Efficient Fuzzy Private Set Intersection from Fuzzy Mapping

Ying Gao$^{1,2(\boxtimes)[0000-0001-8992-651X]}$, Lin Qi$^{1[0009-0003-0489-9412]}$, Xiang Liu$^{1[0009-0000-8128-7872]}$, Yuanchao Luo$^{1[0009-0001-6541-690X]}$, and Longxin Wang$^{1[0009-0007-2277-7039]}$

[1] School of Cyber Science and Technology, Beihang University, Beijing, China
{gaoying, 19373287, lx1234, 19373507, wlx_buaa}@buaa.edu.cn
[2] 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 a 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_∞ 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 $L_{\mathsf{p}\in[1,\infty]}$ distances. For Hamming distance, our protocol is the first one that achieves strict linear complexity with input sizes. For $L_{\mathsf{p}\in[1,\infty]}$ 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 [17,21,22] have been designed in last decades. And recent PSI protocols have achieved extremely high efficiency [21]. Facing various complex practical needs, there is also a growing interest in works on variants of PSI, including labeled PSI (LPSI) [3, 5, 9], which outputs labels associated with elements in intersection to Receiver; PSI cardinality (PSI-card) [10,16,25], which only reveals intersection cardinality to Receiver.

This work focus 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 δ, 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 [23], illegal content detection [1], and vulnerable password detection [4]. 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 [11] in 2004, there has been a long list of works related to FPSI [1, 4, 7, 8, 13–15, 23, 26]. The majority of them concern with FPSI for Hamming distance, and the few exceptions [13–15] only consider about one or two of L_1, L_2, and L_∞ distances. Until 2024, Baarsen and Pu [1] make a breakthrough by presenting the first FPSI protocol supporting general L_p distance with $\mathsf{p} \in [1, \infty]$. As the state-of-the-art FPSI for $L_{\mathsf{p}\in[1,\infty]}$ 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 [11], 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 [8].

Second, the efficiency of existing FPSI protocols for $L_{\mathsf{p}\in[1,\infty]}$ 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*[3]*?*
- *Can we construct an FPSI protocol for $L_{\mathsf{p}\in[1,\infty]}$ distance of which costs scale linearly with anyone of m, n, d, and δ?*

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 [1,4,7,11,13,15,23,26] actually are based on instances of Fmap, and complexity bottlenecks in these protocols are derived 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_∞ distance with expansion rate of 1.
- **FPSI for Hamming Distance of Which Costs Scale Strictly Linearly with $\boldsymbol{m}$ and $\boldsymbol{n}$.** 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 $\boldsymbol{L_{\mathsf{p}\in[1,\infty]}}$ Distance of Which Costs Scale Linearly with Anyone of $\boldsymbol{m}$, $\boldsymbol{n}$, $\boldsymbol{d}$, and $\boldsymbol{\delta}$.** 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 variants, including FPSI-card, LFPSI, and FPSI-SP. By instantiating with our Fmap instance for L_∞ distance with expansion rate of 1, we ultimately

[3] 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.

construct a new FPSI protocol for $L_{\mathsf{p}\in[1,\infty]}$ distance. Its costs scale linearly with any one of m, n, d, and δ, 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× reduction in communication cost for Hamming distance when both parties input 128-bit binary strings and δ is set to 4, and achieves a $28-166\times$ speedup and $6-40\times$ reduction in communication cost for $L_{\mathsf{p}\in\{1,2,\infty\}}$ 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 $L_{\mathsf{p}\in[1,\infty]}$ distance. A comparison of asymptotic complexities is given in Table 1.

FPSI for Hamming Distance. Freedman et al. [11] 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 [7]. For a long time, subsequent works on FPSI mainly focus on Hamming distance. Ye et al. [26] design FPSI for Hamming distance with oblivious polynomial evaluation technique. Indyk and Woodruff [15] deal with FPSI for Hamming and L_2 distances, but their protocols rely on AHE and costly garbled circuits. Uzun et al. [23] construct LFPSI for Hamming distance based on fully homomorphic encryption (FHE), another costly technique. Using vector oblivious linear evaluation, Chakraborti et al. [4] 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 $m \cdot n$ explosion in communication and computation complexities. In 2024, Chongchitmate et al. [8] propose the most efficient FPSI for Hamming distance to date. Their protocol reduces the $m \cdot n$ explosion 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 [4, 23], they only consider Hamming distance over $\mathbb{F}_2$.

FPSI for $L_{\mathsf{p}\in[1,\infty]}$ Distance. In 2022, Garimella et al. [13] construct the first FPSI protocols for L_1 and L_∞ 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 in computation cost. In 2024, Baarsen and Pu [1] propose the first FPSI protocol supporting general $L_{\mathsf{p}\in[1,\infty]}$ 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 [8].

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. δ 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 κ and statistical security parameter λ.

Distance	Protocol	Assumption	Communication	Computation	
				Sender	**Receiver**
Hamming	[26]	–	$\mathcal{O}(d^2mn)$	$\mathcal{O}(\mathsf{poly}(d)mn)$	$\mathcal{O}(d^2mn)$
	[23]$^{\ominus}$	FPR&FNR	$\mathcal{O}(B_1dmn)$	$\mathcal{O}(B_2dmn)$	$\mathcal{O}(\binom{d}{\delta}n)$
	[4]$^{\ominus}$	FNR	$\mathcal{O}(\delta^2mn)$	$\mathcal{O}((d+\delta^2)mn)$	$\mathcal{O}((d+\delta)mn)$
	[8]$^{\ominus}$	$\mathsf{R} \wedge \mathsf{S}$. cluster.	$\mathcal{O}(dM\log M)$	$\mathcal{O}(dM\log M)$	$\mathcal{O}(dM\log M)$
	Ours	R. UniqC	$\mathcal{O}(d^2m+\delta dn)$	$\mathcal{O}(d^2m)$	$\mathcal{O}(d^2m+\delta dn)$
$\boldsymbol{L_\infty}$	[13]	R. $l_{min} > 2\delta$	$\mathcal{O}\left(m+(4\log\delta)^dn\right)$	$\mathcal{O}\left((2\log\delta)^dm\right)$	$\mathcal{O}\left((2\delta)^dn\right)$
	[1]	R. $l_{min} > 2\delta$	$\mathcal{O}(2^dm+\delta dn)$	$\mathcal{O}(2^ddm)$	$\mathcal{O}(2^dm+\delta dn)$
		R. disj. proj.	$\mathcal{O}(m+(\delta d)^2n)$	$\mathcal{O}(d^2m)$	$\mathcal{O}(m+(\delta d)^2n)$
	Ours	$\mathsf{R} \wedge \mathsf{S}$. disj. proj.	$\mathcal{O}(\delta dm+\delta dn)$	$\mathcal{O}(\delta dm+n)$	$\mathcal{O}(m+\delta dn)$
$\boldsymbol{L_\mathsf{p}}$	[1]	R. $l_{min} > 2\delta\left(d^{\frac{1}{\mathsf{p}}}+1\right)$	$\mathcal{O}(\delta^\mathsf{p}m+\delta 2^ddn)$	$\mathcal{O}((d+\delta^\mathsf{p})m)$	$\mathcal{O}(m+\delta 2^ddn)$
		R. $l_{min} > \frac{1}{\rho}\delta$	$\mathcal{O}((\delta^\mathsf{p}n^\rho\log n)m+\delta dn^{\rho+1})$	$\mathcal{O}(((d+\delta^\mathsf{p})n^\rho\log n)m)$	$\mathcal{O}((n^\rho\log n)m+\delta dn^{\rho+1})$
	Ours	$\mathsf{R} \wedge \mathsf{S}$. disj. proj.	$\mathcal{O}((\delta d+\mathsf{p}\log\delta)m+\delta dn)$	$\mathcal{O}((\delta d+\mathsf{p}\log\delta)m+n)$	$\mathcal{O}(\mathsf{p}\log\delta m+\delta dn)$

- $\ominus$ 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).
- $\mathsf{R} \wedge \mathsf{S}$. 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 δ 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 2δ from other Receiver's points.
- $\mathsf{R} \wedge \mathsf{S}$. disj. proj. means that the disj. proj. assumption should hold for both Sender's set and Receiver's set.

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.

Parameters: Sender $\mathcal{S}$, Receiver $\mathcal{R}$; Set size m, n; Dimension d; Distance function $\mathsf{dist}(\cdot,\cdot)$, Distance threshold δ; Leakage function $\mathsf{leakage}(\cdot,\cdot)$; Label length σ.

Functionality:

- Wait an input $\mathbf{Q} \in \mathbb{U}^{d\times m}$ from $\mathcal{S}$.
  For LFPSI, wait another input $\mathsf{Label}_{\mathbf{Q}} \in \{0,1\}^{\sigma\times m}$ from $\mathcal{S}$.
- Wait an input $\mathbf{W} \in \mathbb{U}^{d\times n}$ from $\mathcal{R}$.
- Return $\mathsf{leakage}\left(\mathbf{Q},\mathbf{W}\right)$ to $\mathcal{R}$.

Leakage Functions: $\mathsf{leakage}(\mathbf{Q},\mathbf{W})$ is defined as:

- **FPSI:** $\mathsf{leakage}\left(\mathbf{Q},\mathbf{W}\right) = \{\mathbf{q}_j \mid \exists\ i \in [n], \mathsf{dist}\left(\mathbf{q}_j,\mathbf{w}_i\right) \le \delta\}$.
- **LFPSI:** $\mathsf{leakage}(\mathbf{Q},\mathbf{W}) = \{\mathsf{label}_j \mid \exists\ i \in [n], \mathsf{dist}(\mathbf{q}_j,\mathbf{w}_i) \le \delta\}$, where label_j is the label associated with $\mathbf{q}_j$.
- **FPSI-card:** $\mathsf{leakage}(\mathbf{Q},\mathbf{W}) = \sum_{j\in[m],i\in[n]}(\mathsf{dist}(\mathbf{q}_j,\mathbf{w}_i) \le \delta)$.
- **FPSI-SP:** $\mathsf{leakage}(\mathbf{Q},\mathbf{W}) = \{\mathbf{w}_i \mid \exists\ j \in [m], \mathsf{dist}(\mathbf{q}_j,\mathbf{w}_i) \le \delta\}$.

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

2.1 Challenge in Efficient FPSI

Most FPSI protocols [1,4,7,11,13,15,23,26], 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 $\mathsf{dist}(\mathbf{q}_j,\mathbf{w}_i) \le \delta$, 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 [4,7,11,15,23,26], 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[4]. 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 are derived from their coarse mapping methods. For example, the naive coarse mapping which brutally traverses all pairs of inputs from parties results in an unacceptable $m \cdot n$

[4] A point can appear in multiple pairs.

blowup in complexities. Besides, [13] 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 Sec 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 $\big(\mathsf{ID}(\mathbf{q}_j)\big)_{j\in[m]}$ for Sender and $\big(\mathsf{ID}(\mathbf{w}_i)\big)_{i\in[n]}$ for Receiver, where $\mathsf{ID}(\mathbf{q})$ and $\mathsf{ID}(\mathbf{w})$ are subsets of an identifier universe $\mathscr{I}$.

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

- $\mathsf{ID}(\mathbf{q}_j)$ should intersect with $\mathsf{ID}(\mathbf{w}_i)$ when $\mathsf{dist}(\mathbf{q}_j, \mathbf{w}_i)$ is not greater than δ. 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[5].
- The probability that there exist two distinct points $\mathbf{w}_i$ and $\mathbf{w}_{i'}$ in $\mathbf{W}$ such that $\mathsf{ID}(\mathbf{w}_i)$ intersects with $\mathsf{ID}(\mathbf{w}_{i'})$ 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[6].
- 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.

[5] That is to say, cases that $\mathsf{ID}(\mathbf{q}_j)$ intersects with $\mathsf{ID}(\mathbf{w}_i)$ and $\mathsf{dist}(\mathbf{q}_j, \mathbf{w}_i)$ is greater than δ are allowed.

[6] 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 [18, 19]. 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 Sec 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 [8].

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 Sec 5.1.

2.4 UFmap for L_∞ Distance

UFmap for L_∞ Distance is Enough. To overcome complexity bottlenecks in FPSI protocols for $L_{\mathsf{p}\in[1,\infty]}$ distance, we hanker for a UFmap for $L_{\mathsf{p}\in[1,\infty]}$ distance. Fortunately, benefiting from the facts that Fmap can tolerate false positives and that the L_∞ distance between any two points is always no greater than the $L_{\mathsf{p}\in[1,\infty]}$ distance, we can use Fmap for L_∞ distance in FPSI for general $L_{\mathsf{p}\in[1,\infty]}$ distance. Therefore, we will only discuss the construction of UFmap for L_∞ 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 $\mathsf{seed}_{\mathbf{w},\mathcal{R}}$ for $\mathbf{w} \in \mathbf{W}$ as an assignment and Sender wants to choose $\mathsf{seed}_{\mathbf{q},\mathcal{R}}$ as the assignment of $\mathbf{q} \in \mathbf{Q}$ meeting $\mathsf{seed}_{\mathbf{q},\mathcal{R}}$ equals to $\mathsf{seed}_{\mathbf{w},\mathcal{R}}$ when $L_\infty(\mathbf{q}, \mathbf{w})$ is not greater than δ.

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

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.

[7] 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 2δ 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_∞ distance, which we call SAS Fmap. Since its expansion rate is 1, SAS Fmap is capable of circumventing complexity bottlenecks in FPSI protocols for $L_{\mathsf{p}\in[1,\infty]}$ distance.

2.5 Applications of Fmap

mqFRPMT from sUFmap. Chen et al. [6] 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 $\mathbf{Q} = (\mathbf{q}_1, \cdots, \mathbf{q}_m)$ and Receiver holding $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_n)$. After invoking of mqFRPMT for distance $\mathsf{dist}(\cdot,\cdot)$ and threshold δ, Receiver learns an indication bit vector $\mathbf{e} = (e_1, \cdots, e_m) \in \{0,1\}^m$ such that e_i equals to 1 if and only if there exists a point $\mathbf{w}_j \in \mathbf{W}$ meeting $\mathsf{dist}(\mathbf{q}_i, \mathbf{w}_j)$ is not greater than δ, 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.

However, if the ultimate goal is to construct FPSI, this leakage will not affect the security[8]. Thus, any Fmap can be utilized to construct the corresponding FPSI using a generic method, while only sUFmap can directly yield mqFRPMT[9].

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. 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.

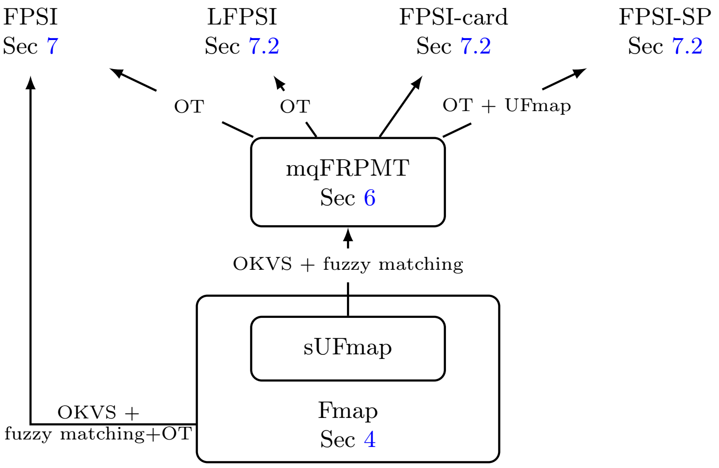

Fig. 2. Summary of our work. The rectangles denote notions newly in this work.

[8] Because FPSI allows Receiver to obtain information about these points.

[9] As will be shown later, using fuzzy matching that outputs secret shares can solve this problem.

3 Preliminaries

For lack of space, we put Additively Homomorphic Encryption, Oblivious Transfer, and Semi-Honest Security Model in the full version.

3.1 Notation

We use κ, λ to denote the computational and statistical security parameters respectively. We use $[n]$ to denote the set $\{1, 2, \cdots, n\}$ and $[n, m]$ to denote the set $\{n, n+1, \cdots, m\}$. We assume that every set X of size $|X|$ has a default order (e.g. lexicographical order), and represent it as $X = \left(x_1, \cdots, x_{|X|}\right) = (x_i)_{i \in [|X|]}$. We use $\leftarrow$ to denote assignment and $x \overset{\mathsf{R}}{\leftarrow} X$ to denote sampling x uniformly at random from X. A function is negligible in ℓ, written $\mathsf{negl}(\ell)$, if it vanishes faster than the inverse of any polynomial in ℓ. $\boldsymbol{x} \| \boldsymbol{y}$ is the concatenation of two strings $\boldsymbol{x}$ and $\boldsymbol{y}$. For a key-value pairs multiset List, we use $\mathsf{List}[k]$ to denote the value for key k.

For parameters in FPSI, we use d to denote the dimension of space, δ to denote the threshold, and $\mathsf{dist}(\cdot, \cdot)$ to denote the distance function. We use $\mathcal{H}$, L_∞, and L_p distance as Hamming, infinite norm, and Minkowski distance, respectively. To simplify the statement, we use $L_{\mathsf{p} \in [1, \infty]}$ to represent the union of $L_{\mathsf{p} \in [1, \infty)}$ and L_∞. 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, where d is the dimension. Here we use 2^u to denote 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. $\mathsf{ball}_{\mathbf{w}_i}^{\mathsf{dist}(\cdot,\cdot)}$ represents a d-dimensional ball with $\mathbf{w}_i$ as center and δ as radius. We use $\mathscr{I}$ to denote the identifier universe. $\mathsf{ID}\left(\mathbf{q}_j\right), \mathsf{ID}\left(\mathbf{w}_i\right) \subset \mathscr{I}$ 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 $\mathcal{H}_{\mathbb{U}}\left(\mathbf{q}, \mathbf{w}\right) = \mathcal{H}\left(\mathbf{q}, \mathbf{w}\right) = \sum_{k=1}^{d} \left(q_k \neq w_k\right)$, and Hamming distance over $\mathbb{U}^P$ is $\mathcal{H}_{\mathbb{U}^P}\left(\mathbf{q}, \mathbf{w}\right) = \sum_{k'=0}^{\frac{d}{P}-1} \left(\widetilde{q_{k'}} \neq \widetilde{w_{k'}}\right)$, where $\widetilde{q_{k'}} = q_{k' \cdot P+1} \| q_{k' \cdot P+2} \| \cdots \| q_{k' \cdot P+P}$ and $\widetilde{w_{k'}} = w_{k' \cdot P+1} \| w_{k' \cdot P+2} \| \cdots \| w_{k' \cdot P+P}$.

3.2 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 [12].

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. [2] prove that their RB-OKVS satisfies independence property [10].

[10] They call this property "random decoding".

The formal definitions of OKVS and its independence property are given in the full version.

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 [2, 12, 21] achieve constant rate, $\mathcal{O}(n\lambda)$ encoding cost, and $\mathcal{O}(\lambda)$ decoding cost.

3.3 Fuzzy Matching

Fuzzy matching enables Sender and Receiver determine whether the Sender's point $\mathbf{q}$ and the Receiver's point $\mathbf{w}$ satisfy $\mathsf{dist}(\mathbf{q}, \mathbf{w}) \leq \delta$ [1]. Its functionality is given in Fig.3. Obviously, let the protocol return the result by secret shares, and we get secret-shared fuzzy matching.

Our work is concerned with Hamming and $L_{\mathsf{p}\in[1,\infty]}$ 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 (secret-shared) fuzzy matching that consists of OT-based conversion of Boolean sharing to Arithmetic sharing and (secret-shared) secure comparing [20]. This approach has $\mathcal{O}(d)$ communication and computation costs. For $L_{\mathsf{p}\in[1,\infty]}$ distance, Baarsen and Pu provide constructions of fuzzy matching in [1].

PARAMETERS: Sender $\mathcal{S}$, Receiver $\mathcal{R}$; Dimension d; Distance function $\mathsf{dist}(\cdot, \cdot)$; Distance threshold δ.

FUNCTIONALITY:

- Wait an input $\mathbf{q} \in \mathbb{U}^d$ from $\mathcal{S}$.
- Wait an input $\mathbf{w} \in \mathbb{U}^d$ from $\mathcal{R}$.
- Return $e \in \{0, 1\}$ to $\mathcal{R}$, where $e = 1$ if and only if $\mathsf{dist}(\mathbf{q}, \mathbf{w}) \leq \delta$.

Fig. 3. Ideal Functionality for Fuzzy Matching $\mathcal{F}_{\mathsf{FMatch}}$

One of building blocks we use is a special case of fuzzy matching, 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 δ is set to 0, this special case of IFmat is private equality test (PEqT). Their functionalities is given in Fig.4.

Using the idea of prefix matching, Chakraborti et al. [4] 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.

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.

Parameters: Sender $\mathcal{S}$, Receiver $\mathcal{R}$; Threshold δ.

Functionality:

- Wait an input $a \in \mathbb{Z}$ from $\mathcal{S}$.
- Wait an input $x \in \mathbb{Z}$ from $\mathcal{R}$.
- Return e to $\mathcal{R}$, where $e = 1$ if and only if:
  IFmat: $x \in [a-\delta, a+\delta]$
  PEqT: $x = a$

Fig. 4. Ideal Functionalities for IFmat $\mathcal{F}_{\mathsf{IFmat}}$ and PEqT $\mathcal{F}_{\mathsf{PEqT}}$

4.1 Definition of Fmap

As mentioned in Sec 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 fuzzy matching to obtain FPSI result.

The formal definition of Fmap is as follows.

Definition 1 (Fuzzy Mapping). *A two-party protocol Π, where Sender's input $\mathbf{Q} = (\mathbf{q}_j)_{j\in[m]} \in \mathbb{U}^{d\times m}$ results in $\mathsf{ID}(\mathbf{Q}) = \left(\mathsf{ID}(\mathbf{q}_j)\right)_{j\in[m]}$ and Receiver's input $\mathbf{W} = (\mathbf{w}_i)_{i\in[n]} \in \mathbb{U}^{d\times n}$ results in $\mathsf{ID}(\mathbf{W}) = \left(\mathsf{ID}(\mathbf{w}_i)\right)_{i\in[n]}$[11], is a semi-honest secure* fuzzy mapping (Fmap) *protocol $\Pi^{\mathsf{dist}(\cdot,\cdot)}_{\mathsf{Fmap}}$ of threshold δ for $\mathsf{dist}(\cdot,\cdot)$, if and only if Π satisfies:*

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

$$\mathsf{dist}(\mathbf{q}_j, \mathbf{w}_i) \le \delta \Longrightarrow \mathsf{ID}(\mathbf{q}_j) \cap \mathsf{ID}(\mathbf{w}_i) \neq \emptyset$$

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

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

- **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*

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

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

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

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

[11] $\mathsf{ID}(\mathbf{q}_j), \mathsf{ID}(\mathbf{w}_i) \subset \mathscr{I}$; for security reason, we default to $|\mathsf{ID}(\mathbf{q}_j)| = |\mathsf{ID}(\mathbf{q}_{j'})|$ for different $j, j' \in [m]$ and $|\mathsf{ID}(\mathbf{w}_i)| = |\mathsf{ID}(\mathbf{w}_{i'})|$ for different $i, i' \in [n]$.

Definition 2 (Expansion Rate). *The expansion rate of Fmap for Sender's input is*

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

The expansion rate of Fmap for Receiver's input is

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

The expansion rate of Fmap is

$$\mathsf{rate} = \max\left\{\mathsf{rate}_{\mathcal{S}}, \mathsf{rate}_{\mathcal{R}}\right\}$$

Definition 3 (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 4 (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* $\mathsf{dist}\left(\cdot,\cdot\right)$ *and* $\mathsf{dist}'\left(\cdot,\cdot\right)$ *such that* $\mathsf{dist}\left(\mathbf{q},\mathbf{w}\right) \leq \mathsf{dist}'\left(\mathbf{q},\mathbf{w}\right)$ *holds for any two points* $\mathbf{q}$ *and* $\mathbf{w}$*, then Fmap protocol* $\Pi_{\mathsf{Fmap}}^{\mathsf{dist}(\cdot,\cdot)}$ *realizes Fmap for* $\mathsf{dist}'\left(\cdot,\cdot\right)$*.*

Proof. As Fmap for $\mathsf{dist}'\left(\cdot,\cdot\right)$, the distinctiveness and security of $\Pi_{\mathsf{Fmap}}^{\mathsf{dist}(\cdot,\cdot)}$ are guaranteed by Definition 1 of Fmap for $\mathsf{dist}\left(\cdot,\cdot\right)$.

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

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

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, $\Pi_{\mathsf{Fmap}}^{\mathcal{H}_{\mathbb{U}^P}}$ can be seen as $\Pi_{\mathsf{Fmap}}^{\mathcal{H}_{\mathbb{U}}}$.*

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

4.2 Existing Fmap Constructions

Table 2 lists existing constructions that fit to Definition 1 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 size of Sender's and Receiver's inputs. d is the space dimension. $\rho \in (0,1)$ is a parameter in LSH scheme. We ignore multiplicative factors of the computational security parameter κ and statistical parameter λ.

Fmap	Distance	$\mathsf{rate}_{\mathcal{S}}$	$\mathsf{rate}_{\mathcal{R}}$	Communication	Computation	
					Sender	Receiver
Naive [11]	**Anyone**	n	1	–	$\mathcal{O}(nm)$	$\mathcal{O}(n)$
Spatial Hashing [13]	$\boldsymbol{L_\infty}$	1	$\mathcal{O}(2^d)$	–	$\mathcal{O}(m)$	$\mathcal{O}(2^d n)$
Separated Balls [1]	$\boldsymbol{L_\infty}$	d	$\mathcal{O}(\delta)$	–	$\mathcal{O}(dm)$	$\mathcal{O}(\delta n)$
LSH [1]	$\boldsymbol{L_{\mathsf{p}\in[1,\infty)}}$	$\mathcal{O}(n^\rho \log n)$	$\mathcal{O}(n^\rho)$	–	$\mathcal{O}((n^\rho \log n) m)$	$\mathcal{O}(n^{\rho+1})$
Ours: UniqC	**Hamming**	d	$\delta + 1$	–	$\mathcal{O}(dm)$	$\mathcal{O}(\delta n)$
Ours: SAS	$\boldsymbol{L_\infty}$	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 fuzzy matching on all pairs of these two inputs to obtain results. This idea can be abstracted into a naive Fmap: for each Sender's point $\mathbf{q}_j$, $\mathsf{ID}(\mathbf{q}_j)$ is $\{i\}_{i\in[n]}$, thus $\mathsf{rate}_{\mathcal{S}}$ is $\mathcal{O}(n)$; for each Receiver's point $\mathbf{w}_i$, $\mathsf{ID}(\mathbf{w}_i)$ is $\{i\}$ where i is the index of this point, thus $\mathsf{rate}_{\mathcal{R}}$ is 1. It does not rely on any assumptions and can be used in FPSI for any distance function. Many existing FPSI protocols [4,7,11,15,23,26] for Hamming distance adopt naive Fmap, which leads to the $m \cdot n$ blowup in their complexities.
- **Prior Non-trivial Fmap.** Recently, some works [1,13] try to avoid the $m \cdot n$ blowup. We abstracted three non-trivial Fmap from them, and the detailed analysis can be found in the full version.
- **New Fmap.** Details of our Fmap instance will be given later.

5 New Fmap Constructions

In this section, we present new semi-honest secure Fmap constructions for Hamming and L_∞ 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.5.

Parameters:

- Sender $\mathcal{S}$ and Receiver $\mathcal{R}$.

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

$\mathcal{S}$'s $\mathsf{UniqC\ Fmap}\,(\mathbf{Q})$:

1. For each $j \in [m]$, $\mathcal{S}$ computes $\mathsf{ID}\,(\mathbf{q}_j) \leftarrow \{k \| q_{j,k}\}_{k\in[d]}$.
2. Return $\mathsf{ID}\,(\mathbf{Q}) = (\mathsf{ID}\,(\mathbf{q}_1), \cdots, \mathsf{ID}\,(\mathbf{q}_m))$.

$\mathcal{R}$'s $\mathsf{UniqC\ Fmap}\,(\mathbf{W})$:

1. For each $i \in [n]$, $\mathcal{R}$ computes $\mathsf{ID}\,(\mathbf{w}_i) \leftarrow \left\{u_k^i \| w_{i,u_k^i}\right\}_{k\in[\delta+1]}$, where $u_k^i \in [d]$ is a dimension such that $w_{i,u_k^i} \neq w_{i',u_k^i}$ holds for any $i' \in [n] \setminus \{i\}$.
2. Return $\mathsf{ID}\,(\mathbf{W}) = (\mathsf{ID}\,(\mathbf{w}_1), \cdots, \mathsf{ID}\,(\mathbf{w}_n))$.

Fig. 5. Fmap Protocol for Hamming distance: $\Pi^{\mathcal{H}}_{\mathsf{UniqC\ Fmap}}$

Definition 5 (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* $1 - \mathsf{negl}(d)$[12].

[12] 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 $\left|\mathbb{U}^P\right| = 2^{uP} > n$, thus R. UniqC assumption for $\mathcal{H}_{\mathbb{U}^P}$ holds and $\Pi^{\mathcal{H}_{\mathbb{U}^P}}_{\mathsf{UniqC\ Fmap}}$ works. According to Corollary 1, we can use $\Pi^{\mathcal{H}_{\mathbb{U}^P}}_{\mathsf{UniqC\ Fmap}}$ as $\Pi^{\mathcal{H}_{\mathbb{U}}}_{\mathsf{UniqC\ Fmap}}$.

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 δ 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

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

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

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

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

Remark 2. Considering 400-dimensional bio-bit-vectors and $\delta = 7$ in [24], 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:

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.5 is a semi-honest secure Fmap for Hamming distance.

Theorem 1 (Correctness). *The protocol presented in Fig.5 satisfies the correctness defined in Definition 1 for Hamming distance.*

Proof. 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 $\mathsf{ID}(\mathbf{q}_j) \cap \mathsf{ID}(\mathbf{w}_i) \neq \emptyset$.

Theorem 2 (Distinctiveness). *The protocol presented in Fig.5 satisfies the distinctiveness defined in Definition 1.*

Proof. The distinctiveness comes from Definition 5 of unique component.

Theorem 3 (Security). *The protocol presented in Fig.5 satisfies the security defined in Definition 1.*

Proof. 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_∞ Distance

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

As described in Sec 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.6.

```
Parameters:
  – Input size $m$.
  – Space dimension $d$.
  – Threshold $\delta$.
  – A finite group $\mathbb{G}$.
Assignment $\left(\mathbf{Q} = (\mathbf{q}_1, \cdots, \mathbf{q}_m) \in \mathbb{U}^{d \times m}\right)$:
  1. Initialize key-value pairs multiset $\mathsf{mList}_{\mathcal{S}} \leftarrow \emptyset$.
  2. For each $j \in [m]$ and each $k \in [d]$:
       Sample $\mathsf{Rand}_{j,k} \overset{\mathsf{R}}{\leftarrow} \mathbb{G}$.
       For each $t \in [-\delta, \delta]$:
         Update $\mathsf{mList}_{\mathcal{S}} \leftarrow \mathsf{mList}_{\mathcal{S}} \cup \{(k \| (q_{j,k} + t), \mathsf{Rand}_{j,k})\}$.
         While there exists assigned $k \| (q_{j,k} + t)$ with other value in $\mathsf{mList}_{\mathcal{S}}$:
           Update values of its $2\delta + 1$ adjacent points with $\mathsf{Rand}_{j,k}$.
  3. Remove duplicates in $\mathsf{mList}_{\mathcal{S}}$.
  4. Pad $\mathsf{mList}_{\mathcal{S}}$ with dummy random elements to get $\mathsf{List}_{\mathcal{S}}$ of size $(2\delta + 1)dm$.
  5. For each $j \in [m]$:
       Set $\mathsf{Seed}_{\mathbf{q}_j, \mathcal{S}} \leftarrow \sum_{k \in [d]} \mathsf{List}_{\mathcal{S}}[k \| q_{j,k}]$.
  6. Return $\mathsf{List}_{\mathcal{S}}$ and $\left(\mathsf{Seed}_{\mathbf{q}_j, \mathcal{S}}\right)_{j \in [m]}$.
```

Fig. 6. Assignment Algorithm: $\mathsf{Assignment}(\cdot)$

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_∞ 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.7.

Definition 6 (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* 2δ*; otherwise,* $\mathbf{q}$ *is* separated *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 anyone of 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* $1 - \mathsf{negl}(d)$*.*

For SAS Fmap, we introduce the $\mathsf{R} \wedge \mathsf{S}$. 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.7 is 1. Now, we prove our protocol is a semi-honest secure Fmap for L_∞ distance.

Theorem 4 (Correctness). *The protocol presented in Fig.7 satisfies the correctness defined in Definition 1 for* L_∞ *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 $\mathsf{List}_{\mathcal{R}}$. According to correctness of OKVS, $\mathcal{S}$ gets ElGamal ciphertexts of $\mathsf{List}_{\mathcal{R}}[k \| q_{j,k}]$ 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 $\mathsf{List}_{\mathcal{R}}[k \| q_{j,k}] = \mathsf{List}_{\mathcal{R}}[k \| w_{i,k}]$ for $k \in [d]$. Therefore, $\mathsf{sum}_{\mathbf{q}_j}^{\mathsf{pk}_{\mathsf{ElG},\mathcal{R}}}$ is the ElGamal ciphertext of $\mathsf{Seed}_{\mathbf{w}_i,\mathcal{R}}$. Then, it is clear that

$$\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j} = \mathsf{sk}_{\mathsf{DH},\mathcal{R}} \cdot \mathsf{mask}_{\mathcal{S},j} \cdot \left(\mathsf{Seed}_{\mathbf{q}_i,\mathcal{S}} + \mathsf{Seed}_{\mathbf{w}_j,\mathcal{R}}\right)$$

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

Similarly, we can find that

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

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

PARAMETERS:

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

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

PROTOCOL:

Phase 1. Assignment and Summation:

1. $\mathcal{S}$ computes $\left(\mathsf{List}_{\mathcal{S}}, \left(\mathsf{Seed}_{\mathbf{q}_j,\mathcal{S}}\right)_{j\in[m]}\right) \leftarrow \mathsf{Assignment}\left(\mathbf{Q}\right)$.
2. $\mathcal{S}$ generates $\left(\mathsf{sk}_{\mathsf{EIG},\mathcal{S}}, \mathsf{pk}_{\mathsf{EIG},\mathcal{S}}\right) \leftarrow \mathsf{Gen}^{\mathsf{EIG}}\left(1^{\kappa}\right)$. $\mathcal{S}$ samples m masks $\left\{\mathsf{mask}_{\mathcal{S},j} \xleftarrow{\mathsf{R}} \mathbb{F}_p\right\}_{j\in[m]}$ and computes their inverses $\left\{\mathsf{mask}_{\mathcal{S},j}^{-1}\right\}_{j\in[m]}$.
3. $\mathcal{S}$ initializes set $\widetilde{\mathsf{List}_{\mathcal{S}}} \leftarrow \emptyset$. For each $(\mathsf{key}, \mathsf{List}_{\mathcal{S}}[\mathsf{key}]) \in \mathsf{List}_{\mathcal{S}}$, $\mathcal{S}$ updates
$$\widetilde{\mathsf{List}_{\mathcal{S}}} \leftarrow \widetilde{\mathsf{List}_{\mathcal{S}}} \cup \left\{\left(\mathsf{key}, \mathsf{Enc}^{\mathsf{EIG}}_{\mathsf{pk}_{\mathsf{EIG},\mathcal{S}}}\left(\mathsf{List}_{\mathcal{S}}\left[\mathsf{key}\right]\right)\right)\right\}$$
.
4. $\mathcal{S}$ encodes $E_{\mathcal{S}} \leftarrow \mathsf{Encode}\left(\widetilde{\mathsf{List}_{\mathcal{S}}}, r\right)$, and sends $E_{\mathcal{S}}, \mathsf{pk}_{\mathsf{EIG},\mathcal{S}}$ to $\mathcal{R}$.
5. Symmetrically, $\mathcal{R}$ sends $E_{\mathcal{R}} \leftarrow \mathsf{Encode}\left(\widetilde{\mathsf{List}_{\mathcal{R}}}, r\right), \mathsf{pk}_{\mathsf{EIG},\mathcal{R}}$ to $\mathcal{S}$.
6. For each $j \in [m]$, $\mathcal{S}$ computes
$$\mathsf{sum}_{\mathbf{q}_j}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}} \leftarrow \bigoplus_{\substack{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}} \\ k\in[d]}}^{\mathsf{EIG}} \mathsf{Decode}\left(E_{\mathcal{R}}, k\|q_{j,k}, r\right)$$
$$\mathsf{cipher}_{\mathbf{q}_j}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}} \leftarrow \mathsf{mask}_{\mathcal{S},j} \times \left(\left(\mathsf{Enc}^{\mathsf{EIG}}_{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}}\left(\mathsf{Seed}_{\mathbf{q}_j,\mathcal{S}}\right)\right) \oplus^{\mathsf{EIG}}_{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}} \mathsf{sum}_{\mathbf{q}_j}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}}\right)$$

Phase 2. Decryption and Exchange:

7. $\mathcal{S}$ sends m ElGamal ciphertexts $\left(\mathsf{cipher}_{\mathbf{q}_j}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{R}}}\right)_{j\in[m]}$ to $\mathcal{R}$.
8. Symmetrically, $\mathcal{R}$ sends n ElGamal ciphertexts $\left(\mathsf{cipher}_{\mathbf{w}_i}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{S}}}\right)_{i\in[n]}$ to $\mathcal{S}$.
9. $\mathcal{S}$ samples $\mathsf{sk}_{\mathsf{DH},\mathcal{S}} \xleftarrow{\mathsf{R}} \mathbb{F}_p$. For each $i \in [n]$, $\mathcal{S}$ computes
$$\mathsf{Seed}_{\mathsf{mk}_{\mathcal{R},i}}^{\mathbf{w}_i} \leftarrow \mathsf{Dec}^{\mathsf{EIG}}_{\mathsf{sk}_{\mathsf{EIG},\mathcal{S}}}\left(\mathsf{cipher}_{\mathbf{w}_i}^{\mathsf{pk}_{\mathsf{EIG},\mathcal{S}}}\right)$$
$$\mathsf{Seed}_{\mathsf{mk}_{\mathcal{R},i},\mathcal{S}}^{\mathbf{w}_i} \leftarrow \mathsf{sk}_{\mathsf{DH},\mathcal{S}} \cdot \mathsf{Seed}_{\mathsf{mk}_{\mathcal{R},i}}^{\mathbf{w}_i}$$
10. $\mathcal{S}$ sends $\left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{R},i},\mathcal{S}}^{\mathbf{w}_i}\right)_{i\in[n]}$ to $\mathcal{R}$. Symmetrically, $\mathcal{R}$ sends $\left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}\right)_{j\in[m]}$.
11. For each $i \in [m]$, $\mathcal{S}$ computes $\left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R},\mathcal{S}}^{\mathbf{q}_j}\right) \leftarrow \mathsf{sk}_{\mathsf{DH},\mathcal{S}} \cdot \left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}^{\mathbf{q}_j}\right)$, and unmasks $\left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R},\mathcal{S}}^{\mathbf{q}_j}\right)$ to get $\mathsf{id}_{\mathbf{q}_j} \leftarrow \mathsf{mask}_{\mathcal{S},j}^{-1} \cdot \left(\mathsf{Seed}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R},\mathcal{S}}^{\mathbf{q}_j}\right)$.
12. Symmetrically, $\mathcal{R}$ gets $(\mathsf{id}_{\mathbf{w}_i})_{i\in[n]}$.

Fig. 7. UFmap Protocol for L_∞ Distance Without Expansion: $\Pi^{L_\infty}_{\mathsf{SAS\ Fmap}}$

Theorem 5 (Distinctiveness). *The protocol presented in Fig.7 satisfies the distinctiveness defined in Definition 1.*

Proof. First consider the side of $\mathcal{S}$. For $j \in [m]$, let us assume that $\mathbf{q}_j$ is separated from the others in $\mathbf{W}$ on dimension k_j. With reference to proof of Theorem 4, we have

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

Under $\mathsf{R} \wedge \mathsf{S}$. 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, $\mathsf{List}_{\mathcal{S}}\left[k_j \| q_{j,k_j}\right]$ is a uniformly random value independent of any other assignments. So, the probability that $\mathsf{id}_{\mathbf{q}_j}$ equals some $\mathsf{id}_{\mathbf{q}_{j'}}$ is $\frac{m-1}{|\mathbb{G}|}$. Hence, it holds that

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

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

Theorem 6 (Randomness). *In the protocol presented in Fig.7,* $\left(\mathsf{id}_{\mathbf{q}_j}\right)_{j\in[m]}$ *is computationally indistinguishable from* $\mathsf{R}_{\mathsf{id}} \xleftarrow{\mathsf{R}} \mathbb{G}^m$*, and* $\left(\mathsf{id}_{\mathbf{w}_i}\right)_{i\in[n]}$ *is computationally indistinguishable from* $\mathsf{R}_{\mathsf{id}} \xleftarrow{\mathsf{R}} \mathbb{G}^n$*, if the DDH assumption holds.*

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

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

According to DDH assumption, $\left(\mathsf{sk}_{\mathsf{DH},\mathcal{R}} \cdot \mathsf{Seed}_{\mathbf{q}_j,\mathcal{S}}\right)_{j\in[m]}$ 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, $\mathsf{Seed}_{\mathbf{q}_j,\mathcal{R}}$ is independent of $\mathsf{Seed}_{\mathbf{q}_j,\mathcal{S}}$. In conclusion, $\left(\mathsf{id}_{\mathbf{q}_j}\right)_{j\in[m]}$ is computationally indistinguishable from $\mathsf{R}_{\mathsf{id}} \xleftarrow{\mathsf{R}} \mathbb{G}^m$.

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

Theorem 7 (Security). *The protocol presented in Fig.7 satisfies the security defined in Definition 1 if the DDH assumption holds.*

Proof. First consider the side of $\mathcal{S}$. We exhibit simulator $\mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}\left(\mathbf{Q}, \mathsf{id}\left(\mathbf{Q}\right)_{\mathbf{W}}\right)$ for simulating corrupt $\mathcal{S}$ where $\mathsf{id}\left(\mathbf{Q}\right)_{\mathbf{W}}$ is the output of $\mathcal{S}$ holding $\mathbf{Q}$ who invokes the protocol presented in Fig.7 with $\mathcal{R}$ holding $\mathbf{W}$. And we argue the indistinguishability of the produced transcript from the real execution.

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

1. $\mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}$ generates $\left(\overline{\mathsf{sk}_{\mathsf{EIG},\mathcal{S}}}, \overline{\mathsf{pk}_{\mathsf{EIG},\mathcal{S}}}\right) \leftarrow \mathsf{Gen}^{\mathsf{EIG}}\left(1^\kappa\right)$, samples $\left\{\overline{\mathsf{mask}_{\mathcal{S},j}} \xleftarrow{\mathsf{R}} \mathbb{F}_p\right\}_{j\in[m]}$, computes $\left\{\overline{\mathsf{mask}_{\mathcal{S},j}}^{-1}\right\}_{j\in[m]}$, and appends them to the view.

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

Now we show that the view output by $\mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}$ 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 $\mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}$.

- $\mathbf{Hyb}_0$**.** This hybrid is the real interaction described in Fig.7. Let T_0 denote $\mathcal{S}$'s view in the real protocol.
- $\mathbf{Hyb}_1$**.** Let T_1 be the same as T_0, except that OKVS $E_{\mathcal{R}}$ and $\mathsf{pk}_{\mathsf{ElG},\mathcal{R}}$ are replaced by $\overline{E_{\mathcal{R}}}$ and $\overline{\mathsf{pk}_{\mathsf{ElG},\mathcal{R}}}$. The values for encoding $E_{\mathcal{R}}$ are $(2\delta{+}1)dn$ ciphertexts encrypted with $\mathsf{pk}_{\mathsf{ElG},\mathcal{R}}$, 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 $T_1 \overset{c}{\approx} T_0$.
- $\mathbf{Hyb}_2$**.** Let T_2 be the same as T_1, except that $\left(\mathsf{sk}_{\mathsf{ElG},\mathcal{S}}, \mathsf{pk}_{\mathsf{ElG},\mathcal{S}}\right)$ and $\left(\mathsf{cipher}_{\mathbf{w}_i}^{\mathsf{pk}_{\mathsf{ElG},\mathcal{S}}}\right)_{i\in[n]}$ are replaced by $\left(\overline{\mathsf{sk}_{\mathsf{ElG},\mathcal{S}}}, \overline{\mathsf{pk}_{\mathsf{ElG},\mathcal{S}}}\right)$ and $\left(\mathsf{cipher}_i^{\overline{\mathsf{pk}_{\mathsf{ElG},\mathcal{S}}}}\right)_{i\in[n]}$. Since each $\mathsf{Seed}^{\mathbf{w}_i}_{\mathsf{mk}_{\mathcal{R},i}}$ is masked by uniformly random $\mathsf{mask}_{\mathcal{R},i}$ from $\mathcal{R}$, $\left(\mathsf{Seed}^{\mathbf{w}_i}_{\mathsf{mk}_{\mathcal{R},i}}\right)_{i\in[n]}$ are statistically indistinguishable from $\left(\overline{\mathsf{Seed}_i} \xleftarrow{\mathsf{R}} \mathbb{G}\right)_{i\in[n]}$. Therefore, $\left(\mathsf{cipher}_{\mathbf{w}_i}^{\mathsf{pk}_{\mathsf{ElG},\mathcal{S}}}\right)_{i\in[n]}$ and $\left(\mathsf{cipher}_i^{\overline{\mathsf{pk}_{\mathsf{ElG},\mathcal{S}}}}\right)_{i\in[n]}$ are statistically indistinguishable, which means $T_2 \overset{s}{\approx} T_1$.
- $\mathbf{Hyb}_3$**.** Let T_3 be the same as T_2, except that $\left\{\mathsf{mask}_{\mathcal{S},j}\right\}_{j\in[m]}$, $\left\{\mathsf{mask}_{\mathcal{S},j}{}^{-1}\right\}_{j\in[m]}$, $\mathsf{sk}_{\mathsf{DH},\mathcal{S}}$, and $\left(\mathsf{Seed}^{\mathbf{q}_j}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}\right)_{j\in[m]}$ are replaced by $\left\{\overline{\mathsf{mask}_{\mathcal{S},j}}\right\}_{j\in[m]}$, $\left\{\overline{\mathsf{mask}_{\mathcal{S},j}}^{-1}\right\}_{j\in[m]}$, $\overline{\mathsf{sk}_{\mathsf{DH},\mathcal{S}}}$, and $\left(\overline{\mathsf{Seed}^{\mathbf{q}_j}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}}\right)_{j\in[m]}$. It is clear that masks $\left\{\mathsf{mask}_{\mathcal{S},j}\right\}_{j\in[m]}$ and $\left\{\overline{\mathsf{mask}_{\mathcal{S},j}}\right\}_{j\in[m]}$ are distributed identically; $\mathsf{sk}_{\mathsf{DH},\mathcal{S}}$ and $\overline{\mathsf{sk}_{\mathsf{DH},\mathcal{S}}}$ are distributed identically. Hence, $\left(\mathsf{Seed}^{\mathbf{q}_j}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}\right)_{j\in[m]}$ and $\left(\overline{\mathsf{Seed}^{\mathbf{q}_j}_{\mathsf{mk}_{\mathcal{S},j},\mathcal{R}}}\right)_{j\in[m]}$ are statistically indistinguishable. Thus $T_3 \overset{s}{\approx} T_2$ holds.

From the argument above, it holds that

$$\mathsf{view}_{\mathcal{R}}^{\Pi_{\mathsf{SAS}}\ \mathsf{Fmap}}\left(\kappa, \lambda; \mathbf{Q}, \mathbf{W}\right) \overset{c}{\approx} \mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}\left(\mathbf{Q}, \mathsf{id}\left(\mathbf{Q}\right)_{\mathbf{W}}\right)$$

In addition, according to Theorem 6, we have

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

Therefore, it comes that

$$\mathsf{view}^{\Pi_{\mathsf{SAS}}\ \mathsf{Fmap}}_{\mathcal{R}}\left(\kappa, \lambda; \mathbf{Q}, \mathbf{W}\right) \overset{c}{\approx} \mathsf{Sim}^{\mathcal{S}}_{\mathsf{Fmap}}\left(\mathbf{Q}, \mathsf{R}_{\mathsf{id}} \xleftarrow{\mathsf{R}} \mathbb{G}^{m}\right) \overset{c}{\approx} \mathsf{view}^{\Pi_{\mathsf{SAS}}\ \mathsf{Fmap}}_{\mathcal{R}}\left(\kappa, \lambda; \mathbf{Q}, \mathbf{W}'\right)$$

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

Remark 3 (Complexity). The protocol presented in Fig.7 has communication complexity $\mathcal{O}\left(\left((2\delta+1)d\kappa+2\kappa+\kappa\right)(m+n)\right)$, computation complexity $\mathcal{O}\left((2\delta+1)dm+n\right)$ for $\mathcal{S}$, and computation complexity $\mathcal{O}\left((2\delta+1)dn+m\right)$ for $\mathcal{R}$, if the OKVS has a constant rate, linear encoding time, and constant decoding time.

6 Multi-Query Fuzzy RPMT Based on sUFmap

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

6.1 Definition of mqFRPMT

mqFRPMT is the fuzzy version of mqRPMT, and we define the ideal functionality for mqFRPMT in Fig.8. 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 $\mathsf{dist}(\cdot,\cdot)$; Distance threshold δ.

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 $\mathbf{e} = (e_1, \cdots, e_m) \in \{0,1\}^m$ to $\mathcal{R}$, where $e_j = 1$ if and only if there exists $\mathbf{w}_i \in \mathbf{W}$ such that $\mathsf{dist}\left(\mathbf{q}_j, \mathbf{w}_i\right) \le \delta$.

Fig. 8. Ideal Functionality for Multi-Query Fuzzy RPMT $\mathcal{F}_{\mathsf{mqFRPMT}}$

6.2 mqFRPMT for L_∞ Distance from sUFmap

The high-level idea of sUFmap-based mqFRPMT is as described in Sec 2.5. In mqFRPMT for L_∞ distance, we instantiate fuzzy matching with an idea similar to [1]. We give the detailed mqFRPMT protocol for L_∞ distance in Fig.9.

We provide the proofs of correctness and security in the full version.

PARAMETERS:

- Sender $\mathcal{S}$ and Receiver $\mathcal{R}$.
- Space dimension d and threshold δ.
- An AHE scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \oplus)$.
- An OKVS scheme $(\mathsf{Encode}, \mathsf{Decode})$ and its random value r.

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

PROTOCOL:

1. $\mathcal{S}$ and $\mathcal{R}$ invoke $\Pi^{L_\infty}_{\mathsf{sUFmap}}$: $\mathcal{S}$ acts as Sender with input $\mathbf{Q}$ and $\mathcal{R}$ acts as Receiver with input $\mathbf{W}$. $\mathcal{S}$ receives $\mathsf{ID}(\mathbf{Q})$ and $\mathcal{R}$ receives $\mathsf{ID}(\mathbf{W})$.
2. $\mathcal{R}$ generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\kappa)$. $\mathcal{R}$ initializes set $\mathsf{List} \leftarrow \emptyset$.
3. For each $i \in [n]$, each $k \in [d]$, and each $t \in [-\delta, \delta]$, $\mathcal{R}$ update
$$\mathsf{List} \leftarrow \mathsf{List} \cup \{(\mathsf{id}_{\mathbf{w}_i}\|k\|(w_{i,k}+t), \mathsf{Enc}_{\mathsf{pk}}(0))\}_{\mathsf{id}_{\mathbf{w}_i}\in\mathsf{ID}(\mathbf{w}_i)}$$
4. $\mathcal{R}$ encodes $E \leftarrow \mathsf{Encode}(\mathsf{List}, r)$, and sends E, pk to $\mathcal{S}$.
5. For each $j \in [m]$, $\mathcal{S}$ samples $\mathsf{mask}_j \xleftarrow{\mathsf{R}} \mathcal{P}$ and computes
$$\mathsf{cipher}_j \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mathsf{mask}_j) \oplus_{\mathsf{pk}} \left(\bigoplus_{k\in[d]}{}_{\mathsf{pk}} \mathsf{Decode}\left(E, \mathsf{id}_{\mathbf{q}_j}\|k\|q_{j,k}, r\right)\right)$$
Then, $\mathcal{S}$ sends $\left(\mathsf{cipher}_j\right)_{j\in[m]}$ to $\mathcal{R}$.
6. $\mathcal{R}$ computes $\left(v_j \leftarrow \mathsf{Dec}_{\mathsf{sk}}\left(\mathsf{cipher}_j\right)\right)_{j\in[m]}$.
7. For each $j \in [m]$, $\mathcal{S}$ and $\mathcal{R}$ invoke $\mathcal{F}_{\mathsf{PEqT}}$: $\mathcal{S}$ acts as Sender with input mask_j and $\mathcal{R}$ acts as Receiver with input v_j. $\mathcal{S}$ receives nothing and $\mathcal{R}$ receives e_j.
8. $\mathcal{R}$ learns $\mathbf{e} = (e_1, \cdots, e_m)$.

Fig. 9. mqFRPMT for L_∞ from sUFmap: $\Pi^{L_\infty}_{\mathsf{mqFRPMT}}$

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

Theorem 9 (Security). *The protocol presented in Fig.9 realizes the functionality $\mathcal{F}_{\mathsf{mqFRPMT}}$ defined in Fig.8 for L_∞ distance against semi-honest adversaries in the $\mathcal{F}_{\mathsf{PEqT}}$-hybrid model if $\mathcal{E}$ satisfies IND-CPA security.*

Remark 4 (Complexity). The protocol presented in Fig.9 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.7.

6.3 mqFRPMT for $L_\mathsf{p} \in [1, \infty)$ Distance from sUFmap

The construction of mqFRPMT for $L_{\mathsf{p}\in[1,\infty)}$ distance is similar to $\Pi^{L_\infty}_{\mathsf{mqFRPMT}}$. For computing $L_{\mathsf{p}\in[1,\infty)}$ distance, in OKVS encoding, AHE ciphertexts of $|t|^\mathsf{p}$

instead of 0 are used as values of $\mathsf{id}_{\mathbf{w}_i} \| (w_{i,k} + t)$. 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 δ^{p} with Sender to learn final result of mqFRPMT. We give the detailed protocol and relevant proofs in the full version.

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 fuzzy matching 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 theu translation invariance property.

Definition 7 (Translation Invariance). *A distance function* $\mathsf{dist}(\cdot,\cdot)$ *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*

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

It is not difficult to see that Hamming and $L_{\mathsf{p}\in[1,\infty]}$ distances both have translation invariance property. We provide the detailed generic construction from Fmap to FPSI for distance with translation invariance in Fig.10. Thus, this generic construction is a powerful tool for FPSI for Hamming and $L_{\mathsf{p}\in[1,\infty]}$ distances. Specifically, we can instantiate the Fmap and fuzzy matching in the construction in Fig.10 with UniqC Fmap and trivial fuzzy matching for Hamming distance in Sec 3.3 to obtain an FPSI for Hamming distance.

We provide the proofs of correctness and security in the full version.

Theorem 10 (Correctness). *The protocol presented in Fig.10 realizes the functionality $\mathcal{F}_{\mathsf{FPSI}}$ defined in Fig.1 for distance with translation invariance correctly.*

Theorem 11 (Security). *The protocol presented in Fig.10 realizes the functionality $\mathcal{F}_{\mathsf{FPSI}}$ defined in Fig.1 for distance with translation invariance against semi-honest adversaries in the $(\mathcal{F}_{\mathsf{ssFMatch}}, \mathcal{F}_{\mathsf{PEqT}}, \mathcal{F}_{\mathsf{OT}})$-hybrid model, if $\mathcal{E}$ satisfies IND-CPA security.*

Remark 5 (Costs Analysis). The communication cost of protocol presented in Fig.10 consists of: communication cost of Fmap, sending OKVS from $n \cdot \mathsf{rate}_{\mathcal{R}}$ pairs, sending $m \cdot \mathsf{rate}_{\mathcal{S}}$ masked ciphers of points, communication cost of $m \cdot \mathsf{rate}_{\mathcal{S}}$ fuzzy matching, and communication cost of m OTs.

Parameters:

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

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

Protocol:

0. $\mathcal{S}$ permutates $\mathbf{Q}$ randomly.
1. $\mathcal{S}$ and $\mathcal{R}$ invoke $\Pi_{\mathsf{Fmap}}^{\mathsf{dist}(\cdot,\cdot)}$: $\mathcal{S}$ acts as Sender with input $\mathbf{Q}$ and $\mathcal{R}$ acts as Receiver with input $\mathbf{W}$. $\mathcal{S}$ receives $\mathsf{ID}(\mathbf{Q})$ and permutates each $\mathsf{ID}(\mathbf{q}_j)$ randomly; $\mathcal{R}$ receives $\mathsf{ID}(\mathbf{W})$.
2. $\mathcal{R}$ generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\kappa)$. $\mathcal{R}$ initializes set $\mathsf{List} \leftarrow \emptyset$.
3. For each $i \in [n]$, $\mathcal{R}$ update
$$\mathsf{List} \leftarrow \mathsf{List} \cup \left\{ \left(\mathsf{id}_{\mathbf{w}_i}, \left(\mathsf{Enc}_{\mathsf{pk}}(w_{i,1}), \cdots, \mathsf{Enc}_{\mathsf{pk}}(w_{i,d})\right)\right) \right\}_{\mathsf{id}_{\mathbf{w}_i} \in \mathsf{ID}(\mathbf{w}_i)}$$
4. $\mathcal{R}$ encodes $E \leftarrow \mathsf{Encode}(\mathsf{List}, r)$, and sends E, pk to $\mathcal{S}$.
5. For each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$, $\mathcal{S}$ samples $\mathbf{mask}_{j,\ell} = (\mathsf{mask}_{j,\ell,k})_{k\in[d]} \xleftarrow{\mathsf{R}} \mathbb{U}^d$ and computes
$$\mathbf{cipher}_{j,\ell} \leftarrow \left(\mathsf{Enc}_{\mathsf{pk}}(\mathsf{mask}_{j,\ell,k}) \oplus_{\mathsf{pk}} \mathsf{Decode}\left(E, \mathsf{id}_{\mathbf{q}_j,\ell}, r\right)_k\right)_{k\in[d]}$$
$$\mathbf{u}_{j,\ell} \leftarrow \mathbf{mask}_{j,\ell} + \mathbf{q}_j$$
Then, $\mathcal{S}$ sends $\left(\mathbf{cipher}_{j,\ell}\right)_{j\in[m],\ell\in[\mathsf{rate}_{\mathcal{S}}]}$ to $\mathcal{R}$.
6. $\mathcal{R}$ computes $\left(\mathbf{v}_{j,\ell} \leftarrow \mathsf{Dec}_{\mathsf{sk}}\left(\mathbf{cipher}_{j,\ell}\right)\right)_{j\in[m],\ell\in[\mathsf{rate}_{\mathcal{S}}]}$.
7. For each $j \in [m]$ and each $\ell \in [\mathsf{rate}_{\mathcal{S}}]$, $\mathcal{S}$ and $\mathcal{R}$ invoke $\mathcal{F}_{\mathsf{ssFMatch}}^{\mathsf{dist}(\cdot,\cdot)}$: $\mathcal{S}$ 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 $\mathcal{F}_{\mathsf{PEqT}}$: $\mathcal{S}$ acts as Sender with input $(e_{\mathcal{S},j,\ell})_{\ell\in[\mathsf{rate}_{\mathcal{S}}]}$ and $\mathcal{R}$ acts as Receiver with input $(e_{\mathcal{R},j,\ell})_{\ell\in[\mathsf{rate}_{\mathcal{S}}]}$. $\mathcal{S}$ receives nothing and $\mathcal{R}$ receives $1 - e_j$. $\mathcal{R}$ learns $\mathbf{e} = (e_1, \cdots, e_m)$.
9. For each $j \in [m]$, $\mathcal{S}$ and $\mathcal{R}$ invoke $\mathcal{F}_{\mathsf{OT}}$: $\mathcal{S}$ 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 $\mathbf{I}_{\mathsf{fuzzy}} \leftarrow \left\{m_{e_j} | (j \in [m]) \wedge (e_j = 1)\right\}$ as output.

Fig. 10. FPSI for $\mathsf{dist}(\cdot,\cdot)$ with translation invariance from Fmap: $\Pi_{\mathsf{FPSI}}^{\mathsf{dist}(\cdot,\cdot)}$

For $\mathcal{S}$, the computation cost of this protocol consists of: computation cost of Fmap as Sender, $m \cdot \mathsf{rate}_{\mathcal{S}}$ decoding of OKVS, $m \cdot \mathsf{rate}_{\mathcal{S}}$ homomorphic masking of points, computation cost of $m \cdot \mathsf{rate}_{\mathcal{S}}$ fuzzy matching as Sender, and computation cost of m OTs as Sender.

For $\mathcal{R}$, the computation cost of this protocol consists of: computation cost of Fmap as Receiver, $n \cdot \mathsf{rate}_{\mathcal{R}}$ encryptions of points, encoding of OKVS with $n \cdot \mathsf{rate}_{\mathcal{R}}$ pairs, $m \cdot \mathsf{rate}_{\mathcal{S}}$ decryptions of points, computation cost of $m \cdot \mathsf{rate}_{\mathcal{S}}$ fuzzy matching as Receiver, and computation cost of m OTs 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 Sec 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. For lack of space, we put the detailed method and proofs in the full version.

In Sec 6.2 and Sec 6.3, we present mqFRPMT protocols for L_∞ and $L_{\mathsf{p}\in[1,\infty)}$ distances respectively. Based on them, we can easily obtain FPSI for $L_{\mathsf{p}\in[1,\infty]}$ 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 and the data can be found in the full version.

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.

- OKVS: We use RB-OKVS in [2].
- OT: We use OT implementation in libOTe[13].
- Goldwasser-Micali: We use GMP[14] 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[15] with key size of 2048-bit as AHE in our FPSI for $L_{\mathsf{p}\in[1,\infty]}$ distance.
- Others: We use Curve25519 in cryptoTools[16] as the underlying group $\mathbb{G}$ for SAS Fmap. We adopt Coproto[17] to realize network communication.

8.2 Performance

FPSI for Hamming Distance. We compare our FPSI form UniqC Fmap for Hamming distance in Sec 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 [8]. Unfortunately, we do not have their code, thus we use their experimental results directly from their paper [8] 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 [8] in all cases. Note that our protocol achieves a 4.6× 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 $P = 16$ bits as one super-component.

$m = n$	Protocol	Comm. (MB)	Time (s)
256	[8]	465.68	38.7
	Ours	**91.889**	**5.18**
1024	[8]	1779.3	147.85
	Ours	**367.53**	**19.428**
4096	[8]	6870	569.9
	Ours	**1470**	**76.00**

FPSI for $L_{\mathsf{p}\in[1,\infty]}$ Distance. We compare our FPSI from SAS Fmap in Sec 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

[13] https://github.com/osu-crypto/libOTe.git
[14] https://gmplib.org/
[15] https://github.com/intel/pailliercryptolib.git
[16] https://github.com/ladnir/cryptoTools.git
[17] https://github.com/Visa-Research/coproto.git

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 $L_{\mathsf{p}\in\{1,2\}}$ Distance. Since there is no implementation of [1]H for $L_{\mathsf{p}\in[1,\infty)}$ 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 $L_{\mathsf{p}\in\{1,2\}}$ 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 $d \geq 6$.

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

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

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

Table 5. Communication cost (MB) and running time (s) of FPSI for L_∞ distance.

$m = n$	Protocol	(d, δ)											
		(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	**0.173**	0.660	8.251	24.09	220.0	677.1	**0.517**	1.891	24.75	73.61	660.0	2042
	Ours	0.470	**0.347**	**1.384**	**0.825**	**2.298**	**1.282**	1.340	**0.696**	**3.994**	**1.727**	**6.648**	**2.501**
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	**2.766**	9.047	132.0	408.9	3520	11027	**8.266**	25.10	396.0	1225	$> 10^4$	$> 10^4$
	Ours	7.518	**3.732**	**22.14**	**9.029**	**36.75**	**14.96**	21.44	**7.930**	**63.90**	**22.28**	**106.4**	**36.99**
2^{12}	[1]L	**44.25**	143.4	2112	6612	$> 10^4$	$> 10^5$	**132.3**	420.8	> 6000	$> 10^4$	$> 10^5$	$> 10^5$
	Ours	120.2	**53.74**	**354.1**	**151.1**	**588.0**	**253.2**	343.0	**128.9**	**1022**	**391.4**	**1702**	**644.1**
2^{16}	[1]L	**708.0**	2401	$> 10^4$	$> 10^5$	$> 10^5$	$> 10^6$	**2116**	6796	$> 10^4$	$> 10^5$	$> 10^6$	$> 10^6$
	Ours	1924	**945.6**	**5665**	**2623**	**9408**	**4332**	5488	**2218**	**16358**	**6366**	**27228**	**10779**

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 rate of their Fmap instances. We give new constructions of Fmap with small expansion rate for Hamming and $L_{\mathsf{p}\in[1,\infty]}$ 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 $L_{\mathsf{p}\in[1,\infty]}$ distance of which costs scale linearly with anyone of m, n, d, and δ for the first time.

Regarding future works, we present the following thoughts:

- 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 $L_{\mathsf{p}\in[1,\infty]}$ distance uses $\mathsf{R} \wedge \mathsf{S}$. 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. 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 protocol in the malicious setting as a 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
2. 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)
3. 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
4. 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)
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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
11. 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
12. 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

13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. 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
20. 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
21. 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
22. 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
23. 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)
24. 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

25. 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)
26. 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