Symmetrically, the same discussion for  $\mathcal{R}$  will complete the proof.

*Remark 3 (Complexity).* The protocol presented in Fig.8 has communication complexity  $\mathcal{O}(((2\delta + 1)d\kappa + 2\kappa + \kappa)(m + n))$ , computation complexity  $\mathcal{O}((2\delta + 1)dm + n)$  for  $\mathcal{S}$ , and computation complexity  $\mathcal{O}((2\delta + 1)dn + m)$  for  $\mathcal{R}$ , if the OKVS has a constant rate, linear encoding time, and constant decoding time.

*Remark 4 (Limitation of SAS Fmap).* For the sake of distinctiveness and security, SAS Fmap introduces assumptions about input for both two parties, namely  $\mathcal{R} \wedge \mathcal{S}$ . disj. proj. assumption. As a result, the application scope of FPSI(-variants) based on SAS Fmap is limited.

## 6 Multi-Query Fuzzy RPMT Based on sUFmap

In this section, we provide the ideal functionality for mqFRPMT and present mqFRPMT protocols for  $L_\infty$  distance and for  $L_{p \in [1, \infty)}$  distance respectively, using sUFmap for  $L_\infty$  distance.

### 6.1 Definition of mqFRPMT

mqFRPMT is the fuzzy version of mqRPMT, and we define the ideal functionality for mqFRPMT in Fig.9. Combining with OT, mqFRPMT can directly yield FPSI, FPSI-card, and LFPSI.

PARAMETERS: Sender  $\mathcal{S}$ , Receiver  $\mathcal{R}$ ; Set size  $m, n$ ; Dimension  $d$ ; Distance function  $\text{dist}(\cdot, \cdot)$ ; Distance threshold  $\delta$ .

FUNCTIONALITY:

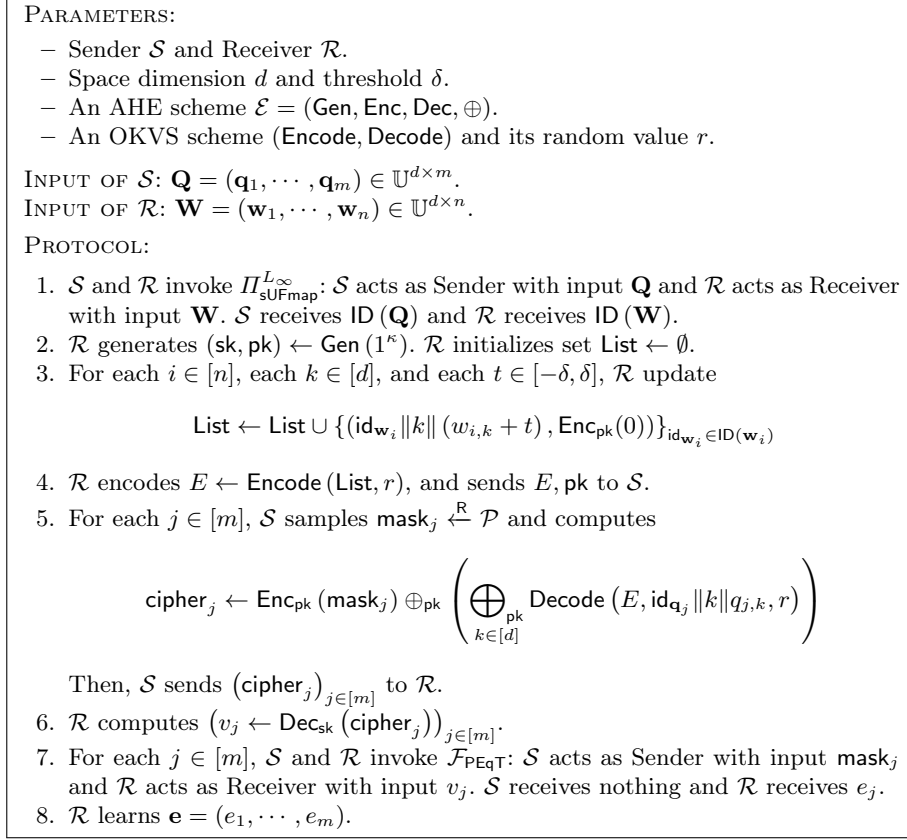
- Wait an input  $\mathbf{Q} \in \mathbb{U}^{d \times m}$  from  $\mathcal{S}$ .
- Wait an input  $\mathbf{W} \in \mathbb{U}^{d \times n}$  from  $\mathcal{R}$ .
- Return  $\mathbf{e} = (e_1, \dots, e_m) \in \{0, 1\}^m$  to  $\mathcal{R}$ , where  $e_j = 1$  if and only if there exists  $\mathbf{w}_i \in \mathbf{W}$  such that  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i) \leq \delta$ .

**Fig. 9.** Ideal Functionality for Multi-Query Fuzzy RPMT  $\mathcal{F}_{\text{mqFRPMT}}$

### 6.2 mqFRPMT for $L_\infty$ Distance from sUFmap

The high-level idea of sUFmap-based mqFRPMT is as described in Section 2.5. In mqFRPMT for  $L_\infty$  distance, we instantiate FMatch with the idea similar to [1]. We give the detailed mqFRPMT protocol for  $L_\infty$  distance in Fig.10.

**Theorem 8 (Correctness).** *The protocol presented in Fig.10 realizes the functionality  $\mathcal{F}_{\text{mqFRPMT}}$  defined in Fig.9 for  $L_\infty$  distance correctly.*



**Fig. 10.** mqFRPMT for  $L_\infty$  from sUFmap:  $\Pi_{\text{mqFRPMT}}^{L_\infty}$

*Proof.* First of all, because of the distinctiveness property of Fmap, keys of List for encoding OKVS  $E$  are all distinct from each other. Thus, the probability of encoding failure is negligible, thereby ensuring that the protocol can be executed completely with overwhelming probability.

For each  $\mathbf{q}_j \in \mathbf{Q}$ , there are two cases.

- **Case 1: There exists  $\mathbf{w}_* \in \mathbf{W}$  satisfying  $L_\infty(\mathbf{q}_j, \mathbf{w}_*) \leq \delta$**   
 According to the correctness of sUFmap, there exists  $\text{id}_{\mathbf{w}_*} \in \text{ID}(\mathbf{w}_*)$  such that  $\text{id}_{\mathbf{q}_j} = \text{id}_{\mathbf{w}_*}$  holds. And note that  $|q_{j,k} - w_{*,k}| \leq \delta$  holds for all  $k \in [d]$ , which means  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \| k \| q_{j,k}, r) = \text{Enc}_{\text{pk}}(0)$  holds for all  $k \in [d]$ . Hence,  $\text{cipher}_j$  is AHE ciphertext of  $\text{mask}_j$ . Thus,  $e_j = 1$  comes from the correctness of PEqT.
- **Case 2: For any  $\mathbf{w}_i \in \mathbf{W}$ , there is  $L_\infty(\mathbf{q}_j, \mathbf{w}_i) > \delta$** 
  - If  $\text{ID}(\mathbf{q}_j)$  does not intersect with anyone of  $\text{ID}(\mathbf{W})$ , according to the independence of OKVS,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \| k \| q_{j,k}, r)$  is always statistically

indistinguishable to uniformly random value. Therefore, the probability that  $\text{cipher}_j$  is an AHE ciphertext of  $\text{mask}_j$  is negligible.

- If there is  $\mathbf{w}_* \in \mathbf{W}$  such that  $\text{ID}(\mathbf{q}_j)$  intersects with  $\text{ID}(\mathbf{w}_*)$ , there exists  $k_* \in [d]$  satisfying  $|q_{j,k_*} - w_{*,i,k_*}| > \delta$ . So, according to the independence of OKVS,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \| k_* \| q_{j,k_*}, r)$  is statistically indistinguishable to uniformly random value. Hence, the probability that  $\text{cipher}_j$  is an AHE ciphertext of  $\text{mask}_j$  is negligible.

To sum up,  $e_j = 0$  holds in this case.

**Theorem 9 (Security).** *The protocol presented in Fig. 10 realizes the functionality  $\mathcal{F}_{\text{mqFRPMT}}$  defined in Fig. 9 for  $L_\infty$  distance against semi-honest adversaries in the  $\mathcal{F}_{\text{PEqT}}$ -hybrid model if  $\mathcal{E}$  satisfies IND-CPA security.*

*Proof.* We exhibit simulators  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}(\mathbf{Q})$  and  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}(\mathbf{W}, \mathbf{e})$  for simulating corrupt  $\mathcal{S}$  and corrupt  $\mathcal{R}$  respectively. And we argue the indistinguishability of the produced transcript from the real execution.

Corrupt Sender:  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  simulating the view of corrupt semi-honest Sender executes as follows:

1.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  samples  $\overline{\mathbf{W}} \xleftarrow{\mathcal{R}} \mathbb{U}^{d \times n}$ . Then, it executes sUFmap where Sender holds  $\mathbf{Q}$  and Receiver holds  $\overline{\mathbf{W}}$  to get Sender's view  $\text{view}_{\mathcal{S}}^{H_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  and appends  $\text{view}_{\mathcal{S}}^{H_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  to the view.
2.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  encodes OKVS  $\overline{E}$  with  $(2\delta + 1)dn \cdot \text{rate}_{\mathcal{R}}$  dummy key-value pairs, generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends  $\overline{E}, \overline{\text{pk}}$  to the view.
3.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  samples  $\{\overline{\text{mask}}_j \xleftarrow{\mathcal{R}} \mathbb{U}\}_{j \in [m]}$  and appends them to the view.
4. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  invokes PEqT simulator  $\text{Sim}_{\text{PEqT}}^{\mathcal{S}}(\overline{\text{mask}}_j)$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  is indistinguishable from the real one via a hybrid argument. We define four hybrid transcripts  $T_0, T_1, T_2, T_3$ , where  $T_0$  is the real view of  $\mathcal{S}$ , and  $T_3$  is the output of  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$ .

- **Hyb<sub>0</sub>.** This hybrid is the real interaction described in Fig. 10. Let  $T_0$  denote  $\mathcal{S}$ 's view in the real protocol.
- **Hyb<sub>1</sub>.** Let  $T_1$  be the same as  $T_0$ , except that the sUFmap execution is replaced by  $\text{view}_{\mathcal{S}}^{H_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{S}}^{H_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}}) \stackrel{c}{\approx} \text{view}_{\mathcal{S}}^{H_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.
- **Hyb<sub>2</sub>.** Let  $T_2$  be the same as  $T_1$ , except that OKVS  $E$  and  $\text{pk}$  are replaced by  $\overline{E}$  and  $\overline{\text{pk}}$ . The indistinguishability follows from the IND-CPA security of  $\mathcal{E}$  and the obliviousness of OKVS. Therefore, we have  $T_2 \stackrel{c}{\approx} T_1$ .
- **Hyb<sub>3</sub>.** Let  $T_3$  be the same as  $T_2$ , except that  $\{\text{mask}_j\}_{j \in [m]}$  and the PEqT execution are replaced by  $\{\overline{\text{mask}}_j\}_{j \in [m]}$  and simulators  $\{\text{Sim}_{\text{PEqT}}^{\mathcal{S}}(\overline{\text{mask}}_j)\}_{j \in [m]}$ . Since  $\{\text{mask}_j\}_{j \in [m]}$  and  $\{\overline{\text{mask}}_j\}_{j \in [m]}$  are distributed identically, the semi-honest security of PEqT functionality guarantees  $T_3 \stackrel{c}{\approx} T_2$ .

Clearly,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$ 's simulated view is identical to  $\mathcal{S}$ 's view in the real execution.

Corrupt Receiver:  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  simulating the view of corrupt semi-honest Receiver executes as follows:

1.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  samples  $\overline{\mathbf{Q}} \xleftarrow{\mathcal{R}} \mathbb{U}^{d \times m}$ . Then, it executes sUFmap where Sender holds  $\mathbf{Q}$  and Receiver holds  $\mathbf{W}$  to get Receiver's view  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  and appends  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  to the view.
2.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends them to the view.
3. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  generates a uniformly random AHE ciphertext as  $\overline{\text{cipher}}_j$ .  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  appends  $(\overline{\text{cipher}}_j)_{j \in [m]}$  to the view.
4. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  invokes PEqT simulator  $\text{Sim}_{\text{PEqT}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_j), e_j)$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  is indistinguishable from the real one via a hybrid argument. We define three hybrid transcripts  $T_0, T_1, T_2$ , where  $T_0$  is the real view of  $\mathcal{R}$ , and  $T_2$  is the output of  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.10. Let  $T_0$  denote  $\mathcal{R}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that the Fmap execution is replaced by  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W}) \stackrel{c}{\approx} \text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.
- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that  $(\overline{\text{sk}}, \overline{\text{pk}})$ ,  $(\overline{\text{cipher}}_j)_{j \in [m]}$ , and the PEqT execution are replaced by  $(\overline{\text{sk}}, \overline{\text{pk}})$ ,  $(\overline{\text{cipher}}_j)_{j \in [m]}$ , and simulators  $\text{Sim}_{\text{PEqT}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_j), e_j)$ . Since each  $\text{cipher}_j$  has a uniformly random mask  $\text{mask}_j$  from  $\mathcal{S}$ , we have  $(\overline{\text{cipher}}_j)_{j \in [m]} \stackrel{s}{\approx} (\text{cipher}_j)_{j \in [m]}$ . Then, the semi-honest security of PEqT functionality guarantees that simulators  $\left\{ \text{Sim}_{\text{PEqT}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_j), e_j) \right\}_{j \in [m]}$  and the PEqT execution are computationally indistinguishable, which means  $T_2 \stackrel{c}{\approx} T_1$ .

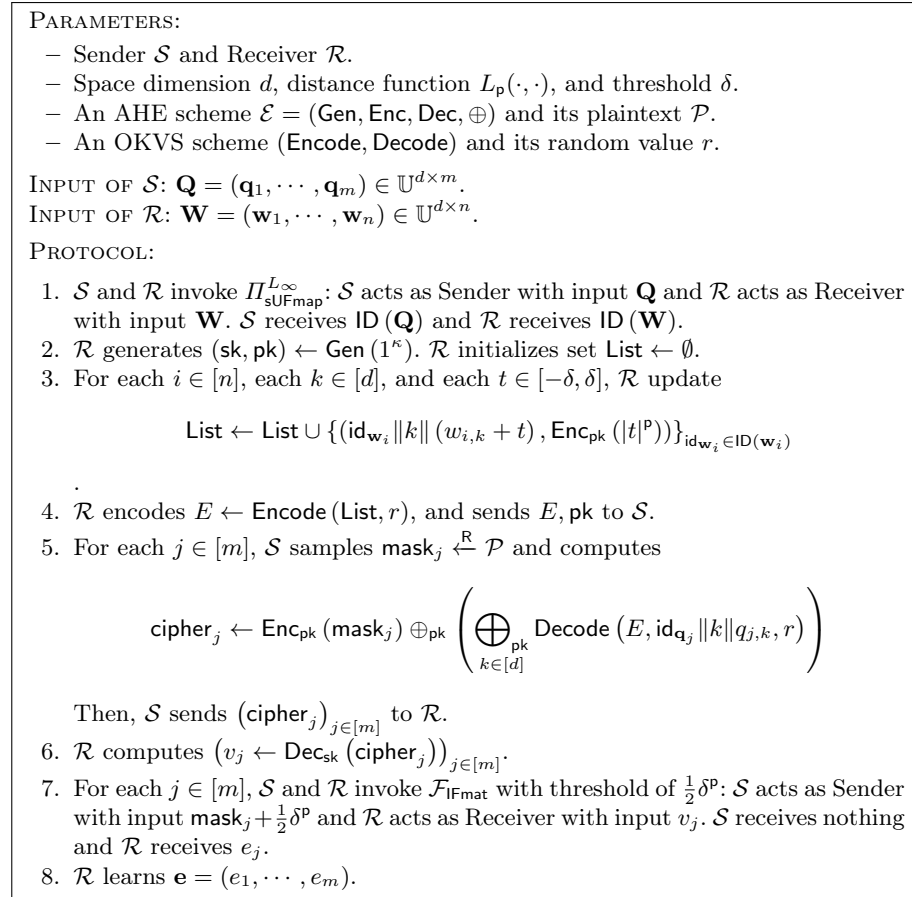
Clearly,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$ 's simulated view is identical to  $\mathcal{R}$ 's view in the real execution.

*Remark 5 (Complexity).* The protocol presented in Fig.10 has communication complexity  $\mathcal{O}(((2\delta + 1)d\kappa + 2\kappa + \kappa)(m + n))$ , computation complexity  $\mathcal{O}((2\delta + 1)dm + n)$  for  $\mathcal{S}$ , and computation complexity  $\mathcal{O}((2\delta + 1)dn + m)$  for  $\mathcal{R}$ , if the OKVS has a constant rate, linear encoding time, and constant decoding time; the UFmap is SAS Fmap in Fig.8.

### 6.3 mqFRPMT for $L_p \in [1, \infty)$ Distance from sUFmap

The construction of mqFRPMT for  $L_{p \in [1, \infty)}$  distance is similar to  $\Pi_{\text{mqFRPMT}}^{L_\infty}$ . For computing  $L_{p \in [1, \infty)}$  distance, in OKVS encoding, AHE ciphertexts of  $|t|^p$  instead of 0 are used as values of  $\text{id}_{\mathbf{w}_i} \parallel (w_{i,k} + t)$ . Therefore, Sender can compute AHE ciphertexts of the  $p$  power of distances and mask them. With IFmat protocol, Receiver can complete the secure comparison between masked  $p$  power of distances and masked  $\delta^p$  with Sender to learn final result of mqFRPMT.

We give the detailed mqFRPMT protocol for  $L_{p \in [1, \infty)}$  distance and relevant proofs in Fig.11.



**Fig. 11.** mqFRPMT for  $L_{p \in [1, \infty)}$  from sUFmap:  $\Pi_{\text{mqFRPMT}}^{L_{p \in [1, \infty)}}$

**Theorem 10 (Correctness).** *The protocol presented in Fig.11 realizes the functionality  $\mathcal{F}_{\text{mqFRPMT}}$  defined in Fig.9 for  $L_{p \in [1, \infty)}$  distance correctly.*

*Proof.* This proof is similar to the proof of Theorem 8.

First of all, because of the distinctiveness property of Fmap, keys of List for encoding OKVS  $E$  are all distinct from each other. Thus, the probability of encoding failure is negligible, thereby ensuring that the protocol can be executed completely with overwhelming probability.

According to Corollary 2,  $\Pi_{\text{sUFmap}}^{L_\infty}$  can be seen as  $\Pi_{\text{sUFmap}}^{L_{p \in [1, \infty)}}$ .

For each  $\mathbf{q}_j \in \mathbf{Q}$ , there are two cases.

- **Case 1: There exists  $\mathbf{w}_* \in \mathbf{W}$  satisfying  $L_p(\mathbf{q}_j, \mathbf{w}_*) \leq \delta$**   
According to the correctness of sUFmap,  $\text{id}_{\mathbf{q}_j} = \text{id}_{\mathbf{w}_*}$  holds for an  $\text{id}_{\mathbf{w}_*} \in \text{ID}(\mathbf{w}_*)$ . And note that  $|q_{j,k} - w_{*,k}| \leq \delta$  holds for all  $k \in [d]$ , which means  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \|k\|q_{j,k}, r) = \text{Enc}_{\text{pk}}(|q_{j,k} - w_{*,k}|^p)$  holds for all  $k \in [d]$ . Hence,  $\text{cipher}_j$  is AHE ciphertext of  $\text{mask}_j + L_p(\mathbf{q}_j, \mathbf{w}_*)^p$ , which is within interval  $[\text{mask}_j, \text{mask}_j + \delta^p]$ . Thus,  $e_j = 1$  comes from the correctness of IFmat.
- **Case 2: For any  $\mathbf{w}_i \in \mathbf{W}$ , there is  $L_p(\mathbf{q}_j, \mathbf{w}_i) > \delta$** 
  - If  $\text{ID}(\mathbf{q}_j)$  does not intersect with anyone of  $\text{ID}(\mathbf{W})$ , according to the independence of OKVS,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \|k\|q_{j,k}, r)$  is always statistically indistinguishable to uniformly random value. Therefore, the probability that  $\mathcal{R}$  gets 1 from IFmat is negligible.
  - If there is  $\mathbf{w}_* \in \mathbf{W}$  such that  $\text{ID}(\mathbf{q}_j)$  intersects with  $\text{ID}(\mathbf{w}_*)$ , there exists  $k_* \in [d]$  satisfying  $|q_{j,k_*} - w_{*,k_*}| > \delta$ . So, according to the independence of OKVS,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j} \|k_*\|q_{j,k_*}, r)$  is statistically indistinguishable to uniformly random value. Hence, the probability that  $\mathcal{R}$  gets 1 from IFmat is negligible.

To sum up,  $e_j = 0$  holds in this case.

**Theorem 11 (Security).** *The protocol presented in Fig.11 realizes the functionality  $\mathcal{F}_{\text{mqFRPMT}}$  defined in Fig.9 for  $L_{p \in [1, \infty)}$  distance against semi-honest adversaries in the  $\mathcal{F}_{\text{IFmat}}$ -hybrid model if  $\mathcal{E}$  satisfies IND-CPA security.*

*Proof.* We exhibit simulators  $\text{Sim}_{\text{mqFRPMT}}^S(\mathbf{Q})$  and  $\text{Sim}_{\text{mqFRPMT}}^R(\mathbf{W}, \mathbf{e})$  for simulating corrupt  $\mathcal{S}$  and corrupt  $\mathcal{R}$  respectively. And we argue the indistinguishability of the produced transcript from the real execution.

Corrupt Sender:  $\text{Sim}_{\text{mqFRPMT}}^S$  simulating the view of corrupt semi-honest Sender executes as follows:

1.  $\text{Sim}_{\text{mqFRPMT}}^S$  samples  $\overline{\mathbf{W}} \xleftarrow{R} \mathbb{U}^{d \times n}$ . Then, it executes sUFmap where Sender holds  $\mathbf{Q}$  and Receiver holds  $\overline{\mathbf{W}}$  to get Sender's view  $\text{view}_{\mathcal{S}}^{\Pi_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  and appends  $\text{view}_{\mathcal{S}}^{\Pi_{\text{sUFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  to the view.
2.  $\text{Sim}_{\text{mqFRPMT}}^S$  encodes OKVS  $\overline{E}$  with  $(2\delta + 1)dn \cdot \text{rate}_{\mathcal{R}}$  dummy key-value pairs, generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends  $\overline{E}, \overline{\text{pk}}$  to the view.
3.  $\text{Sim}_{\text{mqFRPMT}}^S$  samples  $\{\overline{\text{mask}}_j \xleftarrow{R} \mathbb{U}\}_{j \in [m]}$  and appends them to the view.
4. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^S$  invokes IFmat simulator  $\text{Sim}_{\text{IFmat}}^S(\overline{\text{mask}}_j + \frac{1}{2}\delta^p)$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$  is indistinguishable from the real one via a hybrid argument. We define four hybrid transcripts  $T_0, T_1, T_2, T_3$ , where  $T_0$  is the real view of  $\mathcal{S}$ , and  $T_3$  is the output of  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.11. Let  $T_0$  denote  $\mathcal{S}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that the sUFmap execution is replaced by  $\text{view}_{\mathcal{S}}^{\text{UFmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{S}}^{\text{UFmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}}) \stackrel{c}{\approx} \text{view}_{\mathcal{S}}^{\text{UFmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.
- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that OKVS  $E$  and  $\text{pk}$  are replaced by  $\overline{E}$  and  $\overline{\text{pk}}$ . The indistinguishability follows from the IND-CPA security of  $\mathcal{E}$  and the obliviousness of OKVS. Therefore, we have  $T_2 \stackrel{c}{\approx} T_1$ .
- **Hyb<sub>3</sub>**. Let  $T_3$  be the same as  $T_2$ , except that  $\{\text{mask}_j\}_{j \in [m]}$  and the IFmat execution are replaced by  $\{\overline{\text{mask}}_j\}_{j \in [m]}$  and simulators  $\left\{ \text{Sim}_{\text{IFmat}}^{\mathcal{S}}(\overline{\text{mask}}_j + \frac{1}{2}\delta^{\text{p}}) \right\}_{j \in [m]}$ . Since  $\{\text{mask}_j\}_{j \in [m]}$  and  $\{\overline{\text{mask}}_j\}_{j \in [m]}$  are distributed identically, the semi-honest security of IFmat functionality guarantees  $T_3 \stackrel{c}{\approx} T_2$ .

Clearly,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}$ 's simulated view is identical to  $\mathcal{S}$ 's view in the real execution.

Corrupt Receiver:  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  simulating the view of corrupt semi-honest Receiver executes as follows:

1.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  samples  $\overline{\mathbf{Q}} \leftarrow^{\mathcal{R}} \mathbb{U}^{d \times m}$ . Then, it executes sUFmap where Sender holds  $\overline{\mathbf{Q}}$  and Receiver holds  $\mathbf{W}$  to get Receiver's view  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  and appends  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  to the view.
2.  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends them to the view.
3. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  generates a uniformly random AHE ciphertext as  $\overline{\text{cipher}}_j$ .  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  appends  $(\overline{\text{cipher}}_j)_{j \in [m]}$  to the view.
4. For each  $j \in [m]$ ,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  invokes IFmat simulator  $\text{Sim}_{\text{IFmat}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_j), e_j)$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$  is indistinguishable from the real one via a hybrid argument. We define three hybrid transcripts  $T_0, T_1, T_2$ , where  $T_0$  is the real view of  $\mathcal{R}$ , and  $T_2$  is the output of  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.11. Let  $T_0$  denote  $\mathcal{R}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that the sUFmap execution is replaced by  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W}) \stackrel{c}{\approx} \text{view}_{\mathcal{R}}^{\text{UFmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.



- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that  $(\text{sk}, \text{pk})$ ,  $(\text{cipher}_j)_{j \in [m]}$ , and the IFmat execution are replaced by  $(\overline{\text{sk}}, \overline{\text{pk}})$ ,  $(\overline{\text{cipher}_j})_{j \in [m]}$ , and simulators  $\text{Sim}_{\text{IFmat}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}_j}), e_j)$ . Since each  $\text{cipher}_j$  has a uniformly random mask  $\text{mask}_j$  from  $\mathcal{S}$ , we have  $(\overline{\text{cipher}_j})_{j \in [m]} \stackrel{s}{\approx} (\text{cipher}_j)_{j \in [m]}$ . Then, the semi-honest security of IFmat functionality guarantees that simulators  $\left\{ \text{Sim}_{\text{IFmat}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}_j}), e_j) \right\}_{j \in [m]}$  and the IFmat execution are computationally indistinguishable, which means  $T_2 \stackrel{c}{\approx} T_1$ .

Clearly,  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}$ 's simulated view is identical to  $\mathcal{R}$ 's view in the real execution.

*Remark 6 (Complexity).* The protocol presented in Fig.11 has communication complexity  $\mathcal{O}(((2\delta + 1)d\kappa + 2\kappa + \kappa)(m + n) + mp \log \delta \kappa)$ , computational complexity  $\mathcal{O}((2\delta + 1)dm + n + mp \log \delta)$  for  $\mathcal{S}$ , and computational complexity  $\mathcal{O}((2\delta + 1)dn + m + mp \log \delta)$  for  $\mathcal{R}$ , if the OKVS has a constant rate, linear encoding time, and constant decoding time; the UFmap is SAS Fmap in Fig.8; the IFmat has communication and computation costs scale logarithmically in the threshold.

## 7 FPSI Protocols

### 7.1 Generic Construction of FPSI from Fmap

Fmap generates the same identifier for Sender's point and Receiver's point that are close to each other, and then Sender and Receiver can use OKVS and ssF-Match to further filter the point pairs implied by these identifiers to obtain the FPSI output. This is a generic approach to constructing FPSI from Fmap, indicating the adaptability of Fmap for various distance functions.

**FPSI for Distances with Translation Invariance.** As a specific example, let us now focus on constructing FPSI from Fmap for those distance functions having the translation invariance property.

**Definition 10 (Translation Invariance).** A distance function  $\text{dist}(\cdot, \cdot)$  on  $\mathbb{U}^d \times \mathbb{U}^d$  has translation invariance property if and only if, for any two point  $\mathbf{q}, \mathbf{w} \in \mathbb{U}^d$  and any vector  $\mathbf{v} \in \mathbb{U}^d$ , it holds that

$$\text{dist}(\mathbf{q}, \mathbf{w}) = \text{dist}(\mathbf{q} + \mathbf{v}, \mathbf{w} + \mathbf{v})$$

It is not difficult to see that Hamming and  $L_{p \in [1, \infty]}$  distances both have translation invariance property. We provide the detailed generic construction from Fmap to FPSI for distance with translation invariance in Fig.12. Thus,

this generic construction is a powerful tool for FPSI for Hamming and  $L_{p \in [1, \infty]}$  distances. Specifically, we can instantiate the Fmap and ssFMatch in the construction in Fig.12 with UniqC Fmap and trivial ssFMatch for Hamming distance in Section 3.5 to obtain an FPSI for Hamming distance.

**Theorem 12 (Correctness).** *The protocol presented in Fig.12 realizes the functionality  $\mathcal{F}_{\text{FPSI}}$  defined in Fig.1 for distance with translation invariance correctly.*

*Proof.* First of all, because of the distinctiveness property of Fmap, keys of List for encoding OKVS  $E$  are all distinct from each other. Thus, the probability of encoding failure is negligible, thereby ensuring that the protocol can be executed completely with overwhelming probability.

For each  $\mathbf{q}_j \in \mathbf{Q}$ , there are two cases.

- **Case 1: There exists  $\mathbf{w}_* \in \mathbf{W}$  satisfying  $\text{dist}(\mathbf{q}_j, \mathbf{w}_*) \leq \delta$**   
According to the correctness of Fmap, there exist  $\ell_0 \in [\text{rate}_{\mathcal{S}}], \ell_1 \in [\text{rate}_{\mathcal{R}}]$  such that  $\text{id}_{\mathbf{q}_j, \ell_0} = \text{id}_{\mathbf{w}_*, \ell_1}$  holds. Hence,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j, \ell_0}, r)$  is the AHE ciphertext of  $\mathbf{w}_*$ . Thus we have  $\mathbf{u}_{j, \ell_0} = \text{mask}_{j, \ell_0} + \mathbf{q}_j$  and  $\mathbf{v}_{j, \ell_0} = \text{mask}_{j, \ell_0} + \mathbf{w}_*$  from homomorphism. Because of  $\text{dist}(\cdot, \cdot)$ 's translation invariance and the correctness of ssFMatch,  $e_{\mathcal{S}, j, \ell_0} \oplus e_{\mathcal{R}, j, \ell_0}$  equals to 1, which means  $(e_{\mathcal{S}, j, \ell})_{\ell \in [\text{rate}_{\mathcal{S}}]} \neq (e_{\mathcal{R}, j, \ell})_{\ell \in [\text{rate}_{\mathcal{S}}]}$ . Therefore,  $e_j = 1$  because of the correctness of PEqT, thus  $\mathbf{q}_j \in \mathbf{I}_{\text{fuzzy}}$  holds.
- **Case 2: For any  $\mathbf{w}_i \in \mathbf{W}$ , there is  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i) > \delta$** 
  - If  $\text{ID}(\mathbf{q}_j)$  does not intersect with anyone of  $\text{ID}(\mathbf{W})$ , according to the independence of OKVS,  $\text{Decode}(E, \text{id}_{\mathbf{q}_j, \ell}, r)$  is always statistically indistinguishable to uniformly random value. Therefore, the probability that  $\mathcal{R}$  gets  $\mathbf{q}_j$  is negligible.
  - If there is  $\mathbf{w}_* \in \mathbf{W}$  such that  $\text{ID}(\mathbf{q}_j)$  intersects with  $\text{ID}(\mathbf{w}_*)$ , from the same discussion as in case 1, it can be known that  $\mathbf{u}_{j, \ell_0} = \text{mask}_{j, \ell_0} + \mathbf{q}_j$  and  $\mathbf{v}_{j, \ell_0} = \text{mask}_{j, \ell_0} + \mathbf{w}_*$ . According to  $\text{dist}(\cdot, \cdot)$ 's translation invariance and the correctness of ssFMatch and PEqT,  $\mathcal{R}$  can not get  $\mathbf{q}_j$ .

To sum up,  $\mathbf{q}_j \notin \mathbf{I}_{\text{fuzzy}}$  holds in this case.

**Theorem 13 (Security).** *The protocol presented in Fig.12 realizes the functionality  $\mathcal{F}_{\text{FPSI}}$  defined in Fig.1 for distance with translation invariance against semi-honest adversaries in the  $(\mathcal{F}_{\text{ssFMatch}}, \mathcal{F}_{\text{PEqT}}, \mathcal{F}_{\text{OT}})$ -hybrid model, if  $\mathcal{E}$  satisfies IND-CPA security.*

*Proof.* We exhibit simulators  $\text{Sim}_{\text{FPSI}}^{\mathcal{S}}(\mathbf{Q})$  and  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}(\mathbf{W}, \mathbf{I}_{\text{fuzzy}})$  for simulating corrupt  $\mathcal{S}$  and corrupt  $\mathcal{R}$  respectively. And we argue the indistinguishability of the produced transcript from the real execution.

Corrupt Sender:  $\text{Sim}_{\text{FPSI}}^{\mathcal{S}}$  simulating the view of corrupt semi-honest Sender executes as follows:

1.  $\text{Sim}_{\text{FPSI}}^{\mathcal{S}}$  samples  $\overline{\mathbf{W}} \xleftarrow{\mathcal{R}} \mathbb{U}^{d \times n}$ . Then, it executes Fmap where Sender holds  $\mathbf{Q}$  and Receiver holds  $\overline{\mathbf{W}}$  to get Sender's view  $\text{view}_{\mathcal{S}}^{\text{IFmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  and appends  $\text{view}_{\mathcal{S}}^{\text{IFmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  to the view.

PARAMETERS:

- Sender  $\mathcal{S}$  and Receiver  $\mathcal{R}$ .
- Space dimension  $d$ , distance function  $\text{dist}(\cdot, \cdot)$  having translation invariance, and threshold  $\delta$ .
- An AHE scheme  $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec}, \oplus)$ .
- An OKVS scheme  $(\text{Encode}, \text{Decode})$  and its random value  $r$ .

INPUT OF  $\mathcal{S}$ :  $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m) \in \mathbb{U}^{d \times m}$ .  
INPUT OF  $\mathcal{R}$ :  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n) \in \mathbb{U}^{d \times n}$ .

PROTOCOL:

0.  $\mathcal{S}$  permutes  $\mathbf{Q}$  randomly.
1.  $\mathcal{S}$  and  $\mathcal{R}$  invoke  $\Pi_{\text{Fmap}}^{\text{dist}(\cdot, \cdot)}$ :  $\mathcal{S}$  acts as Sender with input  $\mathbf{Q}$  and  $\mathcal{R}$  acts as Receiver with input  $\mathbf{W}$ .  $\mathcal{S}$  receives  $\text{ID}(\mathbf{Q})$ ;  $\mathcal{R}$  receives  $\text{ID}(\mathbf{W})$ .
2.  $\mathcal{R}$  generates  $(\text{sk}, \text{pk}) \leftarrow \text{Gen}(1^\kappa)$ .  $\mathcal{R}$  initializes set  $\text{List} \leftarrow \emptyset$ .
3. For each  $i \in [n]$ ,  $\mathcal{R}$  update
$$\text{List} \leftarrow \text{List} \cup \left\{ \left( \text{id}_{\mathbf{w}_i}, (\text{Enc}_{\text{pk}}(w_{i,1}), \dots, \text{Enc}_{\text{pk}}(w_{i,d})) \right) \right\}_{\text{id}_{\mathbf{w}_i} \in \text{ID}(\mathbf{w}_i)}$$
4.  $\mathcal{R}$  encodes  $E \leftarrow \text{Encode}(\text{List}, r)$ , and sends  $E, \text{pk}$  to  $\mathcal{S}$ .
5. For each  $j \in [m]$  and each  $\ell \in [\text{rate}_{\mathcal{S}}]$ ,  $\mathcal{S}$  samples  $\text{mask}_{j,\ell} \leftarrow (\text{mask}_{j,\ell,k})_{k \in [d]} \stackrel{\mathcal{R}}{\leftarrow} \mathbb{U}^d$  and computes
$$\text{cipher}_{j,\ell} \leftarrow \left( \text{Enc}_{\text{pk}}(\text{mask}_{j,\ell,k}) \oplus_{\text{pk}} \text{Decode}(E, \text{id}_{\mathbf{q}_j, \ell}, r) \right)_{k \in [d]}$$

$$\mathbf{u}_{j,\ell} \leftarrow \text{mask}_{j,\ell} + \mathbf{q}_j$$

Then,  $\mathcal{S}$  sends  $(\text{cipher}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}}]}$  to  $\mathcal{R}$ .

6.  $\mathcal{R}$  computes  $(\mathbf{v}_{j,\ell} \leftarrow \text{Dec}_{\text{sk}}(\text{cipher}_{j,\ell}))_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}}]}$ .
7. For each  $j \in [m]$  and each  $\ell \in [\text{rate}_{\mathcal{S}}]$ ,  $\mathcal{S}$  and  $\mathcal{R}$  invoke  $\mathcal{F}_{\text{ssFMatch}}^{\text{dist}(\cdot, \cdot)}$ :  $\mathcal{S}$  acts as Sender with input  $\mathbf{u}_{j,\ell}$  and  $\mathcal{R}$  acts as Receiver with input  $\mathbf{v}_{j,\ell}$ .  $\mathcal{S}$  receives  $e_{\mathcal{S},j,\ell}$  and  $\mathcal{R}$  receives  $e_{\mathcal{R},j,\ell}$ .
8. For each  $j \in [m]$ ,  $\mathcal{S}$  and  $\mathcal{R}$  invoke  $\mathcal{F}_{\text{PEqT}}$ :  $\mathcal{S}$  acts as Sender with input  $(e_{\mathcal{S},j,\ell})_{\ell \in [\text{rate}_{\mathcal{S}}]}$  and  $\mathcal{R}$  acts as Receiver with input  $(e_{\mathcal{R},j,\ell})_{\ell \in [\text{rate}_{\mathcal{S}}]}$ .  $\mathcal{S}$  receives nothing and  $\mathcal{R}$  receives  $1 - e_j$ .  $\mathcal{R}$  learns  $\mathbf{e} = (e_1, \dots, e_m)$ .
9. For each  $j \in [m]$ ,  $\mathcal{S}$  and  $\mathcal{R}$  invoke  $\mathcal{F}_{\text{OT}}$ :  $\mathcal{S}$  acts as Sender with input  $(m_0 = \perp, m_1 = \mathbf{q}_j)$  and  $\mathcal{R}$  acts as Receiver with input  $e_j$ .  $\mathcal{S}$  receives nothing and  $\mathcal{R}$  receives  $m_{e_j}$ .
10.  $\mathcal{R}$  learns  $\mathbf{I}_{\text{fuzzy}} \leftarrow \{m_{e_j} | (j \in [m]) \wedge (e_j = 1)\}$  as output.

**Fig. 12.** FPSI for  $\text{dist}(\cdot, \cdot)$  with Translation Invariance from Fmap:  $\Pi_{\text{FPSI}}^{\text{dist}(\cdot, \cdot)}$

2.  $\text{Sim}_{\text{FPSI}}^S$  encodes OKVS  $\overline{E}$  with  $(2\delta + 1)dn \cdot \text{rate}_{\mathcal{R}}$  dummy key-value pairs, generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends  $\overline{E}, \overline{\text{pk}}$  to the view.
3.  $\text{Sim}_{\text{FPSI}}^S$  samples  $\left( \overline{\text{mask}}_{j,\ell} \leftarrow \left( \overline{\text{mask}}_{j,\ell,k} \stackrel{\mathcal{R}}{\leftarrow} \mathbb{U} \right)_{k \in [d]} \right)_{j \in [m], \ell \in [\text{rate}_S]}$  and appends them to the view.
4. For each  $j \in [m]$  and each  $\ell \in [\text{rate}_S]$ ,  $\text{Sim}_{\text{FPSI}}^S$  samples  $\overline{e}_{\mathcal{S},j,\ell} \stackrel{\mathcal{R}}{\leftarrow} \mathbb{U}$ , invokes ssFMatch simulator  $\text{Sim}_{\text{ssFMatch}}^S(\overline{\text{mask}}_{j,\ell} + \mathbf{q}_j, \overline{e}_{\mathcal{S},j,\ell})$ , and appends the output to the view.
5. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI}}^S$  invokes PEqT simulator  $\text{Sim}_{\text{PEqT}}^S\left(\left(\overline{e}_{\mathcal{S},j,\ell}\right)_{\ell \in [\text{rate}_S]}\right)$  and appends the output to the view.
6. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI}}^S$  invokes OT simulator  $\text{Sim}_{\text{OT}}^S(\perp, \mathbf{q}_j)$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{mqFRPMT}}^S$  is indistinguishable from the real one via a hybrid argument. We define five hybrid transcripts  $T_0, T_1, T_2, T_3, T_4$ , where  $T_0$  is the real view of  $\mathcal{S}$ , and  $T_4$  is the output of  $\text{Sim}_{\text{FPSI}}^S$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.12. Let  $T_0$  denote  $\mathcal{S}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that the Fmap execution is replaced by  $\text{view}_{\mathcal{S}}^{\text{Fmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{S}}^{\text{Fmap}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}}) \stackrel{c}{\approx} \text{view}_{\mathcal{S}}^{\text{Fmap}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.
- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that OKVS  $E$  and  $\text{pk}$  are replaced by  $\overline{E}$  and  $\overline{\text{pk}}$ . The indistinguishability follows from the IND-CPA security of  $\mathcal{E}$  and the obliviousness of OKVS. Therefore, we have  $T_2 \stackrel{c}{\approx} T_1$ .
- **Hyb<sub>3</sub>**. Let  $T_3$  be the same as  $T_2$ , except that  $(\overline{\text{mask}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$ ,  $(\overline{e}_{\mathcal{S},j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$ , the ssFMatch execution, and the PEqT execution are replaced by  $(\overline{\text{mask}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$ ,  $(\overline{e}_{\mathcal{S},j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$ , simulators  $\left(\text{Sim}_{\text{ssFMatch}}^S(\overline{\text{mask}}_{j,\ell} + \mathbf{q}_j, \overline{e}_{\mathcal{S},j,\ell})\right)_{j \in [m], \ell \in [\text{rate}_S]}$ , and simulators  $\left(\text{Sim}_{\text{PEqT}}^S\left(\left(\overline{e}_{\mathcal{S},j,\ell}\right)_{\ell \in [\text{rate}_S]}\right)\right)_{j \in [m]}$ . Since  $(\overline{\text{mask}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$  is statistically indistinguishable from  $(\overline{\text{mask}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$  and  $(\overline{e}_{\mathcal{S},j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$  is statistically indistinguishable from  $(\overline{e}_{\mathcal{S},j,\ell})_{j \in [m], \ell \in [\text{rate}_S]}$ , the semi-honest security of ssFMatch and PEqT functionality guarantees  $T_3 \stackrel{c}{\approx} T_2$ .
- **Hyb<sub>4</sub>**. Let  $T_4$  be the same as  $T_3$ , except that the OT execution is replaced by  $\left(\text{Sim}_{\text{OT}}^S(\perp, \mathbf{q}_j)\right)_{j \in [m]}$ .  $T_4 \stackrel{c}{\approx} T_3$  comes from the semi-honest security of OT functionality.

Clearly,  $\text{Sim}_{\text{FPSI}}^S$ 's simulated view is identical to  $\mathcal{S}$ 's view in the real execution. Corrupt Receiver:  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  simulating the view of corrupt semi-honest Receiver executes as follows:

1.  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  initializes  $\overline{\mathbf{Q}} \leftarrow \mathbf{I}_{\text{fuzzy}}$ , pads it with dummy points until its size is  $m$ , and permutes it randomly. And  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  computes  $\overline{\mathbf{e}} \leftarrow (\overline{e}_j)_{j \in [m]} \in$

- $\{0, 1\}^m$  where  $\overline{e_j} = 1$  if and only if  $\overline{\mathbf{q}_j}$  belongs to  $\mathbf{I}_{\text{fuzzy}}$ . Then, it executes Fmap where Sender holds  $\overline{\mathbf{Q}}$  and Receiver holds  $\mathbf{W}$  to get Receiver's view  $\text{view}_{\mathcal{R}}^{II_{\text{Fmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  and appends  $\text{view}_{\mathcal{R}}^{II_{\text{Fmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  to the view.
2.  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  generates  $(\overline{\text{sk}}, \overline{\text{pk}}) \leftarrow \text{Gen}(1^\kappa)$ , and appends them to the view.
  3. For each  $j \in [m]$  and each  $\ell \in [\text{rate}_{\mathcal{S}}]$ ,  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  generates  $d$  uniformly random AHE ciphertexts as  $\overline{\text{cipher}}_{j,\ell} \leftarrow (\text{cipher}_{j,\ell,k})_{k \in [d]}$ .  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  appends  $(\overline{\text{cipher}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}}]}$  to the view.
  4. For each  $j \in [m]$  and each  $\ell \in [\text{rate}_{\mathcal{R}}]$ ,  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  samples  $\overline{e_{\mathcal{R},j,\ell}} \xleftarrow{\mathcal{R}} \mathbb{U}$ , invokes ssFMatch simulator  $\text{Sim}_{\text{ssFMatch}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_{j,\ell}), \overline{e_{\mathcal{R},j,\ell}})$ , and appends the output to the view.
  5. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  invokes PEqT simulator  $\text{Sim}_{\text{PEqT}}^{\mathcal{R}}\left(\overline{(e_{\mathcal{R},j,\ell})_{\ell \in [\text{rate}_{\mathcal{S}}]}}\right), \overline{e_j}$  and appends the output to the view.
  6. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  invokes OT simulator  $\text{Sim}_{\text{OT}}^{\mathcal{R}}(\overline{e_j}, \overline{\mathbf{q}_j})$  and appends the output to the view.

Now we show that the view output by  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$  is indistinguishable from the real one via a hybrid argument. We define five hybrid transcripts  $T_0, T_1, T_2, T_3, T_4$ , where  $T_0$  is the real view of  $\mathcal{R}$ , and  $T_4$  is the output of  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$ .

- **Hyb<sub>0</sub>**. This hybrid is the real interaction described in Fig.12. Let  $T_0$  denote  $\mathcal{R}$ 's view in the real protocol.
- **Hyb<sub>1</sub>**. Let  $T_1$  be the same as  $T_0$ , except that the Fmap execution is replaced by  $\text{view}_{\mathcal{R}}^{II_{\text{Fmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$ . The security property of Fmap guarantees  $\text{view}_{\mathcal{R}}^{II_{\text{Fmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W}) \stackrel{c}{\approx} \text{view}_{\mathcal{R}}^{II_{\text{Fmap}}}(\kappa, \lambda; \mathbf{Q}, \mathbf{W})$ , thus  $T_1 \stackrel{c}{\approx} T_0$  holds.
- **Hyb<sub>2</sub>**. Let  $T_2$  be the same as  $T_1$ , except that  $(\overline{\text{sk}}, \overline{\text{pk}})$ ,  $(\overline{\text{cipher}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}}]}$ ,  $(\overline{e_{\mathcal{R},j,\ell}})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$ , and the ssFMatch execution are replaced by  $(\overline{\text{sk}}, \overline{\text{pk}})$ ,  $(\overline{\text{cipher}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$ ,  $(\overline{e_{\mathcal{R},j,\ell}})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$ , and simulators  $(\text{Sim}_{\text{ssFMatch}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_{j,\ell}), \overline{e_{\mathcal{R},j,\ell}}))_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$ . Since each  $\text{cipher}_{j,\ell,k}$  has a uniformly random mask  $\text{mask}_{j,\ell,k}$  from  $\mathcal{S}$ , we have  $(\overline{\text{cipher}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$   $\stackrel{s}{\approx}$   $(\overline{\text{cipher}}_{j,\ell})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$ . According to the ssFMatch functionality, the real  $(\overline{e_{\mathcal{R},j,\ell}})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$  are uniformly random, thus  $(\overline{e_{\mathcal{R},j,\ell}})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$  and  $(\overline{e_{\mathcal{R},j,\ell}})_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$  are statistically indistinguishable. Then, the semi-honest security of ssFMatch functionality guarantees that simulators  $(\text{Sim}_{\text{ssFMatch}}^{\mathcal{R}}(\text{Dec}_{\overline{\text{sk}}}(\overline{\text{cipher}}_{j,\ell}), \overline{e_{\mathcal{R},j,\ell}}))_{j \in [m], \ell \in [\text{rate}_{\mathcal{S}]}$  and the ssFMatch execution are computationally indistinguishable. Therefore, we have  $T_2 \stackrel{c}{\approx} T_1$ .
- **Hyb<sub>3</sub>**. Let  $T_3$  be the same as  $T_2$ , except that  $(e_j)_{j \in [m]}$  and the PEqT execution are replaced by  $(\overline{e_j})_{j \in [m]}$  and  $(\text{Sim}_{\text{PEqT}}^{\mathcal{R}}(\overline{(e_{\mathcal{R},j,\ell})_{\ell \in [\text{rate}_{\mathcal{S}]}}}, \overline{e_j}))_{j \in [m]}$ . Note that  $\mathcal{H}((e_j)_{j \in [m]}) = \mathcal{H}((\overline{e_j})_{j \in [m]}) = |\mathbf{I}_{\text{fuzzy}}|$ . Due to the random permutation over  $\mathbf{Q}$  and  $\overline{\mathbf{Q}}$ ,  $(e_j)_{j \in [m]}$  and  $(\overline{e_j})_{j \in [m]}$  are statistically indistinguishable. Then, the semi-honest security of PEqT functionality guarantees  $T_3 \stackrel{c}{\approx} T_2$ .

- **Hyb<sub>4</sub>**. Let  $T_4$  be the same as  $T_3$ , except that the OT execution is replaced by  $\left(\text{Sim}_{\text{OT}}^{\mathcal{R}}(\bar{e}_j, \mathbf{q}_j)\right)_{j \in [m]}$ . The semi-honest security of OT functionality guarantees  $T_4 \stackrel{c}{\approx} T_3$ .

Clearly,  $\text{Sim}_{\text{FPSI}}^{\mathcal{R}}$ 's simulated view is identical to  $\mathcal{R}$ 's view in the real execution.

*Remark 7 (Costs Analysis).* The communication cost of protocol presented in Fig.12 consists of: communication cost of Fmap, sending OKVS from  $n \cdot \text{rate}_{\mathcal{R}}$  pairs, sending  $m \cdot \text{rate}_{\mathcal{S}}$  masked ciphers of points, communication cost of  $m \cdot \text{rate}_{\mathcal{S}}$  ssFMatch, communication cost of  $m$  PEqT, and communication cost of  $m$  OT.

For  $\mathcal{S}$ , the computation cost of this protocol consists of: computation cost of Fmap as Sender,  $m \cdot \text{rate}_{\mathcal{S}}$  decoding of OKVS,  $m \cdot \text{rate}_{\mathcal{S}}$  homomorphic masking of points, computation cost of  $m \cdot \text{rate}_{\mathcal{S}}$  ssFMatch as Sender, computation cost of  $m$  PEqT as Sender, and computation cost of  $m$  OT as Sender.

For  $\mathcal{R}$ , the computation cost of this protocol consists of: computation cost of Fmap as Receiver,  $n \cdot \text{rate}_{\mathcal{R}}$  encryptions of points, encoding of OKVS with  $n \cdot \text{rate}_{\mathcal{R}}$  pairs,  $m \cdot \text{rate}_{\mathcal{S}}$  decryptions of points, computation cost of  $m \cdot \text{rate}_{\mathcal{S}}$  ssFMatch as Receiver, computation cost of  $m$  PEqT as Receiver, and computation cost of  $m$  OT as Receiver.

**FPSI for Functions with Invariance.** Note that our construction is not limited to distance functions with translation invariance, such as Hamming distance. For any function with some invariance, we can obtain FPSI from Fmap using a similar construction.

For example, a generic construction for function with rotation invariance, such as cosine similarity, can be proposed via simply replacing additive masks and AHE by rotational masks and homomorphic encryption allowing rotation on ciphertexts.

## 7.2 FPSI(-Variants) from mqFRPMT

As shown in Section 2.6, mqFRPMT can be used as a central building block to construct FPSI and its various variants, including LFPSI, FPSI-card, and the special FPSI-SP.

As shown in Fig.13, methods of constructing FPSI, LFPSI, and FPSI-card from mqFRPMT are the same as methods of constructing their counterparts from mqRPMT. Regarding exceptive FPSI-SP, we convert FPSI-SP into LFPSI using UFmap.

We provide proofs of correctness and security for FPSI-SP. For proofs for other protocols, we recommend interested readers to refer to the proofs in [9].

In Section 6.2 and Section 6.3, we present mqFRPMT protocols for  $L_{\infty}$  and  $L_{p \in [1, \infty)}$  distances respectively. Based on them, we can easily obtain FPSI for  $L_{p \in [1, \infty]}$  distance.

**Theorem 14 (Correctness).** *The protocol  $\Pi_{\text{FPSI-SP}}^{\text{dist}(\cdot, \cdot)}$  presented in Fig.13 realizes the functionality  $\mathcal{F}_{\text{FPSI-SP}}$  defined in Fig.1 for  $\text{dist}(\cdot, \cdot)$  correctly.*

<p>PARAMETERS:</p> <ul style="list-style-type: none"> <li>– Sender <math>\mathcal{S}</math> and Receiver <math>\mathcal{R}</math>.</li> <li>– Size of Sender's input <math>m</math>, size of Receiver's input <math>n</math>.</li> <li>– Distance function <math>\text{dist}(\cdot, \cdot)</math>.</li> </ul> <p>INPUT OF <math>\mathcal{S}</math>: <math>\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m) \in \mathbb{U}^{d \times m}</math>.  <math>\text{Label}_{\mathbf{Q}} = (\text{label}_1, \dots, \text{label}_m) \in \{0, 1\}^{\sigma \times m}</math> for LFPSI.</p> <p>INPUT OF <math>\mathcal{R}</math>: <math>\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n) \in \mathbb{U}^{d \times n}</math>.</p> <p>PROTOCOL:</p> <ol style="list-style-type: none"> <li>0. <math>\mathcal{S}</math> permutes <math>\mathbf{Q}</math> randomly.</li> <li>1. <math>\mathcal{S}</math> and <math>\mathcal{R}</math> invoke <math>\mathcal{F}_{\text{mqFRPMT}}^{\text{dist}(\cdot, \cdot)}</math>: <math>\mathcal{S}</math> acts as Sender with input <math>\mathbf{Q}</math> and <math>\mathcal{R}</math> acts as Receiver with input <math>\mathbf{W}</math>. <math>\mathcal{S}</math> receives nothing and <math>\mathcal{R}</math> receives <math>\mathbf{e} = (e_1, \dots, e_m)</math>.</li> <li>2.             <ul style="list-style-type: none"> <li>– <b>FPSI</b>: For each <math>j \in [m]</math>, <math>\mathcal{S}</math> and <math>\mathcal{R}</math> invoke <math>\mathcal{F}_{\text{OT}}</math>: <math>\mathcal{S}</math> acts as Sender with input <math>(m_0 = \perp, m_1 = \mathbf{q}_j)</math> and <math>\mathcal{R}</math> acts as Receiver with input <math>e_j</math>. <math>\mathcal{S}</math> receives nothing and <math>\mathcal{R}</math> receives <math>m_{e_j}</math>.  <math>\mathcal{R}</math> learns <math>\mathbf{I}_{\text{fuzzy}} \leftarrow \{m_{e_j}   (j \in [m]) \wedge (e_j = 1)\}</math> as output.</li> <li>– <b>LFPSI</b>: For each <math>j \in [m]</math>, <math>\mathcal{S}</math> and <math>\mathcal{R}</math> invoke <math>\mathcal{F}_{\text{OT}}</math>: <math>\mathcal{S}</math> acts as Sender with input <math>(m_0 = \perp, m_1 = \text{label}_j)</math> and <math>\mathcal{R}</math> acts as Receiver with input <math>e_j</math>. <math>\mathcal{S}</math> receives nothing and <math>\mathcal{R}</math> receives <math>m_{e_j}</math>.  <math>\mathcal{R}</math> learns <math>\text{Label}_{\text{fuzzy}} \leftarrow \{m_{e_j}   (j \in [m]) \wedge (e_j = 1)\}</math> as output.</li> <li>– <b>FPSI-card</b>: <math>\mathcal{R}</math> computes the Hamming weight of <math>\mathbf{e}</math> as output.</li> <li>– <b>FPSI-SP</b>:                     <ol style="list-style-type: none"> <li>i. <math>\mathcal{S}</math> and <math>\mathcal{R}</math> invoke <math>\Pi_{\text{UFmap}}^{\text{dist}(\cdot, \cdot)}</math>: <math>\mathcal{S}</math> acts as Sender with input <math>\mathbf{Q}</math> and <math>\mathcal{R}</math> acts as Receiver with input <math>\mathbf{W}</math>. <math>\mathcal{S}</math> receives <math>\text{ID}(\mathbf{Q})</math> and <math>\mathcal{R}</math> receives <math>\text{ID}(\mathbf{W})</math>.</li> <li>ii. For each <math>j \in [m]</math>, <math>\mathcal{S}</math> and <math>\mathcal{R}</math> invoke <math>\mathcal{F}_{\text{OT}}</math>: <math>\mathcal{S}</math> acts as Sender with input <math>(m_0 = \perp, m_1 = \text{id}_{\mathbf{q}_j})</math> and <math>\mathcal{R}</math> acts as Receiver with input <math>e_j</math>. <math>\mathcal{S}</math> receives nothing and <math>\mathcal{R}</math> receives <math>m_{e_j}</math>.</li> <li>iii. <math>\mathcal{R}</math> computes <math>\mathbf{I}_{\text{fuzzy}}^{\text{id}} \leftarrow \{m_{e_j}   (j \in [m]) \wedge (e_j = 1)\}</math>, and learns <math>\mathbf{I}_{\text{fuzzy}}^{\text{SP}} \leftarrow \{\mathbf{w}_i \in \mathbf{W}   \text{id}_{\mathbf{w}_i} \in \mathbf{I}_{\text{fuzzy}}^{\text{id}}\}</math> as output.</li> </ol> </li> </ul> </li> </ol>
--

**Fig. 13.** FPSI and Its Variants for  $\text{dist}(\cdot, \cdot)$  from mqFRPMT:  $\Pi_{\text{FPSI}}^{\text{dist}(\cdot, \cdot)}$ ,  $\Pi_{\text{LFPSI}}^{\text{dist}(\cdot, \cdot)}$ ,  $\Pi_{\text{FPSI-card}}^{\text{dist}(\cdot, \cdot)}$ , and  $\Pi_{\text{FPSI-SP}}^{\text{dist}(\cdot, \cdot)}$

*Proof.* For each  $\mathbf{w}_i \in \mathbf{W}$ , there are two cases.

- **Case 1: There exists  $\mathbf{q}_* \in \mathbf{Q}$  satisfying  $\text{dist}(\mathbf{q}_*, \mathbf{w}_i) \leq \delta$**   
 According to the correctness of mqFRPMT,  $e_* = 1$  holds, thus  $\mathcal{R}$  gets  $m_1 = \text{id}_{\mathbf{q}_*}$  via OT. Due to the correctness of UFmap,  $m_1$  also equals to  $\text{id}_{\mathbf{w}_i}$ .  
 Hence, we have  $\mathbf{w}_i \in \mathbf{I}_{\text{fuzzy}}^{\text{SP}}$  in this case.
- **Case 2: For any  $\mathbf{q}_j \in \mathbf{Q}$ , there is  $\text{dist}(\mathbf{q}_j, \mathbf{w}_i) > \delta$**   
 Suppose that  $\mathbf{w}_i$  belongs to  $\mathbf{I}_{\text{fuzzy}}^{\text{SP}}$ , in other words, there exists  $j' \in [m]$  such that  $\text{id}_{\mathbf{q}_{j'}} = \text{id}_{\mathbf{w}_i}$  and  $e_{j'} = 1$  hold. According to the correctness of mqFRPMT,  $e_{j'} = 1$  means there exists  $\mathbf{w}_{i'}$  such that  $\text{dist}(\mathbf{q}_{j'}, \mathbf{w}_{i'}) \leq \delta$ . Thus  $\text{id}_{\mathbf{q}_{j'}} = \text{id}_{\mathbf{w}_{i'}}$  comes from the correctness of UFmap. Therefore, we have  $\text{id}_{\mathbf{w}_i} = \text{id}_{\mathbf{w}_{i'}}$ , which contradicts the distinctiveness of UFmap.  
 Hence, we have  $\mathbf{w}_i \notin \mathbf{I}_{\text{fuzzy}}^{\text{SP}}$  in this case.

**Theorem 15 (Security).** *The protocol  $\Pi_{\text{FPSI-SP}}^{\text{dist}(\cdot, \cdot)}$  presented in Fig.13 realizes the functionality  $\mathcal{F}_{\text{FPSI-SP}}$  defined in Fig.1 for  $\text{dist}(\cdot, \cdot)$  against semi-honest adversaries in the  $(\mathcal{F}_{\text{mqFRPMT}}, \mathcal{F}_{\text{OT}})$ -hybrid model.*

*Proof.* We exhibit simulators  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}(\mathbf{Q})$  and  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}(\mathbf{W}, \mathbf{I}_{\text{fuzzy}}^{\text{SP}})$  for simulating corrupt  $\mathcal{S}$  and corrupt  $\mathcal{R}$  respectively. And we argue the indistinguishability of the produced transcript from the real execution.

Corrupt Sender:  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}$  simulating the view of corrupt semi-honest Sender executes as follows:

1.  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}$  invokes mqFRPMT simulator  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{S}}(\mathbf{Q})$  and appends the output to the view.
2.  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}$  samples  $\overline{\mathbf{W}} \leftarrow^{\mathcal{R}} \mathbb{U}^{d \times n}$ . Then, it executes UFmap where Sender holds  $\mathbf{Q}$  and Receiver holds  $\overline{\mathbf{W}}$  to get Sender's view  $\text{view}_{\mathcal{S}}^{\Pi_{\text{UFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  and appends  $\text{view}_{\mathcal{S}}^{\Pi_{\text{UFmap}}}(\kappa, \lambda; \mathbf{Q}, \overline{\mathbf{W}})$  to the view.
3. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}$  invokes OT simulator  $\text{Sim}_{\text{OT}}^{\mathcal{S}}(\perp, \overline{\text{id}}_{\mathbf{q}_j})$  and appends the output to the view.

By the underlying simulators' indistinguishability,  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{S}}$ 's simulated view is identical to  $\mathcal{S}$ 's view in the real execution.

Corrupt Receiver:  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  simulating the view of corrupt semi-honest Receiver executes as follows:

1.  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  initializes  $\overline{\mathbf{Q}} \leftarrow \mathbf{I}_{\text{fuzzy}}^{\text{SP}}$ , pads it with dummy points until its size is  $m$ , and permutes its elements randomly. Then,  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  computes  $\overline{\mathbf{e}} \leftarrow (\overline{e}_j)_{j \in [m]} \in \{0, 1\}^m$  where  $\overline{e}_j = 1$  if and only if  $\overline{\mathbf{q}}_j$  belongs to  $\mathbf{I}_{\text{fuzzy}}^{\text{SP}}$ .
2.  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  invokes mqFRPMT simulator  $\text{Sim}_{\text{mqFRPMT}}^{\mathcal{R}}(\mathbf{W}, \overline{\mathbf{e}})$  and appends the output to the view.
3.  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  executes UFmap where Sender holds  $\overline{\mathbf{Q}}$  and Receiver holds  $\mathbf{W}$  to get Sender's output  $\text{ID}(\overline{\mathbf{Q}})$  and Receiver's view  $\text{view}_{\mathcal{R}}^{\Pi_{\text{UFmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$ . Then, it appends  $\text{view}_{\mathcal{R}}^{\Pi_{\text{UFmap}}}(\kappa, \lambda; \overline{\mathbf{Q}}, \mathbf{W})$  to the view.
4. For each  $j \in [m]$ ,  $\text{Sim}_{\text{FPSI-SP}}^{\mathcal{R}}$  invokes OT simulator  $\text{Sim}_{\text{OT}}^{\mathcal{R}}(\overline{e}_j, \overline{\text{id}}_{\mathbf{q}_j})$  and appends the output to the view.

In the simulation, the way  $\mathcal{R}$  learns  $\mathbf{I}_{\text{fuzzy}}^{\text{SP}}$  is identical to the real execution. By the underlying simulators' indistinguishability, the simulated view is computationally indistinguishable from the real view.

*Remark 8 (Complexity).* If the mqFRPMT is instantiated with  $\Pi_{\text{mqFRPMT}}^{L_{\infty}}$  in Fig.10 or  $\Pi_{\text{mqFRPMT}}^{L_{p \in \{1, \infty\}}}$  in Fig.11:

- For  $L_{\infty}$  distance, the FPSI protocol presented in Fig.13 has communication complexity  $\mathcal{O}(((2\delta + 1)d\kappa + 2\kappa + \kappa)(m + n))$ , computational complexity  $\mathcal{O}((2\delta + 1)dm + n)$  for  $\mathcal{S}$ , and computational complexity  $\mathcal{O}((2\delta + 1)dn + m)$  for  $\mathcal{R}$ .



- For  $L_{p \in [1, \infty]}$  distance, the FPSI protocol presented in Fig.13 has communication complexity  $\mathcal{O}(((2\delta + 1)d\kappa + 2\kappa + \kappa)(m + n) + mp \log \delta\kappa)$ , computational complexity  $\mathcal{O}((2\delta + 1)dm + n + p \log \delta m)$  for  $\mathcal{S}$ , and computational complexity  $\mathcal{O}((2\delta + 1)dn + m + p \log \delta m)$  for  $\mathcal{R}$ .

*Remark 9 (Limitation of  $\Pi_{\text{FPSI-SP}}^{\text{dist}(\cdot, \cdot)}$  Presented in Fig.13).* This construction of FPSI-SP relies on UFmap. However, we are currently unable to construct a UFmap instance for Hamming distance. Therefore, this construction cannot instantiate the FPSI-SP protocol for Hamming distance yet.

In fact, as far as we know, there is still no FPSI-SP protocol for Hamming distance.

## 8 Implementation

We provide experimental details and specific data for FPSI, and compare our performance with previous works. We also conduct experiments in unbalanced setting for interested readers.

### 8.1 Implementation Details

**Environment.** We run the experiments on a single machine with 2.00GHz Intel Xeon Gold 6330 CPU and 256 GB RAM. We measure the time of online phase in a local network setting with network latency of 0.02 ms and bandwidth of 10 Gbps.

**Instantiations.** We choose the computational security parameter  $\kappa = 128$  and the statistical security parameter  $\lambda = 40$ . Our protocols are written in C++ and we use the following instantiations in our implementation<sup>11</sup>.

- OKVS: We use RB-OKVS in [3].
- OT: We use OT implementation in libOTe<sup>12</sup>.
- Goldwasser-Micali: We use GMP<sup>13</sup> to implement Goldwasser-Micali cryptosystem with key size of 2048-bit as AHE in our FPSI for Hamming distance.
- Paillier: We use the implementation of Paillier in Intel Paillier Cryptosystem Library<sup>14</sup> with key size of 2048-bit as AHE in our FPSI for  $L_{p \in [1, \infty]}$  distance.
- Others: We use Curve25519 in cryptoTools<sup>15</sup> as the underlying group  $\mathbb{G}$  for SAS Fmap. We adopt Coproto<sup>16</sup> to realize network communication.

<sup>11</sup> The implementation is available in

<https://github.com/ql70ql70/Fuzzy-Private-Set-Intersection-from-Fuzzy-Mapping>

<sup>12</sup> <https://github.com/osu-crypto/libOTe.git>

<sup>13</sup> <https://gmplib.org/>

<sup>14</sup> <https://github.com/intel/pailliercryptolib.git>

<sup>15</sup> <https://github.com/ladnir/cryptoTools.git>

<sup>16</sup> <https://github.com/Visa-Research/coproto.git>

## 8.2 Performance in Balanced Setting

**FPSI for Hamming Distance.** We compare our FPSI form UniqC Fmap for Hamming distance in Section 7.1 with the near-linear protocol by Chongchitmate et al., which is the only one overcomes the  $m \cdot n$  blowup in complexity among prior works for Hamming distance [11]. Unfortunately, we do not have their code, thus we use their experimental results directly from their paper [11] and run our code with the same parameters.

The comparison is shown in Table 3. It can be observed that as  $m$  and  $n$  increase from 256 to 4096, the communication and computation costs of our protocol both scale linearly, and our protocol performs better than [11] in all cases. Note that our protocol achieves a  $4.6\times$  reduction in communication cost, which is independent of the running environment.

**Table 3.** Communication cost and running time of FPSI for Hamming distance, where input set sizes  $m = n \in \{256, 1024, 4096\}$ , universe  $\mathbb{U} = \mathbb{F}_2$ , dimension  $d = 128$ , and threshold  $\delta = 4$ . UniqC Fmap packs  $P = 24$  bits as one super-component.

$m = n$	Protocol	Comm. (MB)	Time (s)
256	[11]	465.68	38.7
	Ours	<b>91.889</b>	<b>5.18</b>
1024	[11]	1779.3	147.85
	Ours	<b>367.53</b>	<b>19.428</b>
4096	[11]	6870	569.9
	Ours	<b>1470</b>	<b>76.00</b>

**FPSI for  $L_{p \in [1, \infty]}$  Distance.** We compare our FPSI from SAS Fmap in Section 7.2 with the state-of-the-art protocols in [1] including FPSI in low-dimensional (denoted by [1]L) and high-dimensional (denoted by [1]H) space. We report the performances for input sizes  $m = n \in \{2^4, 2^8, 2^{12}, 2^{16}\}$ , dimension  $d \in \{2, 6, 10\}$ , and threshold  $\delta \in \{10, 30\}$ . Since [1]H needs more than  $10^4$  seconds when  $n \geq 2^{12}$ , we omit these data in our tables.

*FPSI for  $L_{p \in \{1, 2\}}$  Distance.* Since there is no implementation of [1]H for  $L_{p \in [1, \infty]}$  distance, we estimate its costs with the hyper-parameter  $\rho = 0.5$  for  $L_1$  distance and  $\rho = 0.365$  for  $L_2$  distance as reported in [1]. For comparison, we assume that the costs of [1]H only consist of OKVS encoding and sending, and estimate the encoding to take 800 machine cycles per pair, which is the best performance of our machine. In short, we report a conservative estimates of [1]H for  $L_{p \in \{1, 2\}}$  distance in our table. Table 4 shows that, for  $L_1$  and  $L_2$  distance, our protocol achieves a  $28 - 166\times$  speedup and reduces communication cost by a factor of  $6 - 40\times$  when  $d \geq 6$ .

*FPSI for  $L_\infty$  Distance.* Table 5 shows that our protocol for  $L_\infty$  distance achieves a  $30 - 305\times$  speedup and reduces communication cost by a factor of  $6 - 67\times$  when  $d \geq 6$ .

**Table 4.** Communication cost (MB) and running time (s) of FPSI for  $L_{p \in \{1,2\}}$  distance.

$m = n$	Protocol	$(d, \delta)$											
		(2,10)		(6,10)		(10,10)		(2,30)		(6,30)		(10,30)	
		Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time
$L_1$ Distance													
$2^4$	[1]H	4.512	31.64	13.54	94.93	22.56	158.2	13.11	87.96	39.32	263.9	65.53	439.8
	[1]L	<b>0.178</b>	0.692	8.257	24.10	220.0	677.0	<b>0.532</b>	1.888	24.77	73.69	660.0	2042
	Ours	0.469	<b>0.374</b>	<b>1.371</b>	<b>0.840</b>	<b>2.274</b>	<b>1.261</b>	1.330	<b>0.742</b>	<b>3.951</b>	<b>1.884</b>	<b>6.570</b>	<b>2.636</b>
$2^8$	[1]H	290.7	2084	872.1	6253	1453	10422	844.5	5714	2533	17140	4222	28567
	[1]L	<b>2.854</b>	9.296	132.1	409.7	3520	11034	<b>8.510</b>	25.70	396.3	1225	$> 10^4$	$> 10^4$
	Ours	7.502	<b>4.057</b>	<b>21.84</b>	<b>9.570</b>	<b>36.38</b>	<b>15.23</b>	21.28	<b>8.433</b>	<b>63.21</b>	<b>22.81</b>	<b>105.2</b>	<b>37.37</b>
$2^{12}$	[1]L	<b>45.66</b>	148.2	2113	6630	$> 10^4$	$> 10^5$	<b>136.2</b>	433.3	$> 6000$	$> 10^4$	$> 10^5$	$> 10^5$
	Ours	120.0	<b>56.92</b>	<b>351.0</b>	<b>155.0</b>	<b>589.2</b>	<b>260.2</b>	340.3	<b>130.2</b>	<b>1024</b>	<b>395.1</b>	<b>1703</b>	<b>650.4</b>
$2^{16}$	[1]L	<b>730.5</b>	2480	$> 10^4$	$> 10^5$	$> 10^5$	$> 10^6$	<b>2179</b>	7008	$> 10^4$	$> 10^5$	$> 10^6$	$> 10^6$
	Ours	1919	<b>966.3</b>	<b>5685</b>	<b>2736</b>	<b>9427</b>	<b>4359</b>	5513	<b>2238</b>	<b>16382</b>	<b>6416</b>	<b>27253</b>	<b>10800</b>
$L_2$ Distance													
$2^4$	[1]H	3.117	21.86	9.352	65.60	15.59	109.3	9.050	60.78	27.16	182.3	45.30	303.9
	[1]L	<b>0.222</b>	0.844	8.300	24.19	220.1	677.9	<b>0.957</b>	3.082	25.19	74.80	660.4	2046
	Ours	0.475	<b>0.372</b>	<b>1.377</b>	<b>0.889</b>	<b>2.279</b>	<b>1.181</b>	1.339	<b>0.820</b>	<b>3.960</b>	<b>1.783</b>	<b>6.581</b>	<b>2.801</b>
$2^8$	[1]H	137.2	983.4	411.5	2950	685.8	4917	398.4	2695	1195	8087	1992	13478
	[1]L	<b>3.557</b>	11.19	132.8	411.9	3521	11042	<b>15.31</b>	45.34	403.1	1246	$> 10^4$	$> 10^4$
	Ours	7.588	<b>4.307</b>	<b>22.03</b>	<b>9.882</b>	<b>36.91</b>	<b>16.25</b>	21.42	<b>8.825</b>	<b>63.35</b>	<b>23.18</b>	<b>106.6</b>	<b>38.97</b>
$2^{12}$	[1]L	<b>56.91</b>	180.4	2124	6657	$> 10^4$	$> 10^5$	<b>244.9</b>	742.6	$> 6000$	$> 10^4$	$> 10^5$	$> 10^5$
	Ours	122.8	<b>64.42</b>	<b>356.7</b>	<b>164.7</b>	<b>590.6</b>	<b>264.8</b>	346.8	<b>142.3</b>	<b>1026</b>	<b>402.7</b>	<b>1706</b>	<b>657.2</b>
$2^{16}$	[1]L	<b>910.5</b>	2992	$> 10^4$	$> 10^5$	$> 10^5$	$> 10^6$	<b>3919</b>	12017	$> 10^4$	$> 10^5$	$> 10^6$	$> 10^6$
	Ours	1964	<b>1070</b>	<b>5707</b>	<b>2765</b>	<b>9449</b>	<b>4443</b>	5549	<b>2366</b>	<b>16419</b>	<b>6539</b>	<b>27289</b>	<b>10953</b>

**Table 5.** Communication cost (MB) and running time (s) of FPSI for  $L_\infty$  distance.

$m = n$	Protocol	$(d, \delta)$											
		(2,10)		(6,10)		(10,10)		(2,30)		(6,30)		(10,30)	
		Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time
$2^4$	[1]H	2.073	5.333	18.65	45.70	52.03	122.9	17.55	43.37	158.0	298.1	439.4	759.3
	[1]L	<b>0.173</b>	0.660	8.251	24.09	220.0	677.1	<b>0.517</b>	1.891	24.75	73.61	660.0	2042
	Ours	0.470	<b>0.347</b>	<b>1.384</b>	<b>0.825</b>	<b>2.298</b>	<b>1.282</b>	1.340	<b>0.696</b>	<b>3.994</b>	<b>1.727</b>	<b>6.648</b>	<b>2.501</b>
$2^8$	[1]H	33.22	78.67	298.8	734.6	833.7	2011	281.0	697.5	2528	4933	$> 7000$	$> 10^4$
	[1]L	<b>2.766</b>	9.047	132.0	408.9	3520	11027	<b>8.266</b>	25.10	396.0	1225	$> 10^4$	$> 10^4$
	Ours	7.518	<b>3.732</b>	<b>22.14</b>	<b>9.029</b>	<b>36.75</b>	<b>14.96</b>	21.44	<b>7.930</b>	<b>63.90</b>	<b>22.28</b>	<b>106.4</b>	<b>36.99</b>
$2^{12}$	[1]L	<b>44.25</b>	143.4	2112	6612	$> 10^4$	$> 10^5$	<b>132.3</b>	420.8	$> 6000$	$> 10^4$	$> 10^5$	$> 10^5$
	Ours	120.2	<b>53.74</b>	<b>354.1</b>	<b>151.1</b>	<b>588.0</b>	<b>253.2</b>	343.0	<b>128.9</b>	<b>1022</b>	<b>391.4</b>	<b>1702</b>	<b>644.1</b>
$2^{16}$	[1]L	<b>708.0</b>	2401	$> 10^4$	$> 10^5$	$> 10^5$	$> 10^6$	<b>2116</b>	6796	$> 10^4$	$> 10^5$	$> 10^6$	$> 10^6$
	Ours	1924	<b>945.6</b>	<b>5665</b>	<b>2623</b>	<b>9408</b>	<b>4332</b>	5488	<b>2218</b>	<b>16358</b>	<b>6366</b>	<b>27228</b>	<b>10779</b>

### 8.3 Performance in Unbalanced Setting

We also conduct experiments in unbalanced setting where  $m$  could not equal  $n$  to demonstrate the different effects of  $m$  and  $n$  on our protocols' performances.

**FPSI for Hamming Distance.** We report the performances of our FPSI for Hamming distance with input sizes  $m, n \in \{2^8, 2^{10}, 2^{12}\}$ . We choose universe  $\mathbb{U} = \mathbb{F}_2$ , dimension  $d = 128$ , and threshold  $\delta = 4$ . UniqC Fmap packs  $P = 24$  bits as one super-component.

**Table 6.** Communication cost (MB) and running time (s) of our FPSI for Hamming distance in unbalanced setting.

$m$	$n$					
	$2^8$		$2^{10}$		$2^{12}$	
	Comm.	Time	Comm.	Time	Comm.	Time
$2^8$	91.89	5.184	223.9	6.704	752.0	12.74
$2^{10}$	235.5	17.61	367.5	19.43	895.7	25.91
$2^{12}$	809.9	66.06	942.0	68.94	1470	76.00

Table 6 shows that the size of Sender's set  $m$  has a greater impact on the costs of our FPSI protocol for Hamming distance. This is because  $m$  has a greater multiplicative coefficient in the complexity.

**FPSI for  $L_{p \in \{1, 2, \infty\}}$  Distance.** We report the performances of our FPSI for  $L_{p \in \{1, 2, \infty\}}$  distance with input sizes  $m, n \in \{2^4, 2^8, 2^{12}, 2^{16}\}$ . We choose dimension  $d = 2$  and threshold  $\delta = 10$ .

Table 7 shows that the size of Receiver's set  $n$  has a greater impact on the communication cost of our FPSI protocol for  $L_{p \in \{1, 2, \infty\}}$  distance.

## 9 Conclusion

In this work, we abstract a new primitive called Fmap, which is a powerful technique for FPSI. Many existing FPSI protocols are based on Fmap and their complexity bottlenecks mainly due to high expansion rates of their Fmap instances. We give new constructions of Fmap with small expansion rates for Hamming and  $L_{p \in [1, \infty]}$  distances to break bottlenecks.

We report a generic construction of FPSI from Fmap, which leads to the first FPSI for Hamming distance of which costs are strictly linear with  $m$  and  $n$ . Meanwhile, we show a construction of mqFRPMT from sUFmap, an enhanced Fmap. We propose an FPSI(-variants) framework from mqFRPMT. Using this framework, we finally get FPSI for  $L_{p \in [1, \infty]}$  distance of which costs scale linearly with anyone of  $m$ ,  $n$ ,  $d$ , and  $\delta$  for the first time.

Regarding future works, we present the following thoughts:

**Table 7.** Communication cost (MB) and running time (s) of our FPSI for  $L_{p \in \{1,2,\infty\}}$  distance in unbalanced setting.

$m$	$n$							
	$2^4$		$2^8$		$2^{12}$		$2^{16}$	
	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time
$L_\infty$ Distance								
$2^4$	0.470	0.347	6.668	3.253	105.8	50.63	1692	840.2
$2^8$	1.321	3.195	7.518	3.732	106.7	52.12	1693	852.2
$2^{12}$	14.87	48.16	21.07	49.03	120.2	53.74	1707	872.4
$2^{16}$	231.7	797.0	237.9	807.7	337.1	825.8	1924	945.6
$L_1$ Distance								
$2^4$	0.469	0.374	6.672	3.361	105.8	50.52	1692	840.1
$2^8$	1.395	3.699	7.502	4.057	106.8	52.16	1693	852.5
$2^{12}$	16.06	54.14	22.26	54.98	120.0	56.92	1707	874.1
$2^{16}$	250.7	877.1	256.9	887.8	356.1	902.4	1919	966.3
$L_2$ Distance								
$2^4$	0.475	0.372	6.678	3.307	105.8	50.64	1692	842.6
$2^8$	1.481	3.985	7.588	4.307	106.8	52.50	1693	855.3
$2^{12}$	17.44	58.29	23.63	59.25	122.8	64.42	1709	881.5
$2^{16}$	272.7	947.5	278.9	959.4	378.1	977.1	1964	1070

- The distinctiveness property of Fmap is intended to make subsequent OKVS encoding possible, which seems unnatural. How to avoid the distinctiveness property to gain a more general abstraction is an interesting question.
- Our FPSI for  $L_{p \in [1,\infty]}$  distance uses  $R \wedge S$ . disj. proj., a stronger assumption than  $R$ . disj. proj. of [1], to obtain optimal complexity. Is it possible to construct a protocol under a more realistic assumption (i.e., something weaker than  $R \wedge S$ . disj. proj.) to achieve a near-linear asymptotic complexity and a practical efficiency comparable to the protocol in this work? Any relevant progress would be quite valuable.
- All these protocols above are in the semi-honest setting. We leave the construction of efficient FPSI protocols in the malicious setting as future work.

## Acknowledgement

We thank all the anonymous reviewers for helpful feedback. This work is supported by the National Key Research and Development Program of China (2022YFB2701600).

## References

1. van Baarsen, A., Pu, S.: Fuzzy private set intersection with large hyperballs. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024.

- pp. 340–369. Springer Nature Switzerland, Cham (2024). [https://doi.org/10.1007/978-3-031-58740-5\\_12](https://doi.org/10.1007/978-3-031-58740-5_12)
2. Badrinarayanan, S., Miao, P., Raghuraman, S., Rindal, P.: Multi-party threshold private set intersection with sublinear communication. In: Garay, J.A. (ed.) *Public-Key Cryptography – PKC 2021*. pp. 349–379. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-75248-4\\_13](https://doi.org/10.1007/978-3-030-75248-4_13)
  3. Bienstock, A., Patel, S., Seo, J.Y., Yeo, K.: Near-Optimal oblivious Key-Value stores for efficient PSI, PSU and Volume-Hiding Multi-Maps. In: *32nd USENIX Security Symposium (USENIX Security 23)*. pp. 301–318. USENIX Association, Anaheim, CA (2023)
  4. Bienstock, A., Patel, S., Seo, J.Y., Yeo, K.: Batch pir and labeled psi with oblivious ciphertext compression. *Cryptology ePrint Archive*, Paper 2024/215 (2024), <https://eprint.iacr.org/2024/215>
  5. Branco, P., Döttling, N., Pu, S.: Multiparty cardinality testing for threshold private intersection. In: Garay, J.A. (ed.) *Public-Key Cryptography – PKC 2021*. pp. 32–60. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-75248-4\\_2](https://doi.org/10.1007/978-3-030-75248-4_2)
  6. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*. p. 494–503. STOC '02, Association for Computing Machinery, New York, NY, USA (2002). <https://doi.org/10.1145/509907.509980>
  7. Chakraborti, A., Fanti, G., Reiter, M.K.: Distance-Aware private set intersection. In: *32nd USENIX Security Symposium (USENIX Security 23)*. pp. 319–336. USENIX Association, Anaheim, CA (2023)
  8. Chen, H., Huang, Z., Laine, K., Rindal, P.: Labeled psi from fully homomorphic encryption with malicious security. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. pp. 1223–1237. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3243734.3243836>
  9. Chen, Y., Zhang, M., Zhang, C., Dong, M., Liu, W.: Private set operations from multi-query reverse private membership test. In: Tang, Q., Teague, V. (eds.) *Public-Key Cryptography – PKC 2024*. pp. 387–416. Springer Nature Switzerland, Cham (2024). [https://doi.org/10.1007/978-3-031-57725-3\\_13](https://doi.org/10.1007/978-3-031-57725-3_13)
  10. Chmielewski, L., Hoepman, J.H.: Fuzzy private matching (extended abstract). In: *2008 Third International Conference on Availability, Reliability and Security*. pp. 327–334 (2008). <https://doi.org/10.1109/ARES.2008.170>
  11. Chongchitmate, W., Lu, S., Ostrovsky, R.: Approximate psi with near-linear communication. *Cryptology ePrint Archive*, Paper 2024/682 (2024), <https://eprint.iacr.org/2024/682>
  12. Cong, K., Moreno, R.C., da Gama, M.B., Dai, W., Iliashenko, I., Laine, K., Rosenberg, M.: Labeled PSI from homomorphic encryption with reduced computation and communication. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. pp. 1135–1150. CCS '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3460120.3484760>
  13. Duong, T., Phan, D.H., Trieu, N.: Catalic: Delegated psi cardinality with applications to contact tracing. In: Moriai, S., Wang, H. (eds.) *Advances in Cryptology – ASIACRYPT 2020*. pp. 870–899. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-64840-4\\_29](https://doi.org/10.1007/978-3-030-64840-4_29)

14. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory* **31**(4), 469–472 (1985). <https://doi.org/10.1109/TIT.1985.1057074>
15. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) *Advances in Cryptology - EUROCRYPT 2004*. pp. 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_1](https://doi.org/10.1007/978-3-540-24676-3_1)
16. Garimella, G., Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: Oblivious key-value stores and amplification for private set intersection. In: *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II*. pp. 395–425. Springer-Verlag, Berlin, Heidelberg (2021). [https://doi.org/10.1007/978-3-030-84245-1\\_14](https://doi.org/10.1007/978-3-030-84245-1_14)
17. Garimella, G., Rosulek, M., Singh, J.: Structure-aware private set intersection, with applications to fuzzy matching. In: Dodis, Y., Shrimpton, T. (eds.) *Advances in Cryptology – CRYPTO 2022*. pp. 323–352. Springer Nature Switzerland, Cham (2022). [https://doi.org/10.1007/978-3-031-15802-5\\_12](https://doi.org/10.1007/978-3-031-15802-5_12)
18. Garimella, G., Rosulek, M., Singh, J.: Malicious secure, structure-aware private set intersection. In: Handschuh, H., Lysyanskaya, A. (eds.) *Advances in Cryptology – CRYPTO 2023*. pp. 577–610. Springer Nature Switzerland, Cham (2023). [https://doi.org/10.1007/978-3-031-38557-5\\_19](https://doi.org/10.1007/978-3-031-38557-5_19)
19. Ghosh, S., Simkin, M.: The communication complexity of threshold private set intersection. In: Boldyreva, A., Micciancio, D. (eds.) *Advances in Cryptology – CRYPTO 2019*. pp. 3–29. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-030-26951-7\\_1](https://doi.org/10.1007/978-3-030-26951-7_1)
20. Ghosh, S., Simkin, M.: Threshold private set intersection with better communication complexity. In: Boldyreva, A., Kolesnikov, V. (eds.) *Public-Key Cryptography – PKC 2023*. pp. 251–272. Springer Nature Switzerland, Cham (2023). [https://doi.org/10.1007/978-3-031-31371-4\\_9](https://doi.org/10.1007/978-3-031-31371-4_9)
21. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*. pp. 365–377. STOC '82, Association for Computing Machinery, New York, NY, USA (1982). <https://doi.org/10.1145/800070.802212>
22. Indyk, P., Woodruff, D.: Polylogarithmic private approximations and efficient matching. In: Halevi, S., Rabin, T. (eds.) *Theory of Cryptography*. pp. 245–264. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). [https://doi.org/10.1007/11681878\\_13](https://doi.org/10.1007/11681878_13)
23. Ion, M., Kreuter, B., Nergiz, A.E., Patel, S., Saxena, S., Seth, K., Raykova, M., Shanahan, D., Yung, M.: On deploying secure computing: Private intersection-sum-with-cardinality. In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. pp. 370–389 (2020). <https://doi.org/10.1109/EuroSP48549.2020.00031>
24. Kolesnikov, V., Kumaresan, R., Rosulek, M., Trieu, N.: Efficient batched oblivious prf with applications to private set intersection. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. pp. 818–829. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2976749.2978381>
25. Liu, F.H., Zhang, E., Qin, L.: Efficient multiparty probabilistic threshold private set intersection. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. pp. 2188–2201. CCS '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3576915.3623158>

26. Manku, G.S., Jain, A., Das Sarma, A.: Detecting near-duplicates for web crawling. In: Proceedings of the 16th International Conference on World Wide Web. p. 141–150. WWW '07, Association for Computing Machinery, New York, NY, USA (2007). <https://doi.org/10.1145/1242572.1242592>
27. Mohammadi-Kambs, M., Hölz, K., Somoza, M.M., Ott, A.: Hamming distance as a concept in dna molecular recognition. ACS Omega **2**(4), 1302–1308 (2017). <https://doi.org/10.1021/acsomega.7b00053>
28. Oded, G.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, USA, 1st edn. (2009)
29. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99. pp. 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). [https://doi.org/10.1007/3-540-48910-X\\_16](https://doi.org/10.1007/3-540-48910-X_16)
30. Patra, A., Schneider, T., Suresh, A., Yalame, H.: ABY2.0: Improved Mixed-Protocol secure Two-Party computation. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2165–2182. USENIX Association (2021), <https://www.usenix.org/conference/usenixsecurity21/presentation/patra>
31. Rabin, M.O.: How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Paper 2005/187 (2005), <https://eprint.iacr.org/2005/187>, <https://eprint.iacr.org/2005/187>
32. Raghuraman, S., Rindal, P.: Blazing fast psi from improved okvs and subfield vole. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2505–2517. CCS '22, Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3548606.3560658>
33. Rindal, P., Schoppmann, P.: Vole-psi: Fast oprf and circuit-psi from vector-ole. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 901–930. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-77886-6\\_31](https://doi.org/10.1007/978-3-030-77886-6_31)
34. Uzun, E., Chung, S.P., Kolesnikov, V., Boldyreva, A., Lee, W.: Fuzzy labeled private set intersection with applications to private Real-Time biometric search. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 911–928. USENIX Association (2021)
35. Uzun, E., Yagemann, C., Chung, S., Kolesnikov, V., Lee, W.: Cryptographic key derivation from biometric inferences for remote authentication. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. p. 629–643. ASIA CCS '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3433210.3437512>
36. Wu, M., Yuen, T.H.: Efficient unbalanced private set intersection cardinality and user-friendly privacy-preserving contact tracing. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 283–300. USENIX Association, Anaheim, CA (2023)
37. Ye, Q., Steinfeld, R., Pieprzyk, J., Wang, H.: Efficient fuzzy matching and intersection on private datasets. In: Lee, D., Hong, S. (eds.) Information, Security and Cryptology – ICISC 2009. pp. 211–228. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14423-3\\_15](https://doi.org/10.1007/978-3-642-14423-3_15)