Unbalanced Fuzzy Private Set Intersection for L_∞ Distance: Achieving Sublinear Communication with Large Set Size

Shengzhe Meng[1] Xiaodong Wang[1] Xv Zhou[2*] Bei Liang[3†]

[1]*Tsinghua University,* [2]*Beihang University*
[3]*Beijing Institute of Mathematical Sciences and Applications*
{*msz22*, *wangxd22*}@*mails.tsinghua.edu.cn*, {*zhouxv*, *lbei*}@*bimsa.cn*

Abstract

Fuzzy private set intersection (PSI) is a cryptographic protocol that enables two parties to compute the intersection of their sets under approximate matching, with variants including standard fuzzy PSI and fuzzy PSI with sender privacy (PSI-SP). Although recent advances have led to efficient fuzzy PSI protocols, most are designed for the balanced case where both sets are of similar size. In practice, however, many applications involve highly unbalanced sets (e.g., where the receiver's set is much larger than the sender's, or vice versa). This work focuses on unbalanced fuzzy PSI for the l_∞ metric. We observe that communication in existing protocols is dominated by the transmission of oblivious key-value stores (OKVS), especially when set sizes are imbalanced. This overhead can be reduced using batch private information retrieval (BatchPIR) if the OKVS is sparse. However, such optimization requires spatial hashing with specific properties, and few existing spatial hashing schemes satisfy these requirements.

In this work, we reformulate spatial hashing and propose two new schemes: one non-interactive and one interactive, each suited to different input set conditions. Based on these, we design two unbalanced fuzzy PSI protocols that combine sparse OKVS with BatchPIR to achieve sublinear communication in the size of the larger set. The first protocol is suitable for scenarios where the receiver holds a larger set, while the second is designed for cases where the sender possesses more items. Our protocols significantly outperform state-of-the-art in communication and runtime. For example, in a 100 Mbps network with parameters $(N, M, d, \sigma) = (2^{20}, 2^5, 2, 10)$, our protocol based on our non-interactive spatial hashing reduces communication from 16,128 MB (Baarsen and Pu, Eurocrypto'24) to 0.35 MB. Furthermore, our fuzzy PSI protocol, which utilizes our interactive spatial hashing approach, achieves at least 31× faster online runtime and 1762× lower communication than Gao et al. (Asiacrypt'25). For our fuzzy PSI protocol with sender privacy, we outperform Piske et al. (CCS'25) with at least 4× faster online runtime and 6× lower communication.

1 Introduction

Private Set Intersection (PSI) allows a receiver and a sender to compute the intersection of their respective input sets without leaking additional information beyond the result. Traditional PSI research focuses on exact matching, where the receiver learns only elements that identically exist in both sets. This primitive has been widely adopted in various privacy-sensitive applications, such as confidential contact detection [16, 21], and sharing of location data while preserving user privacy [26]. Recent advances have led to highly efficient PSI protocols [5, 22, 28, 30, 31] achieving practical performance for large-scale datasets.

However, the functionality of traditional PSI is insufficient for real-world applications that requires approximate matching, such as biometric identification [33], where the receiver requires fingerprints or iris scans from the sender's set that are close to its inputs due to inherent sensor noise, as well as in the vulnerable password detection [4], where the receiver aims to verify whether its passwords share similarities with previously exposed passwords.

Fuzzy PSI allows a receiver to obtain the sender's inputs within a specified distance norm. This work mainly focuses on fuzzy PSI on L_∞ distance as it provides a worst-case guarantee than other distance metrics and acts as a core metric in several practical applications. For instance, in biometric identification, fingerprints or iris scans are often converted into feature vectors using deep learning models, and the L_∞ distance offers superior robustness against inherent noise compared to alternatives. Furthermore, the L_∞ norm serves as the standard metric in location-based services (LBS). For instance, in applications like Geo-fencing, determining whether a location falls within a square-shaped area is naturally formulated using the L_∞ distance. Fuzzy PSI for L_∞-norm provides an alternative approach for a privacy-preserving LBS protocol.

Recent advances in fuzzy PSI include [11–14, 29, 34]. As

*Also with Beijing Institute of Mathematical Sciences and Applications
†Corresponding author

most prior work in fuzzy focuses on the balanced setting, where both parties' input sets are similarly sized, which is not the best fit for some real-world use cases such as vulnerable password detection and image matching [23]. In these applications, the input set of one party is significantly larger than the input set of the counterpart. For instance, in the application of vulnerable password detection, the sender holds millions of leaked passwords (e.g., Rockyou leaked password dataset [32]), while the receiver checks only hundreds of personal passwords. If we apply the existing fuzzy PSI protocols designed for the balanced setting, the communication complexity would grow linearly with the size of the larger set. This can be highly prohibitive, especially for users with limited resources.

In this paper, we focus on the fuzzy PSI under unbalanced settings for L_∞ distance, where the one party's set is significantly larger than the other's, and aim to provide fuzzy PSI protocols that achieve sublinear communication complexity relative to the larger set size.

1.1 Motivation

Current fuzzy PSI protocols for L_∞ distance typically consist of three key components: fuzzy matching, spatial hashing, and a framework integrating these into a fuzzy PSI protocol.

Fuzzy Matching: For items x (from receiver) and y (from sender) in a d-dimensional domain U^d, fuzzy matching allows the receiver to learn whether $\|x-y\|_\infty \leq \sigma$ for a given distance threshold σ.

Spatial Hashing: The naive pairwise comparison approach for fuzzy PSI incurs $O(MN)$ communication complexity when comparing M items of receiver with N items of sender. To reduce the comparison times, Garimella et al. [13] proposed spatial hashing, a technique employing two hash functions $\mathcal{H}_1, \mathcal{H}_2$ that map a d-dimensional element to a set of certain number of items (e.g., a number of h_1 and h_2 items, respectively). Given a distance threshold σ and some specific restrictions on both parties' inputs, the spatial hashing scheme between parties P_1 holding set X and P_2 holding set Y satisfies two specific properties:

- **Non-Conflict**: For any $x \in X$, the hashed sets $\mathcal{H}_1(x)$ are pairwise disjoint.
- **Local Overlapping**: $\mathcal{H}_1(x) \bigcap \mathcal{H}_2(y) \neq \emptyset$ once $\|x-y\|_\infty \leq \sigma$ for any $x \in X$ and $y \in Y$.

A spatial hashing scheme can be instantiated either by an interactive or a non-interactive protocol. With a spatial hashing scheme, both parties first compute their spatial hashing output, then perform fuzzy matching only on items with intersecting hashes to achieve the fuzzy PSI protocol.

Framework: If the output sets of spatial hashing on input sets X and Y, namely $\mathcal{H}_1(x)$ and $\mathcal{H}_2(y)$ intersect, it might happen that $\|x-y\|_\infty > \sigma$. Both parties must invoke the fuzzy matching protocol on the input items whose spatial hashing outputs intersect. Recent works by Baarsen et al. [34] and Gao et al. [11] each proposed a framework leveraging the OKVS technique. When applying these frameworks with spatial hashing to unbalanced fuzzy PSI or PSI-SP protocols, we observe that the party P_L (as the receiver of fuzzy PSI protocols and the sender of PSI-SP protocols) holding a large set of cardinality N_L needs to encode and transmit an OKVS of size $O(h_1 N_L)$ to the party P_S, which becomes the dominant communication overhead in the protocol. In contrast, the party P_S, with the smaller set of size N_S, only requires decoding $O(h_2 N_S)$ values from the OKVS. Our key insight is that instead of directly sending the entire OKVS to P_S, we can employ Private Information Retrieval (PIR) techniques to allow P_S to selectively obtain only the necessary decoded values, thereby reducing the overall communication cost to sub-linear in N_L. This optimization approach has been successfully implemented in unbalanced circuit-PSI and private set union (PSU) [15, 36], demonstrating practical effectiveness in reducing communication costs while maintaining all security and functionality requirements.

While the PIR technique offers potential communication savings, its direct application to existing fuzzy PSI frameworks has its challenges. On the one hand, as the decoding cost for party P_S scales linearly with h_2 (the output size of the spatial hashing function $\mathcal{H}_2$), most of the spatial hashing schemes can not guarantee a small constant of h_2. Baarsen et al. [34] proposed two non-interactive spatial hashing schemes, the first of which requires the input set X of P_1 to be 2σ apart, namely every two distinct items $x_i, x_j \in X$ satisfy that $\|x_i - x_j\|_\infty > 2\sigma$, and achieves $h_2 = 2^d$. Their second scheme achieves $h_2 = 1$ but requires the input set X to be 4σ apart, which is a strict constraint and severely limits its applicability. Gao et al. [11] developed an interactive spatial hashing protocol with $h_2 = 1$ relying on dimensional separation. Given any $x = \{x_1, \cdots, x_d\} \in U^d$, we denote x_i as the projection x on dimension i. If the distance for the projection of any two points $x \in X$ is larger than 2σ in dimension i, the set X is "separated" in that dimension. Their protocol is based on the assumption that both parties' sets are "separated" on at least one dimension. While this condition holds with high probability for randomly distributed high-dimensional data, the protocol incurs linear communication costs to both $|X|$ and $|Y|$. Consequently, even with the spatial hashing protocol from Gao et al. [11], fuzzy PSI retains linear overhead relative to the larger set.

In this paper, we aim to provide new fuzzy PSI protocols that achieve sublinear communication complexity to the larger set size N_L and maintain linear complexity to the smaller set size N_S. Our protocols rely on novel non-interactive and interactive spatial hashing schemes, where the output of $\mathcal{H}_2$ consists of only one element. The non-interactive scheme releases the restriction to P_1 from 4σ apart to 2σ apart, and the interactive spatial hashing protocol has a sublinear com-

munication complexity in terms of P_1's input size.

1.2 Our Contributions

Spatial Hashing. We revisit the definition of spatial hashing, and generalize the notion as a functionality. We particularly introduce the property of $\mathcal{H}_2$**-unique**, which requires the hash function $\mathcal{H}_2$ to output only one element, namely, $h_2 = 1$.

Non-interactive Spatial Hashing. We present a novel non-interactive $\mathcal{H}_2$-unique spatial hashing scheme when the input set of P_1 is T apart for any $T > 2\sigma$. In particular, we provide a spatial hashing scheme that $h_2 = 1$ when $T = 3\sigma$, while the existing approaches lack a solution for this specific case.

Our scheme partitions the input domain into grid cells, where each cell forms an L_∞ ball with side length $l = T - 2\sigma$, represented as $[c_1, c_1 + l) \times \cdots \times [c_d, c_d + l)$. We define the point $(c_1, \cdots, c_d)$ as the label of the cell. For any $x \in X$, P_1 computes $\mathcal{H}_1(x)$ as the labels of all grid cells intersecting the σ-radius L_∞ ball centered at x, while P_2 computes $\mathcal{H}_2(y)$ as the single label of the containing cell for any $y \in Y$. We prove that this construction satisfies both the spatial hashing functionality and the $\mathcal{H}_2$-unique property.

Interactive Spatial Hashing. We propose an interactive $\mathcal{H}_2$-unique spatial hashing protocol that achieves $h_1 = h_2 = 1$ and is more suitable for constructing fuzzy PSI protocols with high-dimensional input domains. The protocol's communication complexity is sub-linear in P_1's input size n_1. Our spatial hashing protocol requires the input set X of P_1 to be "separated" on at least one dimension, while the input set Y of P_2 to be "separated" on every dimension.

Our main idea is that for each L_∞ ball centered at $x = (x_1, \cdots, x_d) \in X$ with radius σ, it is represented as a projection on each dimension i, namely $[x_i - \sigma, x_i + \sigma]$. On every dimension, for all of the input in X, P_1 merges any intersecting projection intervals from all inputs into single intervals and associates each resulting interval with a random value. The spatial hash $\mathcal{H}_1(x)$ is then computed by identifying which interval x_i falls into for each dimension $i \in 1, \cdots, d$ and aggregating the corresponding random values. The separation condition on P_1's inputs ensures distinct outputs for different $x \in X$. For each $y = (y_1, \cdots, y_d) \in Y$, P_2 computes $\mathcal{H}_2(y)$ by summing the random values associated with the intervals containing each y_i.

To abstract such a mechanism, we introduce a new functionality called *Private Value Sum* ($\mathcal{F}_{\mathsf{PVS}}$), which allows a receiver with m inputs $x^1, \cdots, x^m \in U$ to obtain the sum of values associated with each x^i across m sender's sets $Y^1, \cdots, Y^m$ (each containing n key-value pairs). We realize the functionality $\mathcal{F}_{\mathsf{PVS}}$ through two different approaches. The first is to use additively homomorphic encryption (AHE), and the second is to employ oblivious pseudorandom functions (OPRF). Those two PVS protocols have a similar communication complexity, while our AHE-based protocol requires less communication overhead and incurs a larger computation cost compared to our OPRF-based protocol. With our PVS protocols, our spatial hashing protocol achieves the communication complexity of $O(n_2 d \sqrt[3]{n_1 \sigma})$, which is sublinear in terms of the input size of P_1. In contrast, the protocol from Gao et al. [11] requires $O(\sigma d(n_1 + n_2))$ communication complexity.

We implement our interactive spatial hashing protocol. Our protocol outperforms the protocol from Gao et al. [11] and requires less communication cost. When $n_1 = 2^{18}$, $n_2 = 2^5$, $d = 2$, and $\sigma = 10$, our protocol that utilizes AHE-based PVS protocol only requires 8.62 seconds of online running time under 100 Mbps bandwidth with 0.34 MB communication cost, which achieves $82\times$ speedup in runtime and reduces $2000\times$ communication cost than their protocol.

Both our non-interactive and interactive spatial hashing schemes require at least one party's input set to satisfy certain conditions. This ensures the Non-Conflict property of spatial hashing, which is necessary for our fuzzy PSI protocols. Invoking our fuzzy PSI protocols with arbitrary inputs may fail, since the receiver (sender) may be unable to encode the OKVS. However, our solutions remain robust and practical despite those structural requirements on participants' inputs. In practice, those conditions can be naturally satisfied through a straightforward pre-processing. For instance, clustered data points can be merged into a single representative point. For example, in the private ride-sharing scenario, where passengers want to know whether there are any available Ubers, the system can merge closed cars into a single car to represent them. Additionally, parties may also adjust the distance threshold σ, or select between our interactive and non-interactive hashing methods based on their data's distribution to balance precision with feasibility before protocol execution.

Fuzzy PSI protocols for L_∞ Distance. We address two distinct unbalanced cases for fuzzy PSI, where either the receiver or the sender has a large input set.

When the receiver has a large input set, we propose a fuzzy PSI(-CA) protocol that allows the receiver to obtain the (cardinality of) the sender's input that is close to at least one of the receiver's inputs. Conversely, when the sender possesses the larger set, we construct an alternative protocol for the fuzzy PSI-SP functionality, enabling the receiver to determine which of its own inputs are close to any element in the sender's inputs. Both protocols leverage our novel spatial hashing schemes combined with BatchPIR techniques to achieve sublinear communication complexity relative to the larger set size.

We implement our fuzzy PSI protocols using our spatial hashing schemes and test them against existing protocols under WAN settings (100 Mbps and 10 Mbps bandwidth with 80ms RTT latency). For our fuzzy PSI(-CA) protocol with our non-interactive spatial hashing scheme, we conducted experiments for the case when the receiver's set is $T = 3\sigma$ or $T = 4\sigma$ apart. We compared them with the protocols by [34], which require the receiver's set to be 2σ and 4σ apart, respectively.

Under the parameter set $(N, M, d, \sigma) = (2^{18}, 2^5, 2, 10)$, our protocol demonstrates significant performance improvements. For $T = 3\sigma$ under 10 Mbps bandwidth, it achieves a 2880× reduction in communication cost and a 5.7× improvement in online runtime. When $T = 4\sigma$, our protocol requires only 0.35 MB of communication and 53.5 seconds of online runtime, whereas the protocol from Baarsen et al. requires 4032 MB and at least 334 seconds under the same conditions. We also compare our fuzzy PSI(-CA) protocol instantiated with our interactive spatial hashing scheme against the fuzzy PSI protocol of Gao et al. [11] (excluding the interactive spatial hashing and setup phases in both). Our protocol achieves at least a 31× improvement in online runtime and a 1762× reduction in communication overhead across various parameter sets. For fuzzy PSI-SP, our protocol, which utilizes non-interactive spatial hashing, is evaluated against Piske et al. [29]. Across all tested configurations with a bandwidth of 100 Mbps or less, it achieves a speedup of 6.18× to 101.04× in online runtime and reduces communication cost by 4.43× to 284.34×. The detailed communication complexity is presented in Tab. 1.

2 Preliminary

Notation : Let κ and λ denote the computational and statistical security parameters, respectively. We use $r \xleftarrow{\$} R$ to denote sampling r randomly from the set R. Given a set X, we denote $|X|$ as the cardinality of the set. Given two values or strings a and b, $a\|b$ denotes the concatenation of those values or strings. For a s-bits string $r \in \{0,1\}^s$ or a s-length vector $r \in \mathbb{F}^s$, we denote r^a as the a-th bit(component) of the string(vector) r, where $a \in \{1, \cdots, s\}$. $\langle a, b \rangle$ denotes the inner product of vectors a and b. In our fuzzy PSI protocol, we denote $\mathcal{R}$ as the receiver and N as the cardinality of the receiver's input set, while $\mathcal{S}$ is the sender and M is the cardinality of the sender's set. Given a set of n key-value pairs $Set = \{(k_i, v_i)\}_{i=\{1,\cdots,n\}}$, we define $Set[k]$ as the value associated to the key k, and $K_{Set} = \{k_i\}_{i=\{1,\cdots,n\}}$ as the key set of Set. By negl (κ) we denote a negligible function, i.e., a function f such that $f(\kappa) < 1/p(\kappa)$ holds for any polynomial $p(\cdot)$ and sufficiently large κ.

2.1 Batch Private Information Retrieval

Private Information Retrieval (PIR) [7] enables a client to input an index i and retrieve the i-th entry in the database DB of n entries privately. The communication cost for a PIR process is sublinear to n. Batch Private Information Retrieval (BatchPIR) [1, 18, 24, 25] enables the client to retrieve a batch of entries indexed as $\{i_1, \cdots, i_b\}$ from the server's database DB at a lower amortized cost. The detailed definitions of BatchPIR are given in the Appendix B.2.

In our paper, we utilize the BatchPIR scheme from Mughees et al. [25], which requires $O(\sqrt[3]{n})$ communication complexity for retrieving a single entry from the database. Their scheme achieves a better communication complexity than $O(b\sqrt[3]{n})$ when retrieving b entries simultaneously, but we neglect this aspect to simplify our theoretical analysis of communication overhead.

2.2 Oblivious Key-Value Store

A key-value store (KVS) is a data structure that maps a set of keys to corresponding values. A KVS is an oblivious key-value store (OKVS) once the distribution of the data structure is independent of the key set when those values are random. OKVS consists of two algorithms: Encode and Decode. In our construction, we require the OKVS to meet the properties of randomness, linearity, and sparsity. An OKVS is linear (over a field $\mathbb{F}$) if the output of Encode is a vector T in $\mathbb{F}^m$, and the Decode function is defined as : $\mathsf{Decode}(T, k) = \langle \mathrm{d}(k), T \rangle = \sum_{j=1}^{m} \mathrm{d}(k)_j T_j$ for some function $\mathrm{d} : \mathcal{K} \to \mathbb{F}^m$. An OKVS is sparse once the output of the Encode function T can be divided into two parts $T = T_0 \| T_1$, where $T_0 \in \mathbb{F}^s$ and $T_1 \in \mathbb{F}^d$ and $d = o(s)$. There are two functions involving the Decode function, $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$. $\mathsf{Decode}(T, k) \stackrel{\text{def}}{=} \langle T_0, \mathsf{sparse}(k) \rangle + \langle T_1, \mathsf{dense}(k) \rangle$ for decoding the sparse OKVS T with any key $k \in \mathcal{K}$. The Hamming weight of the output of the sparse function is a fixed value w.We provide the detailed definition of OKVS and its properties in Appendix B.3.

In this paper, we utilize the OKVS scheme proposed by Raghuraman et al. [30] that satisfies the properties of randomness, linearity, and sparsity. The output of the sparse function has a Hamming weight $w = 3$, the size of the sparse part $s = O(n)$ while the size of the dense part $d = O(\log n)$.

2.3 Fuzzy Matching for L_∞ Distance

Fuzzy matching allows the receiver to determine whether its input x is close to the sender's input y according to a distance threshold σ for any x and y in a d-dimensional domain U^d. The receiver obtain 1 once $\|x - y\|_\infty \le \sigma$ and 0 otherwise. Baarsen and Pu [34] proposed a protocol achieving $\mathcal{F}_{\mathsf{match}}$ whose communication cost is $O(d\sigma)$. The detailed functionality is shown in Appendix B.5.

2.4 Fuzzy PSI Functionality

Fuzzy PSI enables the receiver and the sender to input a private set, allowing the receiver to obtain different fuzzy intersection information based on varied functionalities, including $\mathcal{F}_{\text{Fuzzy PSI}}$, $\mathcal{F}_{\text{Fuzzy PSI-CA}}$, and $\mathcal{F}_{\text{Fuzzy PSI-SP}}$. The detailed functionality is shown in Fig. 1.

3 Spatial Hashing functions for L_∞ Distance

In this section, we introduce a hashing primitive suitable for unbalanced fuzzy PSI protocols, defined over an input do-

Functionality	Protocol	Condition	Communication
Fuzzy PSI (-CA)	[34]	$R > 2\sigma$	$O(\sigma d N_L + 2^d N_S)$
		$R > 4\sigma$	$O(\sigma 2^d d N_L + N_S)$
		R 1-separate	$O((\sigma d)^2 N_L + N_S)$
	[11]	R & S 1-separate	$O(\sigma d(N_L + N_S))$
	[12]	-	$O(dN_L \log(\sigma) + N_S(2\log(\sigma))^d)$
	Ours (Fig. 6)	$R \geq T > 2\sigma$	$O(dN_S \sqrt[3]{N_L d h^d \sigma})$
		R 1-separate & S d-separate	$O(dN_S \sqrt[3]{N_L \sigma})$
Fuzzy PSI-SP	[11]	R & S 1-separate	$O(\sigma d(N_L + N_S))$
	[29]	$R > 4\sigma$ & $S > 2\sigma$	$O(d(N_S 2^d + N_L \sigma))$
	Ours (Fig. 7)	$S \geq T > 2\sigma$	$O(dN_S \sqrt[3]{N_L h^d} + dN_S \sigma)$
		S 1-separate & R d-separate	$O(dN_S \sqrt[3]{N_L \sigma} + dN_S \sigma)$

Table 1: Comparison of protocols achieving functionalities fuzzy PSI(-CA) and fuzzy PSI-SP for L_∞ distance across various metrics. N_L: number of set elements of the larger set. N_S: number of set elements of the smaller set. For the functionality fuzzy PSI(-CA), we assume the receiver holds the larger set of N_L elements, while the sender holds a smaller set of N_S elements. For the functionality fuzzy PSI-SP, we assume the sender holds the larger set of N_L elements, while the receiver holds a smaller set of N_S elements. σ: distance threshold. d: dimension of each input item. $h = \max\{\lceil (2\sigma+1)/(T-2\sigma) \rceil, 2\}$. $R/S > 2\sigma/4\sigma$: The distance between any two items in the receiver/(sender)'s set is greater than $2\sigma/4\sigma$. $R/S \geq T > 2\sigma$: The distance between any two items in the receiver/(sender)'s set no less than T, for a given $T > 2\sigma$. R/S 1-separate: The receiver(sender)'s set is "separated" on at least one dimension. R/S d-separate: The receiver(sender)'s set is "separated" on every dimension.

Functionality 1. (Fuzzy PSI)

- **Parameters:** Two parties, the receiver and sender. The distance threshold $\sigma \geq 0$. Input domain U^d. The cardinality of both parties' sets, N and M, respectively.
- **Inputs:** The receiver inputs $W = \{w_1, \cdots, w_N\} \in U^{d \times N}$ and the sender inputs $Q = \{q_1, \cdots, q_M\} \in U^{d \times M}$.
- **Outputs:**
  - $\mathcal{F}_{\text{Fuzzy PSI}}$: Output $\{q_i \in Q \mid \exists w_j \in W, \|w_j - q_i\|_\infty \leq \sigma\}$ to the receiver.
  - $\mathcal{F}_{\text{Fuzzy PSI-CA}}$: Output $|\{q_i \in Q \mid \exists w_j \in W, \|w_j - q_i\|_\infty \leq \sigma\}|$ to the receiver.
  - $\mathcal{F}_{\text{Fuzzy PSI-SP}}$: Output $|\{w_j \in W \mid \exists q_i \in Q, \|w_j - q_i\|_\infty \leq \sigma\}|$ to the receiver.

Figure 1: Ideal functionality for $\mathcal{F}_{\text{Fuzzy PSI}}$, $\mathcal{F}_{\text{Fuzzy PSI-CA}}$ and $\mathcal{F}_{\text{Fuzzy PSI-SP}}$.

main U, a distance threshold σ, and an output domain $\mathcal{B}$. The spatial hashing involves two parties P_1 and P_2 inputting sets $X = \{x_1, \cdots, x_{n_1}\} \subseteq U$ and $Y = \{y_1, \cdots, y_{n_2}\} \subseteq U$, respectively, which may satisfy certain specific conditions depending on the spatial hashing scheme. Both parties will receive their spatial hashing outputs: $\mathcal{H}_1(x_1), \cdots, \mathcal{H}_1(x_{n_1}) \subseteq \mathcal{B}$ for P_1 and $\mathcal{H}_2(x_1), \cdots, \mathcal{H}_2(x_{n_2}) \subseteq \mathcal{B}$ for P_2, where the upper bounds on the cardinalities of these sets are h_1 and h_2, respectively. The scheme ensures that for any x_i and y_j that $\|x_i - y_j\| \leq \sigma$, $\mathcal{H}_1(x_i)$ has at least one common element with $\mathcal{H}_2(y_j)$, while $\mathcal{H}_1(x_i)$ does not intersect with $\mathcal{H}_1(x_j)$ for any $i \neq j$. However, $\mathcal{H}_1(x) \bigcap \mathcal{H}_2(y) \neq \emptyset$ doesn't guarantee that $\|x - y\| \leq \sigma$. The detailed functionality is shown in Fig. 2.

Functionality 2. (Spatial Hashing - $\mathcal{F}_{\text{SH}}$)

- **Parameters:** Parties P_1 and P_2. Input domain U, a distance threshold σ, an output domain $\mathcal{B}$, two positive integer h_1, h_2 and conditions $\mathcal{C}_1$, $\mathcal{C}_2$.
- **Inputs:** P_1 inputs $X = \{x_1, \cdots, x_{n_1}\} \subseteq U$ under condition $\mathcal{C}_1$ while P_2 inputs $Y = \{y_1, \cdots, y_{n_2}\} \subseteq U$ under condition $\mathcal{C}_2$.
- **Outputs:** Generate two function $\mathcal{H}_1, \mathcal{H}_2 : U \to \mathsf{PowerSet}(\mathcal{B})$ that $0 < |\mathcal{H}_1(i)| \leq h_1$, $0 < |\mathcal{H}_2(j)| \leq h_2$ for any $i, j \in U$. Output $\mathcal{H}_1(x_1), \cdots, \mathcal{H}_1(x_{n_1})$ to P_1 and $\mathcal{H}_2(y_1), \cdots, \mathcal{H}_2(y_{n_2})$ to P_2, respectively. Those outputs satisfy the two properties below:
  1. **Locally overlapping:** $\mathcal{H}_1(x_i) \bigcap \mathcal{H}_2(y_j) \neq \emptyset$ once $\|x_i - y_j\|_\infty \leq \sigma$.
  2. **Non-conflicting:** $\mathcal{H}_1(p) \bigcap \mathcal{H}_1(q) = \emptyset$ for any distinct $p, q \in X$.

Figure 2: Ideal functionality for spatial hashing.

This functionality can be achieved either locally or through a two-party protocol, depending on the input set conditions. We denote those two cases as non-interactive spatial hashing and interactive spatial hashing. Specifically, when a spatial hashing scheme or protocol satisfies $h_2 = 1$, which $\mathcal{H}_2(x)$

outputs a singleton set for every $x \in U$, we refer to it as $\mathcal{H}_2$-**unique spatial hashing**. In Sec. 3.1 and Sec. 3.2, we present a non-interactive spatial hashing scheme and an interactive spatial hashing protocol, both of which are $\mathcal{H}_2$-**unique**.

3.1 Our Non-interactive Spatial Hashing Scheme

Suppose the input domain $U^d = \{0, \cdots, 2^u - 1\}^d$. Our non-interactive spatial hashing scheme requires that items in P_1's set X satisfy a minimum separation of $T > 2\sigma$, i.e., for every distinct $x_i, x_j \in X$, $\|x_i - x_j\|_\infty \geq T$. No restrictions are imposed on P_2's set Y, which may be arbitrary. The scheme partitions the input domain U^d into several d-dimensional cells of side-length $l = T - 2\sigma$ along each dimension. Each cell $[c_1, c_1 + l) \times \cdots \times [c_d, c_d + l)$ is assigned a label $(c_1, \cdots, c_d)$, and output domain $\mathcal{B}$ is defined as the set of the set of all labels: $\mathcal{B} = \{(k_1 l, \cdots, k_d l)\}_{k_1, \cdots, k_d \in \{0, \cdots, \lfloor \frac{2^u}{l} \rfloor\}}$. For any point $x = (x_1, \cdots, x_d) \in X$, we define the function $cell_l(x) = (\lfloor \frac{x_1}{l} \rfloor l, \cdots, \lfloor \frac{x_d}{l} \rfloor l)$, which maps x to the label of its containing cell. The high-level idea of our non-interactive spatial hashing scheme is that $\mathcal{H}_1(x)$ returns the labels of all cells intersected by the L_∞ ball of radius σ centered at x (which has side length $2\sigma + 1$), while $\mathcal{H}_2(y)$ returns $cell_l(y)$. If $\|x - y\|_\infty \leq \sigma$, y lies within the $2\sigma + 1$ side-length L_∞ ball centered at x, ensuring $\mathcal{H}_2(y) \subseteq \mathcal{H}_1(x)$, thereby satisfying the spatial hashing requirement. The detailed construction is presented below.

Given any point $x = (x_1, \cdots, x_d), y = (y_1, \cdots, y_d)$, let $h = \max\{\lceil \frac{2\sigma+1}{l} \rceil, 2\}$, $h_1 = h^d$, $h_2 = 1$. $\mathcal{H}_1(x) = \{cell_l(p)\}_{p \in [x_1 - \sigma, x_1 + \sigma] \times \cdots \times [x_d - \sigma, x_d + \sigma]}$. $\mathcal{H}_2(y) = \{cell_l(y)\}$. P_1 and P_2 will calculate and output $\mathcal{H}_1(x)$ and $\mathcal{H}_2(y)$ for every $x \in X$ and $y \in Y$ locally.

We define the security of our non-interactive spatial hashing scheme in Theorem 1, and a detailed proof is provided in Appendix E.1.

3.2 Our Interactive Spatial Hashing Protocol for High-Dimensional Inputs

The non-interactive spatial hashing scheme described in Sec. 3.1 allows both parties to compute their spatial hashing output locally. However, this scheme has a limitation: the output for P_1, $\mathcal{H}_1(x)$, contains $h_1 = h^d$ elements, which leads to inefficient performance in fuzzy PSI protocols when the dimension d is large. To address this issue, we introduce an interactive spatial hashing protocol in this section. The protocol involves two parties, P_1 and P_2, with input sets $X = \{x_1, \cdots, x_{n_1}\} \subseteq U^d$ and $Y = \{y_1, \cdots, y_{n_2}\} \subseteq U^d$, respectively. For a given distance threshold σ, they interactively compute hashing outputs $\mathcal{H}_1(x_i)$ for every $x_i \in X$ and $\mathcal{H}_2(y_j)$ for every $y_j \in Y$. In contrast to the non-interactive scheme, both $\mathcal{H}_1$ and $\mathcal{H}_2$ now output only a single element each.

Definition 1. *Given a d-dimensional domain U^d and a set $P = \{p_1, \cdots, p_n\} \subseteq U^d$, and a distance threshold σ, the set P is separated on dimension k, $k \in \{1, \cdots, d\}$ once for every $p_i = (p_i^1, \cdots, p_i^d), p_j = (p_j^1, \cdots, p_j^d) \in P$ that $i \neq j$, $\|p_i^k - p_j^k\| > 2\sigma$.*

To achieve this property, our spatial hashing protocol imposes a different constraint compared to the non-interactive scheme introduced in Sec. 3.1. We assume that P_1's input set is separated on at least one dimension, while P_2's input set is separated on every dimension. Baarsen et al. [34] proved that a set is separated on at least one dimension with high probability when its items are uniformly distributed in the input domain. We also analyze the probability that P_2's input set satisfies the condition of being separated on every dimension in Appendix D. The analysis shows that P_2's set meets this requirement with high probability when it contains fewer items and its inputs are uniformly distributed.

We propose and employ a novel functionality, private value sum $\mathcal{F}_{\mathsf{PVS}}$, for the construction of our interactive spatial hashing protocol. This functionality involves two parties, the receiver and the sender. The receiver inputs m keys $x_1, \cdots, x_m$, and the sender inputs m sets of key-value pairs $Y_1, \cdots, Y_m$. If every key x_i belongs to the key set of Y_i, i.e., $x_i \in K_{Y_i}$, the receiver obtains the sum of the values $Y_i[x_i]$ corresponding to each key x_i. Otherwise, the receiver obtains a random value. The detailed functionality is shown in Fig. 3. We present two protocols that realize $\mathcal{F}_{\mathsf{PVS}}$ in Sec. 3.3.

Functionality 3. (Private Value Sum - $\mathcal{F}_{\mathsf{PVS}}$)

- **Parameters:** Two parties receiver and sender. Input domain U. A finite field $\mathbb{F}$. Set number m and size n.
- **Inputs:** The receiver inputs m keys $x_1, \cdots, x_m \in U$ while the sender inputs m sets of key-value pairs $Y_1, \cdots, Y_m$, where $Y_i = \{k_{ij}, v_{ij}\}_{j \in \{1, \cdots, n\}}$ that $k_{ij} \in U$ and $v_{ij} \in \mathbb{F}$. Let $K_{Y_i} := \{k_{ij}\}_{j \in \{1, \cdots, n\}}$ to be the key set of Y_i.
- **Outputs:** For $i \in \{1, \cdots, m\}$, let $c_i = Y_i[x_i]$ if $x_i \in K_{Y_i}$, otherwise c_i is randomly sampled from $\mathbb{F}$. Output $c = \sum_{i=1}^m c_i$ to the receiver.

Figure 3: Ideal functionality for private value sum.

With this new functionality $\mathcal{F}_{\mathsf{PVS}}$, we present our interactive spatial hashing protocol Π_{ISH} in Fig. 4. We assume the input domain is d-dimensional, namely $U^d = \{0, \cdots, 2^u - 1\}^d$ and P_1 inputs n_1 items $X = \{x_1, \cdots, x_{n_1}\}$ within the domain. For every item $x_i = (x_i^1, \cdots, x_i^d) \in X$, we denote $[x_i^1 - \sigma, x_i^1 + \sigma] \times \cdots \times [x_i^d - \sigma, x_i^d + \sigma] \subseteq U^d$ as the L_∞ ball centered at x_i with radius σ, and the interval $[x_i^j - \sigma, x_i^j + \sigma]$ as the projection of the L_∞ ball on dimension j, where $j \in \{1, \cdots, d\}$.

For each dimension $i \in \{1, \cdots, d\}$, P_1 first computes, for each item of its inputs, the projection onto the i-th dimension of a L_∞ ball with radius σ centered at that item, and collects

PROTOCOL 1. Interactive Spatial Hashing Protocol Π_{ISH}

- **Parameters:** Two parties P_1 and P_2. A distance threshold σ. An input domain U^d. A finite field $\mathbb{F}$. Output domain $\mathcal{B} = \mathbb{F}$.
- **Inputs:**
  - P_1 has input set $X = \{x_1, \cdots, x_{n_1}\} \subseteq U^{d \times n_1}$ which is separated on at least one dimension.
  - P_2 has input set $Y = \{y_1, \cdots, y_{n_2}\} \subseteq U^{d \times n_2}$ which is separated on every dimension.
- **Protocol:**

Offline Phase

1. P_1 initializes d empty sets $S_i = \emptyset$ and d empty key-value sets $Y_i = \emptyset$ for $i \in \{1, \cdots, d\}$.
2. For every dimension $i \in \{1, \cdots, d\}$, $x_j = (x_j^1, \cdots, x_j^d) \in X$, let set $s_{ij} = \{k \mid k \in \mathbb{Z}, |k - x_j^i| \leq \sigma\}$. P_1 check every set $s \in S_i$, if $s \bigcap s_{ij} \neq \emptyset$, $s_{ij} \longleftarrow s_{ij} \bigcup s$ and remove s from S_i. P_1 add s_{ij} to S_i in the end.
3. For every dimension $i \in \{1, \cdots, d\}$ and every set $s_{ij} \in S_i$, P_1 generates random $r_{ij} \xleftarrow{\$} \mathbb{F}$ and add key-value pairs $\{s^l, r_{ij}\}$ to Y_i for every $s^l \in s_{ij}$. P_1 add random key-value pairs to Y_i such that $|Y_i| = n_1 \cdot (2\sigma + 1)$.
4. For $i \in \{1, \cdots, n_1\}$, let $x_i = (x_i^1, \cdots, x_i^d)$. P_1 outputs $\mathcal{H}_1(x_i) = \sum_{j=1}^{d} Y_j[x_i^j]$ for $i \in \{1, \cdots, n_1\}$.

Online Phase

1. For $i \in \{1, \cdots, n_2\}$, let $y_i = (y_i^1, \cdots, y_i^d)$. P_1 and P_2 invoke an instance of $\mathcal{F}_{\mathsf{PVS}}$ that P_1 acts as the sender with the inputs $Y_1, \cdots, Y_d$ and P_2 acts as the receiver with the inputs $y_i^1, \cdots, y_i^d$. P_2 obtains c_i.
2. P_2 outputs $\mathcal{H}_2(y_i) = c_i$ for $i = \{1, \cdots, n_2\}$.

Figure 4: Interactive Spatial Hashing Protocol.

these intervals into a set S_i. Then, P_1 checks every pair of intervals in S_i, merging any two that intersect into a single continuous interval. This process is repeated until all intervals in S_i are pairwise disjoint. Suppose the resulting set is $S_i = \{s_{i1}, \cdots, s_{ij}\}$, P_1 then generates j random values $r_{i1}, \cdots, r_{ij}$ from a finite field $\mathbb{F}$ and assign r_{ik} to each interval s_{ik} for $k = \{1, \cdots, j\}$. P_1 initializes an empty key-value set Y_i, and for every value $s \in s_{ik}$ in the interval and $k = \{1, \cdots, j\}$, it adds the pair $\{s, r_{ik}\}$ to Y_i. Additionally, P_1 inserts dummy random key-value pairs into Y_i such that the total size $|Y_i|$ equals $n_1(2\sigma + 1)$. For every input point $x_i = (x_i^1, \cdots, x_i^d) \in X$ and for every dimension $j \in \{1, \cdots, d\}$, P_1 retrieves the value $Y_i[x_i^j]$ associate with the key x_i^j in the key-value set Y_i, and sums these values to form the spatial hashing output $\mathcal{H}_1(x_i)$. This entire process can be carried out by P_1 without the participation of P_2 in the offline phase.

To compute $\mathcal{H}_2(y_i)$ for each input $y_i = (y_i^1, \cdots, y_i^d) \in Y$, P_1 and P_2 invoke an instance of $\mathcal{F}_{\mathsf{PVS}}$, where P_1 acts as the sender with inputs $Y_1, \cdots, Y_d$ and P_2 acts as the receiver with inputs $y_i^1, \cdots, y_i^d$. The output of $\mathcal{F}_{\mathsf{PVS}}$ for P_2 is the sum of the values associated with the key $y_i^1, \cdots, y_i^d$ in the respective key-value sets $Y_1, \cdots, Y_d$, or a random value once there exist $j \in \{1, \cdots, d\}$ that $y_i^j \notin K_{Y_i}$. If for any $j \in \{1, \cdots, d\}$, the key y_i^j is not in the key set K_{Y_j}, the output is a random value. P_2 uses this result as $\mathcal{H}_2(y_i)$. The detailed interactive spatial hashing protocol is shown in Fig. 4.

We define the security of our interactive spatial hashing protocol (Π_{ISH}) in Theorem 2, and a detailed proof is provided in Appendix E.1.

3.3 Private Value Sum protocols

AHE-based PVS protocol: In this section, we propose a protocol that achieves functionality $\mathcal{F}_{PVS}$ from the AHE technique before invoking the protocol. We suppose the receiver inputs m keys $x_1, \cdots, x_m \in U$ and the sender inputs m sets $Y_1, \cdots, Y_m$, each set contains n key-value pairs. Both parties share an AHE scheme. The sender will first initialize m empty key-value sets $Set_1, \cdots, Set_m = \emptyset$, generate an AHE key pair (pk, sk), and send the public key pk to the receiver. For $i \in \{1, \cdots, d\}$ and any key-value pair $\{k, v\} \in Y_i$, the sender computes the encryption $\mathsf{AEnc}(pk, v)$ and adds the key-value pair $\{k \| i, \mathsf{AEnc}(pk, v)\}$ to Set_i. Then, the sender generates m OKVS $T_i \longleftarrow \mathsf{Encode}(Set_i)$ for $i \in \{1, \cdots, m\}$ and sends it to the receiver. The receiver decodes the OKVS T_i key, $x_i \| i$ to obtain $c_i = \mathsf{Decode}(T_i, x_i \| i)$. Notice that once x_i belongs to the key set of Y_i and $\{x_i, v_i\} \in Y_i$, $c_i = \mathsf{AEnc}(pk, v_i)$, and c_i equal to a random ciphertext otherwise. The receiver generates a random value *mask* and computes the sum of ciphertexts $c_1, \cdots, c_m, \mathsf{AEnc}(pk, mask)$ using an additive homomorphic algorithm and sends the result c to the sender. The sender decrypts and returns the plaintext $t = \mathsf{ADec}(sk, c)$ to the receiver. The receiver computes $t - mask$ as the output of the protocol.

Note that each OKVS generated by the sender contains n key-value pairs, but the receiver only needs to decode one key per OKVS. To reduce communication costs, we employ BatchPIR and sparse OKVS techniques, which avoid the sender from transmitting the entire OKVS directly to the receiver. We assume both parties have agreed upon a BatchPIR scheme as defined in Sec. 2.1 and a sparse OKVS scheme

PROTOCOL 2. Private Value Sum Protocol from AHE. $\Pi_{\mathsf{PVS}_{\mathsf{AHE}}}$

- **Parameters:** Two parties: the receiver and the sender. Input domain U. A finite field $\mathbb{F}$. Set number m and set size n. A sparse OKVS scheme $(\mathsf{Encode}, \mathsf{Decode})$ with key space $\mathcal{K} = \{0,1\}^*$ and value space $\mathcal{V} = \mathbb{F}$. Let $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$ be the random functions used in Encode. Let w denote the Hamming weight of the sparse part. An additively homomorphic encryption scheme $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec}, \mathsf{ASum}, \mathsf{ARef})$ and a BatchPIR scheme.
- **Inputs:**
  - The receiver inputs m keys $x_1, \cdots, x_m \in U$.
  - The sender inputs m sets of key-value pairs $Y_1, \cdots, Y_m$, where $Y_i = \{k_{ij}, v_{ij}\}_{j \in \{1,\cdots,n\}}$ that $k_{ij} \in U$ and $v_{ij} \in \mathbb{F}$.
- **Protocol:**

Offline Phase

1. The sender initializes m empty key-value sets $Set_1, \cdots, Set_m = \emptyset$, generates a fresh key-pair $(pk, sk) \longleftarrow \mathsf{AGen}(\kappa)$ for the additive homomorphic encryption scheme, and sends the public key pk to the receiver.
2. For every $i \in \{1, \cdots, m\}$ and key-value pair $\{k_{ij}, v_{ij}\} \in Y_i$, the sender adds key-value pairs $\{k_{ij} \| i, \mathsf{AEnc}(pk, v_{ij})\}$ to Set_i.

Online Phase

1. The sender generates m sparse OKVS $T_i = T_{i0} \| T_{i1} \longleftarrow \mathsf{Encode}(\{k, v\}_{\{k,v\} \in Set_i})$ for $i \in \{1, \cdots, m\}$, where $T_{i0} \in \mathbb{F}^s$ and $T_{i1} \in \mathbb{F}^d$. The sender sends $T_{11}, \cdots, T_{m1}$ to the receiver.
2. The receiver computes $r_i = \mathsf{sparse}(x_i \| i) \in \{0,1\}^s$ for $i \in \{1, \cdots, m\}$. Every r_i has a Hamming weight of w. The receiver defines a set $I_i = \{a_{i1}, \cdots, a_{iw}\}$ that $r_i^{a_{ij}} = 1$ for every $j \in \{1, \cdots, w\}$. The receiver obtains $c_{ia_{ij}} = T_{i0}^{a_{ij}}$ via batchPIR and computes $c_i = \langle T_{i1}, \mathsf{dense}(x_i \| i) \rangle + \sum_{j=1}^{w} c_{ia_{ij}}$ and $c = \mathsf{ASum}(\{c_i\}_{i \in \{1,\cdots,m\}})$.
3. The receiver generates a random mask $mask \xleftarrow{\$} \mathbb{F}$ and send the cipher-text $\{c' = \mathsf{ASum}(\mathsf{AEnc}(pk, mask), c)\}$ to the sender. The sender computes $t = \mathsf{ADec}(sk, c')$ and sends t to the receiver. The receiver outputs $o = t - mask$.

Figure 5: AHE-based Private Value Sum Protocol.

$(\mathsf{Encode}, \mathsf{Decode})$as defined in Sec. 2.2 before invoking the protocol. Additionally, we assume the Hamming weight of the output of the sparse function in the sparse OKVS scheme is a fixed value w.

In our protocol, the sender computes the sparse OKVS $T_i = T_{i0} \| T_{i1} \longleftarrow \mathsf{Encode}(Set_i)$ and the receiver decodes each T_i with the key $x_i \| i$ for $i \in \{1, \cdots, m\}$. Instead of sending the whole OKVS T_i, the sender only sends the dense part T_{i1} to the receiver. To compute $\mathsf{Decode}(T_i, x_i \| i)$, the receiver first evaluates $r_i = \mathsf{sparse}(x_i \| i) \in \{0,1\}^s$, where the output has a Hamming weight w. Let $I_i = \{a_{i1}, \cdots, a_{iw}\}$ denote the set of indices where r_i equals to 1 (i.e., $r_i^{a_{ij}} = 1$ for every $j \in \{1, \cdots, w\}$). Using the BatchPIR technique, the receiver retrieves the components of the vector T_{i0} at the indices in I_i and sums them, which is equivalent to evaluating the inner product $\langle T_{i0}, \mathsf{sparse}(x_i \| i) \rangle$. The receiver then locally computes $\langle T_{i1}, \mathsf{dense}(x_i \| i) \rangle$ and sums the two results to obtain $\mathsf{Decode}(T_i, x_i \| i)$. This method is also utilized in the OPRF-based private value sum protocol and fuzzy PSI protocols described subsequently. The complete AHE-based protocol is specified in Fig. 5.

We define the security of our AHE-based PVS protocol ($\Pi_{\mathsf{PVS}_{\mathsf{AHE}}}$) in Theorem 3, and a detailed proof is provided in Appendix E.2.

OPRF-based PVS protocol: We also proposed a private value sum protocol from the OPRF technique. The detailed protocol is introduced in Appendix C.

3.4 Complexity Analysis

Our AHE-based and OPRF-based protocols exhibit similar communication complexity. In the AHE-based protocol, the receiver is required to decode one key in an OKVS of size $O(n)$ using BatchPIR for m times. For each key decoding operation on an OKVS T_i, the communication includes sending the dense part T_{i1}, which contributes $O(\log n)$ to the communication cost. Executing the BatchPIR scheme requires $O(\sqrt[3]{n})$ communication per decoding according to the scheme from Mughees et al. [25]. Thus, the total communication complexity is $O(m\sqrt[3]{n})$. In the OPRF-based protocol, the communication overhead of invoking $\mathcal{F}_{\mathsf{OPRF}}$ is linear in the receiver's input size. The remaining communication overhead arises from decoding m keys across m OKVS, each of size $O(n)$, utilizing the BatchPIR technique. Thus, the overall communication complexity of our OPRF-based protocol is $O(m\sqrt[3]{n})$.

In our interactive spatial hashing protocol, P_1 and P_2 invoke n_2 instances of $\mathcal{F}_{\mathsf{PVS}}$, each time P_2 acts as the receiver inputting d keys while P_1 acts as the sender inputting $n_1(2\sigma+1)$

key-value pairs. The overall communication complexity cost of our interactive spatial hashing protocol is $n_2 d \sqrt[3]{n_1 \sigma}$, which is sub-linear to the input size of P_1.

4 Unbalanced Fuzzy PSI Protocols

In this section, we propose the fuzzy PSI protocols under two unbalanced scenarios: one where the receiver's input set is larger than the sender's ($N \gg M$), and the other where the sender's set is larger ($M \gg N$). Our objective is to design protocols whose communication complexity is linear in the size of the smaller set and sublinear in the size of the larger set. However, due to the inherent requirements of the fuzzy PSI functionality $\mathcal{F}_{\text{Fuzzy PSI}}$, the communication complexity of any protocol realizing this functionality must be at least linear in the size of the sender's input set. Similarly, for the functionality $\mathcal{F}_{\text{Fuzzy PSI-SP}}$, the communication complexity is inherently lower-bounded by the size of the receiver's input set. Thus, we provide a protocol for $\mathcal{F}_{\text{Fuzzy PSI}}$ (which can also achieve $\mathcal{F}_{\text{Fuzzy PSI-CA}}$) when the receiver has a larger set. We provide an alternative protocol for functionality $\mathcal{F}_{\text{Fuzzy PSI-SP}}$ when the sender has a larger set.

4.1 Unbalanced Fuzzy PSI Protocol when the Receiver has a Larger Set

We first propose our unbalanced fuzzy PSI protocol, which achieves the functionalities $\mathcal{F}_{\text{Fuzzy PSI}}$ and $\mathcal{F}_{\text{Fuzzy PSI-CA}}$ in the scenario where the receiver possesses a larger set. Our protocol builds upon the $\mathcal{H}_2$**-unique** spatial hashing schemes introduced in Sec. 3.1 and Sec. 3.2, combined with the additive homomorphic encryption technique. We assume the receiver inputs a set $W = \{w_1, \cdots, w_N\}$ of N items satisfying condition $\mathcal{C}_1$ while the sender inputs a set $Q = \{q_1, \cdots, q_M\}$ of M items satisfying condition $\mathcal{C}_2$, where $N \gg M$. According to those conditions, both parties first compute their $\mathcal{H}_2$**-unique** spatial hashing outputs either locally or via the interactive spatial hashing protocol. The receiver obtains $\mathcal{H}_1(w_i) = \{w_{i1}, \cdots, w_{ih_1}\}$ for every $w_i \in W$ while the sender obtains $\{k_i\} = \mathcal{H}_2(q_i)$ for every $q_i \in Q$ during the spatial hashing phase.

In the setup phase, the receiver generates a fresh AHE key pair (pk, sk) and shares the public key pk with the sender. The receiver initializes an empty key-value set $Set = \emptyset$. For every item $w_i = (w_i^1, \cdots, w_i^d) \in W$ and its corresponding spatial hashing output $\mathcal{H}_1(w_i) = \{w_{i1}, \cdots, w_{ih_1}\}$, the receiver adds key-value pairs $\{w_{ik} \| x \| j, \mathsf{AEnc}(pk, 0)\}$ to Set for all $k \in \{1, \cdots, h_1\}$, $j \in \{1, \cdots, d\}$ and every $x \in [w_i^j - \sigma, w_i^j + \sigma]$. $\mathsf{AEnc}(pk, 0)$ represents a fresh encryption of 0 under pk, , with each encryption being distinct across different key-value pairs. Due to the non-conflicting property of the spatial hashing scheme, all keys in Set are distinct. The receiver then encodes Set into an OKVS $T \longleftarrow \mathsf{Encode}(Set)$ and sends it to the sender.

For every input $q_i = (q_i^1, \cdots, q_i^d) \in Q$ and its corresponding spatial hash output $\{k_i\} = \mathcal{H}_2(q_i)$, the sender computes $c_{ij} \longleftarrow \mathsf{Decode}(T, k_i \| q_i^j \| j)$ for each dimension $j \in \{1, \cdots, d\}$. Note that if $\|q_i - w_l\| \leq \sigma$ for some $w_l \in W$, then $k_i \in \mathcal{H}_1(w_l)$ due to the locally overlapping property of spatial hashing. Furthermore, since $q_i^j \in [w_i^j - \sigma, w_i^j + \sigma]$ for every $j \in \{1, \cdots, d\}$, the ciphertext c_{ij} will be an encryption of 0. The sender then homomorphically sums the ciphertexts $c_{i1}, \cdots, c_{id}$ and obtains ciphertext c_i, and sends it to the receiver. To realize the functionality $\mathcal{F}_{\text{FPSI-CA}}$, the receiver decrypts each c_i and counts the number of zeros, which corresponds to the number of sender items within the threshold distance of some receiver item. For $\mathcal{F}_{\text{FPSI}}$, both parties invoke an instance of 1-out-of-2 OT for each $i \in \{1, \cdots, M\}$. The receiver inputs 1 as the selection bit if c_i is decrypted to zero, and inputs 0 otherwise; the sender inputs $(\perp, q_i)$, where $\perp$ is a dummy value. The receiver thereby learns q_i once $\|q_i - w_j\| \leq \sigma$ for some $w_j \in W$.

Notice that the OKVS T built by the receiver contains $(2\sigma + 1) d N h_1$ key-value pairs, while the sender only needs to perform dM decodings. Since $N \gg M$, we integrate the sparse OKVS and BatchPIR schemes into our original protocol, allowing the sender to decode efficiently without receiving the entire OKVS, significantly reducing communication overhead. The detailed fuzzy PSI(-CA) protocol is shown in Fig. 6.

We define the security of our fuzzy PSI(-CA) protocol ($\Pi_{\text{Fuzzy PSI(-CA)}}$) in Theorem 5, and a detailed proof is provided in Appendix E.3.

As Protocol 3 is secure in the semi-honest model, its variant achieving the functionalities $\mathcal{F}_{\text{Fuzzy PSI}}$ exhibits during-execution leakage as the receiver learns the cardinality of the fuzzy intersection before the execution completes. The conception of execution leakage was first introduced by a recent PSU paper [20]. We argue that this vulnerability is not severe in our context, as the during-execution leakage in our protocol only reveals the cardinality of the fuzzy intersection, while the execution leakage in [20] refers to the cardinality of the exact intersection set. For the receiver, inferring the sender's specific inputs from this fuzzy intersection cardinality is more challenging, since even if the receiver identifies one of its own inputs as part of the fuzzy intersection, the sender's matching input may have δ^d possible values. We are also actively developing a fuzzy PSI protocol that eliminates such during-execution leakage as future work.

4.2 Unbalanced Fuzzy PSI Protocol when the Sender has a Larger Set

When the sender has a larger set, we propose an alternative fuzzy PSI protocol for $\mathcal{F}_{\text{Fuzzy PSI-SP}}$. This protocol is constructed using the $\mathcal{H}_2$**-unique** spatial hashing schemes. We assume the receiver inputs a set $W = \{w_1, \cdots, w_N\}$ of N items under condition $\mathcal{C}_1$, and the sender inputs a set $Q = \{q_1, \cdots, q_M\}$ of M items under condition $\mathcal{C}_2$, where

PROTOCOL 3. $\Pi_{\text{Fuzzy PSI(-CA)}}$

- **Parameters:** The protocol runs between sender $\mathcal{S}$ and receiver $\mathcal{R}$. Input domain U^d and a distance threshold $\sigma \geq 0$, the cardinalities of both parties' sets N, M. A spatial hashing scheme that achieves $\mathcal{F}_{\mathsf{SH}}$ and $\mathcal{H}_2$-unique under condition $\mathcal{C}_1$ and $\mathcal{C}_2$ for σ. $|\mathcal{H}_1(i)| \leq h_1$ for any $i \in U^d$. A sparse OKVS scheme $(\mathsf{Encode}, \mathsf{Decode})$ with key space $\mathcal{K} = \{0,1\}^*$ and value space $\mathcal{V} = \mathbb{F}$. Let $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$ be the random function used in Encode. Let w denote the Hamming weight of the sparse part. An additively homomorphic encryption scheme $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec}, \mathsf{ASum}, \mathsf{ARef})$ and a BatchPIR scheme.
- **Inputs:**
  - $\mathcal{R}$ inputs $W = \{w_1, \cdots, w_N\} \in U^{d \times N}$ under condition $\mathcal{C}_1$, $\mathcal{S}$ inputs $Q = \{q_1, \cdots, q_M\} \in U^{d \times M}$ under condition $\mathcal{C}_2$.
- **Protocol:**

Phase 1: Spatial Hashing

1. $\mathcal{R}$ and $\mathcal{S}$ calculate the spatial hashing for their inputs locally or by invoking Π_{ISH} according to the condition $\mathcal{C}_1$ and $\mathcal{C}_2$. $\mathcal{R}$ obtains $\mathcal{H}_1(w_i) = \{w_{i1}, \cdots, w_{ih_1}\}$ for every $w_i \in W$ while $\mathcal{S}$ obtains $\{k_i\} = \mathcal{H}_2(q_i)$ for every $q_i \in Q$.

Phase 2: Setup

1. $\mathcal{R}$ generates a fresh key-pair $(pk, sk) \longleftarrow \mathsf{AGen}(\kappa)$ for the additive homomorphic encryption scheme, sends the public key pk to $\mathcal{S}$, and computes $c = \mathsf{AEnc}(pk, 0)$. $\mathcal{R}$ initializes an empty key-value set $Set = \emptyset$.
2. For each $w_i = (w_i^1, \cdots, w_i^d) \in W$ and $\mathcal{H}_1(w_i) = \{w_{i1}, \cdots, w_{ih_1}\}$, $\mathcal{R}$ adds key-value pair $\{w_{ik} \| x \| j, \mathsf{ARef}(c)\}_{x \in [w_i^j - \sigma, w_i^j + \sigma]}$ to Set for $j \in \{1, \cdots, d\}$ and $k \in \{1, \cdots, h_1\}$.
3. $\mathcal{R}$ generates a sparse OKVS $T_0 \| T_1 \longleftarrow \mathsf{Encode}(Set)$, where $T_0 \in \mathbb{F}^s$ and $T_1 \in \mathbb{F}^d$.

Phase 3: Calculating the Fuzzy Intersection

1. $\mathcal{R}$ sends T_1 to $\mathcal{S}$. For every $\mathcal{S}$'s input $q_i = (q_i^1, \cdots, q_i^d) \in Q$, $\{k_i\} = \mathcal{H}_2(q_i)$ and dimension $j \in \{1, \cdots, d\}$, $\mathcal{S}$ computes $r_{ij} = \mathsf{sparse}(k_i \| q_i^j \| j) \in \{0,1\}^s$. $\mathcal{S}$ defines sets $I_{ij} = \{a_{ij1}, \cdots, a_{ijw}\}$ that $r_{ij}^{a_{ijk}} = 1$ for every $k \in \{1, \cdots, w\}$ and $j \in \{1, \cdots, d\}$. The receiver obtains $c_{ijk} = T_0^{a_{ijk}}$ via batchPIR and computes $c_{ij} = \langle T_1, \mathsf{dense}(k_i \| q_{ij} \| j) \rangle + \sum_{k=1}^{w} c_{ijk}$ and $c_i = \mathsf{ASum}(\{c_{ij}\}_{j \in \{1, \cdots, d\}})$. $\mathcal{S}$ sends the set $C = \{c_1, \cdots, c_M\}$ to $\mathcal{R}$.
2. $\mathcal{R}$ initializes an empty string $e = (e_1, \cdots, e_M) = 0^M$ and checks every $c_i \in C$, if c_i is a valid cipher-text and $\mathsf{ADec}(sk, c_i) = 0$, it sets $e_i = 1$.
3. **[PSI-CA]** $\mathcal{R}$ outputs the Hamming weight of string e.
4. **[PSI]** $\mathcal{R}$ and $\mathcal{S}$ invoke M times 1-out-of-2 OT that $\mathcal{R}$ inputs e_i as the receiver and $\mathcal{S}$ inputs $(\perp, q_i)$ as the sender for $i \in \{1, \cdots, M\}$. $\perp$ is a public dummy item. $\mathcal{R}$ outputs the set of non-dummy items received from $\mathcal{S}$.

Figure 6: Fuzzy PSI(-CA) Protocol.

$M \gg N$. According to these conditions, both parties first compute their respective $\mathcal{H}_2$**-unique** spatial hashing outputs either locally or via the interactive spatial hashing protocol. The sender obtains $\mathcal{H}_1(q_i) = \{q_{i1}, \cdots, q_{ih_1}\}$ for each $q_i \in Q$ while the receiver obtains $\{k_i\} = \mathcal{H}_2(w_i)$ for each $w_i \in W$ in the spatial hashing phase.

In the setup phase, the sender first generates a fresh AHE key pair (pk, sk) and shares the public key pk with the receiver. The sender initializes an empty key-value set $Set = \emptyset$. For every item $q_i = (q_i^1, \cdots, q_i^d) \in W$ and its corresponding spatial hashing output $\mathcal{H}_1(q_i) = \{q_{i1}, \cdots, q_{ih_1}\}$, the sender adds the key-value pairs $\{q_{ij}, (\mathsf{AEnc}(pk, q_i^1), \cdots, \mathsf{AEnc}(pk, q_i^d))\}$ to Set for each $j \in \{1, \cdots, h_1\}$. The sender then encodes Set into an OKVS $T \longleftarrow \mathsf{Encode}(Set)$ and sends it to the receiver. Although the values in Set consist of d ciphertexts and are not naturally elements of a finite field $\mathbb{F}$ as required by OKVS constructions, we apply an invertible mapping to convert each ciphertext into a binary bit string. In practice, these binary strings are concatenated to form a single binary field element, which can be encoded into an OKVS.

For every input $w_i = (w_i^1, \cdots, w_i^d) \in W$ and its corresponding spatial hash output $\{k_i\} = \mathcal{H}_2(w_i)$, the receiver computes $c_i \leftarrow \mathsf{Decode}(T, k_i)$ and parses the result as d ciphertexts $(e_i^1, \cdots, e_i^d)$. The receiver then generates d random masks $mask_i^1, \cdots, mask_i^d$ from U^d, and defines $mask_i = (mask_i^1, \cdots, mask_i^d)$, $w_i' = (w_i^1 + mask_i^1, \cdots, w_i^d + mask_i^d)$. Next, the receiver computes $e_i^j = \mathsf{ASum}(e_i^j, \mathsf{AEnc}(pk, mask_i^j))$ for $j \in \{1, \cdots, d\}$ and sends $\{e_i = (e_i^1, \cdots, e_i^d)\}_{i \in \{1, \cdots, N\}}$ to the sender.

The sender decrypts e_i and obtains $p_i = (p_i^1, \cdots, p_i^d) = (\mathsf{ADec}(sk, e_i^1), \cdots, \mathsf{ADec}(sk, e_i^d))$. Notice that if $k_i \in \mathcal{H}_1(q_j)$ for some $q_j \in Q$, then e_i represents an encryption of the vector $q_j + mask_i$, implying $p_i = q_j + mask_i$ and $w_i' = w_i + mask_i$. Therefore, both parties need only invoke an instance of $\mathcal{F}_{\mathsf{match}}$

PROTOCOL 4. $\Pi_{\text{Fuzzy PSI-SP}}$

- **Parameters:** The protocol runs between sender $\mathcal{S}$ and receiver $\mathcal{R}$. Input domain U^d and a distance threshold $\sigma \geq 0$, the cardinalities of both parties' sets N, M. A spatial hashing scheme that achieves $\mathcal{F}_{\mathsf{SH}}$ and $\mathcal{H}_2$-unique under condition $\mathcal{C}_1$ and $\mathcal{C}_2$ for σ. $|\mathcal{H}_1(i)| \leq h_1$ for any $i \in U^d$. A sparse OKVS scheme $(\mathsf{Encode}, \mathsf{Decode})$ with key space $\mathcal{K} = \{0,1\}^*$ and value space $\mathcal{V} = \mathbb{F}$. Let $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$ be the random function used in Encode. Let w denote the Hamming weight of the sparse part. An additively homomorphic encryption scheme $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec}, \mathsf{ASum}, \mathsf{ARef})$ and a BatchPIR scheme.
- **Inputs:**
  - $\mathcal{R}$ inputs $W = \{w_1, \cdots, w_N\} \in U^{d \times N}$ under condition $\mathcal{C}_2$, $\mathcal{S}$ inputs $Q = \{q_1, \cdots, q_M\} \in U^{d \times M}$ under condition $\mathcal{C}_1$.
- **Protocol:**

Phase 1:Spatial Hashing

1. $\mathcal{R}$ and $\mathcal{S}$ calculate the spatial hashing for their inputs locally or by invoking Π_{ISH} according to the condition $\mathcal{C}_1$ and $\mathcal{C}_2$. $\mathcal{S}$ obtains $\mathcal{H}_1(q_i) = \{q_{i1}, \cdots, q_{ih_1}\}$ for every $q_i \in Q$ while $\mathcal{R}$ obtains $\{k_i\} = \mathcal{H}_2(w_i)$ for every $w_i \in W$.

Phase 2: Setup

1. $\mathcal{S}$ generates a fresh key-pair $(pk, sk) \longleftarrow \mathsf{AGen}(\kappa)$ for the additive homomorphic encryption scheme and sends the public key pk to $\mathcal{R}$. $\mathcal{S}$ initializes an empty key-value set $Set = \emptyset$.
2. For each $q_i = (q_i^1, \cdots, q_i^d) \in Q$ and $\mathcal{H}_1(q_i) = \{q_{i1}, \cdots, q_{ih_1}\}$, $\mathcal{S}$ adds key-value pairs $\{q_{ij}, (\mathsf{AEnc}(pk, q_i^1), \cdots, \mathsf{AEnc}(pk, q_i^d))\}_{j \in \{1, \cdots, h_1\}}$ to Set.
3. $\mathcal{S}$ generates a sparse OKVS $T_0 \| T_1 = \longleftarrow \mathsf{Encode}(Set)$, where $T_0 \in \mathbb{F}^s$ and $T_1 \in \mathbb{F}^d$.

Phase 3: Calculating the Fuzzy Intersection

1. $\mathcal{S}$ sends T_1 to $\mathcal{R}$. For every $\mathcal{R}$'s input $w_i \in W$ and $k_i = \mathcal{H}_2(w_i)$, $\mathcal{R}$ calculates $r_i = \mathsf{sparse}(k_i) \in \{0,1\}^s$. $\mathcal{R}$ defines a set $I_i = \{a_{i1}, \cdots, a_{iw}\}$ that $r_i^{a_{ij}} = 1$ for every $j \in \{1, \cdots, w\}$. The receiver obtains $c_{ij} = T_0^{a_{ij}}$ via batchPIR and computes $c_i = \langle T_1, \mathsf{dense}(k_i) \rangle + \sum_{j=1}^{w} c_{ij}$.
2. $\mathcal{R}$ parses c_i as d cipher-text $(e_i^1, \cdots, e_i^d)$ and samples $mask_i = (mask_i^1, \cdots, mask_i^d) \xleftarrow{R} U^d$. $\mathcal{R}$ computes $e_i = (\mathsf{ASum}(e_i^1, \mathsf{AEnc}(pk, mask_i^1)), \cdots, \mathsf{ASum}(e_i^d, \mathsf{AEnc}(pk, mask_i^d)))$ and $w_i' = (w_i^1 + mask_i^1, \cdots, w_i^d + mask_i^d)$. $\mathcal{R}$ sends $\{e_i\}_{i \in \{1, \cdots, N\}}$ to $\mathcal{S}$.
3. $\mathcal{R}$ and $\mathcal{S}$ invoke $\mathcal{F}_{\mathsf{match}}$ N times that $\mathcal{R}$ inputs w_i' and $\mathcal{S}$ inputs $\mathsf{ADec}(e_i)$ for $i \in \{1, \cdots, N\}$. $\mathcal{R}$ adds w_i into intersection set once $\mathcal{F}_{\mathsf{match}}$ output 1 to $\mathcal{R}$.

Figure 7: Fuzzy PSI-SP Protocol.

to allow the receiver to determine whether $\|w_i - q_j\| \leq \sigma$. To minimize communication overhead, we also integrate sparse OKVS and BatchPIR techniques (as described in Sec. 3.3) into this protocol. The complete fuzzy PSI-SP protocol is shown in Fig. 7.

We define the security of our fuzzy PSI-SP protocol ($\Pi_{\text{Fuzzy PSI-SP}}$) in Theorem 6, and a detailed proof is provided in Appendix E.4.

4.3 Complexity Analysis

In our fuzzy PSI protocol, the sender decodes dM keys from the OKVS T, which has size $O(Ndh_1\sigma)$, using BatchPIR techniques. This constitutes the dominant communication cost in the protocol. Sending the dense part of the OKVS T_1 contributes $O(\log(Ndh_1\sigma))$ to the communication, while the BatchPIR operations incur $O(dM\sqrt[3]{\log(Ndh_1\sigma)})$ complexity. Additional communication arises from Step 2 of Phase 3, where the sender returns M ciphertexts to the receiver. When both parties compute the spatial hashing output via the interactive protocol, $h_1 = 1$ and the communication overhead in Phase 1 is $Md\sqrt[3]{N\sigma}$, leading to an overall communication complexity of $O(dM\sqrt[3]{\log(Nd\sigma)})$.

For our PSI-SP protocol, the receiver decodes the OKVS T of size $O(Mh_1\sigma)$ for N keys using BatchPIR. Since each decoded value consists of d ciphertexts, the communication cost for this step is $O(dN\sqrt[3]{\log(Mh_1\sigma)})$. In Phase 3, both parties invoke N instances of $\mathcal{F}_{\mathsf{match}}$, contributing $O(Nd\sigma)$ to the communication. When the spatial hashing outputs are computed interactively, Phase 1 incurs $Nd\sqrt[3]{M\sigma}$ overhead, resulting in an overall communication complexity of $O(Nd\sqrt[3]{M\sigma} + Nd\sigma)$. Notice that in our fuzzy PSI protocol, we assume $N \gg M$ while in our fuzzy PSI-SP protocol, we assume $M \gg N$. Both of our protocols achieve sublinear communication complexity relative to the size of the larger set.

Prot.	(d,σ)	Input size (n_1,n_2)			
		$(2^{18},2^5)$		$(2^{20},2^5)$	
		Comm.	Time	Comm.	Time
[11]	$(2,10)$	763.29	711.07	>2000	-
	$(6,10)$	2241.87	>2000	>6000	-
	$(2,30)$	2171.47	>2000	>6000	-
Ours (AHE)	$(2,10)$	0.34	8.62	0.34	32.64
	$(6,10)$	0.34	22.98	0.34	90.65
	$(2,30)$	0.35	23.11	0.35	173.13

Table 2: Communication cost (in MB) and running time (in seconds) of the online phase of our AHE-based interactive spatial hashing protocol with the protocol from [11], with varying dimensions d, distance threshold σ, and input sizes of P_1 and P_2, n_1 and n_2."-" indicates that the experiment could not be conducted due to insufficient memory.

5 Implementation and Performance

We provide implementation details and evaluate our interactive spatial hashing protocols and fuzzy PSI protocols in comparison to existing protocols within unbalanced settings.

5.1 Implementation Details

Environment. We run all experiments on a single machine with Intel Xeon Gold 6338 CPU and 503 GB RAM. We measure the running time of the online phase of both our protocols under WAN settings (100 Mbps and 10 Mbps bandwidth with 80ms RTT latency). All of our experiments use a single process, where each party in the protocol uses a separate thread.

Implementation. We choose the computational security parameter $\kappa = 128$ and the statistical security parameter $\lambda = 40$. We implement[*] our protocols in C++. We employ the 3-Hash Blazing-fast OKVS[†] in [30] as the sparse OKVS in our protocols. We utilize vectorized BatchPIR [25], supporting batched PIR queries[‡] with low communication costs. Our implementation relies on the OT implementation provided in LibOTe[§]. We also use the Intel Paillier Cryptosystem Library[¶] to implement the Paillier partially homomorphic encryption scheme as the AHE, and instantiate our universal hash functions with BLAKE3 to ensure high performance and cryptographic security. We adopt Coproto[‖] to realize network communication.

[*] https://anonymous.4open.science/r/unbalanced-fuzzy-PSI-B81F

[†] https://github.com/Visa-Research/volepsi

[‡] https://github.com/mhmughees/vectorized_batchpir

[§] https://github.com/osu-crypto/libOTe

[¶] https://github.com/intel/paillliercryptolib

[‖] https://github.com/Visa-Research/coproto

5.2 Performance Comparison

Interactive Spatial Hashing Protocols. We evaluate the efficiency of our interactive spatial hashing protocol utilizing the AHE-based PVS protocol $\Pi_{\mathsf{PVS}_{\mathsf{AHE}}}$ (Fig. 5) and compare it with the interactive spatial hashing protocol from [11] under unbalanced settings. The experiment demonstrates that for the parameter set $(n_1,n_2,d,\sigma) = (2^{18},2^5,2,10)$ with a 100 Mbps bandwidth, our protocol significantly outperforms the baseline. Our solution demonstrates superior efficiency by requiring only 0.34 MB of total communication, resulting in a $2000\times$ reduction compared to their protocol. Furthermore, our protocol completes its online phase in just 8.62 seconds, achieving an $82\times$ speedup in online runtime. These results demonstrate that our protocol reduces both the amount of transmission data and the time required, making it more efficient for real-world applications. Our OPRF-based protocol is less efficient than our AHE-based one. Its online phase incurs longer running times due to the computational overhead of encoding the OKVS. The detailed experiment result is shown in Tab. 2.

We experimentally evaluate our fuzzy PSI(-CA) protocol $\Pi_{\text{Fuzzy PSI(-CA)}}$ (Fig. 6) using both non-interactive and interactive spatial hashing schemes under various set conditions. It should be noted that in this protocol, the receiver is required to encode an OKVS of size $O(Ndh_1\sigma)$, where $h_1 = 2^d$ under our non-interactive scheme when the distance threshold $T = 4\sigma$. Although employing the BatchPIR technique avoids directly sending the OKVS and reduces communication costs, it still imposes high offline computation and memory overhead on the receiver for large d. In contrast, our interactive scheme maintains $h_1 = 1$. This distinction aligns with our earlier claim that the non-interactive scheme is suited for low-dimensional inputs, whereas the interactive scheme is designed for high-dimensional cases. Accordingly, in the following experiments, we provide results for $d = 2$ when using the non-interactive spatial hashing, and for $d = \{2,6\}$ when using the interactive spatial hashing.

Fuzzy PSI(-CA) protocol. When the receiver's set is at least $T = 3\sigma$ apart, we compare our protocol with the protocol from Baarsen et al. [34], which requires the receiver's set to be at least 2σ apart. For the parameter set $(N,M,d,\sigma) = (2^{18},2^5,2,10)$, experimental results show that their protocol requires 1008 MB of online communication, whereas ours achieves a $2880\times$ reduction in communication overhead. Our protocol also performs better in low-bandwidth environments, reducing the online running time by $5.7\times$ under 10 Mbps bandwidth.

When the receiver's set is $T = 4\sigma$ apart, we compare our protocol with Baarsen et al.'s protocol under the same condition. For parameters $(N,M,d,\sigma) = (2^{20},2^5,2,10)$ and 100 Mbps bandwidth, our protocol requires only 0.35 MB of communication, compared to 16,128 MB for theirs. In addition, our protocol achieves a $1.33\times$ speedup over their 2σ protocol

Prot.	(d,σ)	Input size (N,M)											
		$(2^{16},2^5)$			$(2^{18},2^5)$			$(2^{18},2^8)$			$(2^{20},2^5)$		
		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)	
			100Mbps	10Mbps		100Mbps	10Mbps		100Mbps	10Mbps		100Mbps	10Mbps
Fuzzy PSI protocols from non-interactive spatial hashing													
[34] (2σ apart)	$(2,10)$	252	20.63	202.05	1008	82.74	825.37	1008	83.11	826.03	4032	334.27	3276.55
	$(2,30)$	732	59.09	587.10	2928	237.36	2354.96	2928	237.89	2355.41	-	-	-
Ours (3σ apart)	$(2,10)$	0.35	29.04	29.33	0.35	144.52	144.80	0.512	153.38	153.80	-	-	-
	$(2,30)$	0.35	85.50	85.78	0.35	446.41	446.70	-	-	-	-	-	-
[34] (4σ apart)	$(2,10)$	1008	80.79	808.39	4032	334.59	3332.03	4032	336.17	3335.70	16128	1293.41	>3600
	$(2,30)$	2928	234.91	2347.50	11712	942.93	>3600	11712	943.30	>3600	46848	>3600	>3600
Ours (4σ apart)	$(2,10)$	0.35	12.33	12.62	0.35	53.50	53.78	0.51	53.09	53.51	0.35	252.21	252.50
	$(2,30)$	0.35	41.06	41.07	0.35	155.81	156.09	0.51	210.74	211.16	-	-	-
Fuzzy PSI protocols from interactive spatial hashing													
[11]	$(2,10)$	1692	358.40	1586.19	6769	1427.93	>3600	-	-	-	-	-	-
	$(6,10)$	5065	993.61	>3600	20259	>3600	>3600	-	-	-	-	-	-
	$(2,30)$	4904	706.20	>3600	19617	2824.74	>3600	-	-	-	-	-	-
Ours	$(2,10)$	0.96	7.18	7.97	0.96	32.12	32.91	1.18	32.04	33.01	0.96	142.40	143.19
	$(6,10)$	1.00	22.41	23.22	1.00	105.65	106.47	1.42	101.03	102.19	1.00	485.17	485.99
	$(2,30)$	0.96	25.27	26.05	0.96	100.84	101.63	1.18	104.64	105.61	0.96	468.47	469.26

Table 3: Communication cost and running time of the online phase of our Fuzzy PSI protocol with the protocols from [34] and [11], with varying conditions on the input sizes and spatial hashing methods, dimensions d, distance threshold σ, and input sizes of the receiver and the sender, N and M. "-" indicates that the experiment could not be conducted due to insufficient memory.

and a $5\times$ speedup over their 4σ protocol in terms of online running time.

We also evaluate the efficiency of the online phase of our Fuzzy PSI(-CA) protocol, using our interactive spatial hashing method, in comparison to the protocol by Gao et al. [11], excluding the interactive spatial hashing and setup phases of both protocols. Across various parameter sets, our protocol achieves at least a $31\times$ improvement in online runtime and a $1762\times$ reduction in communication overhead. The results further indicate that our protocol, implemented via the interactive spatial hashing approach, incurs a lower computational cost and offers better suitability for high-dimensional data compared to the same protocol using the non-interactive scheme. A detailed performance comparison is provided in Tab. 3.

Fuzzy PSI-SP protocol. We also evaluate our Fuzzy PSI-SP protocol $\Pi_{\text{Fuzzy PSI-SP}}$ (Fig. 7) in scenarios where the sender holds a larger set. When the sender's set is $T = 4\sigma$ apart, we compare our protocol with the PSI-SP protocol by Piske et al. [29], both using a non-interactive spatial hashing scheme. Under the parameter set $(N,M,d,\sigma) = (2^5,2^{20},2,10)$ and 100 Mbps bandwidth, our protocol requires only 1.09 MB of communication and 11.36 seconds in the online phase. This represents a $283\times$ improvement in communication and a $101\times$ improvement in online runtime compared to their protocol. We further test the efficiency of our Fuzzy PSI-SP protocol using our interactive spatial hashing protocol. Although Gao et al. [11] suggested that their fuzzy PSI protocol could be extended to support fuzzy PSI-SP via interactive spatial hashing, no implementation was provided. Therefore, we report the performance of our protocol under various parameter settings using our interactive approach. Detailed results are presented in Tab. 4.

Acknowledgments

This work is supported by National Key R&D Program of China No. 2023YFC3305501.

Ethical Considerations

Multi-party computation enables parties to jointly compute a function using their private data without revealing it. The computation can proceed only with the explicit consent of all participating parties. Our proposed fuzzy PSI protocol also requires the awareness and consent from both parties. Lacking such awareness and consent could lead to misuse of our protocols in real-world scenarios such as the following:

In applications such as compromised credential checking, a non-expert client may not fully understand that our protocol's guarantees rely on a semi-honest threat model, and a malicious server could exploit this misunderstanding to obtain additional information. In illegal content detection, a server could mislead a user about what the protocol is for. The user, thinking it's a normal check, might agree to run it and accidentally leak private information.

To ensure the positive use of this technique, two key conditions must be met. First, its assumptions and limitations must be clearly communicated to non-technical stakeholders, and strong protections must be implemented to prevent deception. For instance, before obtaining user authorization, the software must clearly and standardly disclose the specific

Prot.	(d,σ)	Input size (N,M)											
		$(2^5,2^{18})$			$(2^8,2^{18})$			$(2^5,2^{20})$			$(2^8,2^{20})$		
		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)		Comm. (MB)	Time (s)	
			100Mbps	10Mbps		100Mbps	10Mbps		100Mbps	10Mbps		100Mbps	10Mbps
Fuzzy PSI-SP protocols from non-interactive spatial hashing													
[29]	(2,10)	77.62	224.30	239.87	77.67	222.83	240.51	309.93	1147.77	1358.98	309.99	1141.06	1596.57
	(6,10)	258.08	797.15	821.66	259.97	846.28	890.93	1030.79	4542.01	4895.68	1032.68	4517.23	4996.40
	(2,30)	77.62	235.81	241.49	77.67	226.45	243.42	309.93	1155.61	1618.54	309.99	1157.72	1644.64
	(6,30)	258.08	835.87	936.82	259.97	855.24	875.75	1030.79	4900.78	5068.67	1032.68	4987.88	5127.50
Ours	(2,10)	1.09	3.51	4.41	6.51	4.02	9.35	1.09	11.36	12.25	6.51	12.06	17.39
	(6,10)	2.61	136.67	138.81	18.64	136.96	152.21	-	-	-	-	-	-
	(2,30)	2.47	3.60	5.62	17.51	5.65	19.97	2.47	11.98	14.00	17.51	12.62	26.95
	(6,30)	6.74	131.25	136.76	51.64	134.26	176.51	-	-	-	-	-	-
Fuzzy PSI-SP protocols from interactive spatial hashing													
Ours	(2,10)	1.71	23.91	25.31	7.18	24.43	30.30	1.71	103.07	104.46	7.18	115.09	120.96
	(6,10)	3.24	70.67	73.32	19.42	72.17	88.06	3.24	310.86	313.52	19.42	313.84	329.73
	(2,30)	3.08	68.90	71.42	18.18	71.78	86.65	3.08	313.68	316.20	18.18	312.42	327.29
	(6,30)	7.37	218.12	224.15	52.42	227.42	270.31	7.37	950.21	956.24	52.42	980.52	1023.41

Table 4: Communication cost and running time of the online phase of our Fuzzy PSI-SP protocol with the protocol from [29], with varying spatial hashing methods, dimensions d, distance threshold σ, and input sizes of the receiver and the sender, N and M. "-" indicates that the experiment could not be conducted due to insufficient memory.

database to be matched and the security model of the protocol. Second, independent and public audits of both the protocol's code/algorithms and the server's operational logs should be mandated to detect misuse or malicious behavior that falls outside the security assumptions. Finally, judicial authorization should be required before any new database is added to the system, ensuring that each matching target meets stringent legal and ethical standards.

We argue that when implemented within a transparent governance and compliance framework, this approach offers a net-positive impact. Its core utility lies in enabling efficient, privacy-preserving matching for unbalanced sets, which is critical for many real-world applications. While potential risks such as misunderstandings of security assumptions or deceptive use exist, they are substantially mitigated by the integrated safeguards previously outlined. Consequently, under these appropriate safeguards, the value of the technique demonstrably outweighs its risks.

Open Science

The code implemented in this article is published in : https://doi.org/10.5281/zenodo.18223806.

A Related Work

The fuzzy PSI and fuzzy matching problems were first formally introduced by Freedman et al. [10]. Their protocol focuses on Hamming distance, relying on additively homomorphic encryption and polynomial interpolation. Their protocol was proven to be insecure by Chmielewski et al. [6]. The following study on fuzzy PSI [4,6,17,33,35] also focused on Hamming distance. The basic idea of those protocols is to construct a fuzzy matching protocol from oblivious polynomial evaluation or a homomorphic encryption technique, and extend it to a fuzzy PSI protocol by invoking the fuzzy matching protocol for every pair of inputs from both parties. Those protocol requires $O(MN)$ communication and computation cost.

To reduce the quadratic complexity, Garimella et al. [13] introduced a novel cryptographic primitive, spatial hashing, which allows two parties to hash every input item to a set of elements. Once two items from different parties are close enough for L_∞ distance, the hashing sets of those two items intersect. Via the spatial hashing and function secret sharing techniques, they propose the first fuzzy PSI protocol for L_∞ distances without $O(MN)$ communication cost. However, the spatial hashing always has restrictions on at least one party's input set to guarantee that every hashing output set of one of the parties, namely P_1, is disjoint. The receiver's input set is required to be 2σ apart in their protocol. They propose another fuzzy PSI protocol for malicious security [14], sacrificing the efficiency of the protocol. They also proposed another spatial hashing scheme for overlapped inputs [12]. The scheme neglects the requirement for the receiver's inputs, but has the possibility of failing.

Baarsen and Pu [34] extended the spatial hashing schemes from Garimella et al. [13]. They proposed two schemes when the receiver's inputs are 2σ and 4σ apart. They also provided fuzzy matching schemes for $L_{p\in[1,\infty]}$ distance from the DDH tuple and a framework to extend them to fuzzy PSI protocols

for $L_{p\in[1,\infty]}$ distance, utilizing the OKVS technique. Their protocol outperforms the existing protocols, which are based on function secret sharing. Although those spatial hashing schemes can be computed locally by both parties, the hashing output set size of P_1 is 2^d, resulting in significant overhead for utilizing them in the fuzzy PSI protocol for high-dimensional input. Gao et al. [11] have proposed another efficient interactive spatial hashing scheme that both parties compute their hashing outputs from a protocol. Their scheme optimizes the hashing output set size of P_1 to be 1, but has a stronger restriction on both parties' inputs. Their protocol requires that both parties' sets be "separated" on at least one dimension. They also proposed a new framework for constructing the fuzzy PSI protocol. Piske et al. [29] introduced a new cryptographic primitive, distance-aware random OT, which is similar to fuzzy matching, and provided a protocol to achieve the fuzzy PSI-SP functionality. Their protocol utilized the existing spatial hashing method, which requires the input set of receivers to be 4σ aparts. Although those fuzzy PSI protocols from the spatial hashing technique achieve better efficiency, they are all constructed under the balanced setting and have a communication complexity at least linear to both parties' input size.

B Preliminary

B.1 Additively Homomorphic Encryption

Homomorphic encryption is a form of encryption that allows computations to be performed on encrypted data without first having to decrypt it. An additively homomorphic encryption scheme consists of the following probabilistic polynomial-time algorithms:

AGen: Given a security parameter κ, $\mathsf{AGen}(\kappa)$ outputs a public-private key pair (pk, sk), and specifies a message space $\mathcal{M}$.

AEnc: Given the public key pk and a plaintext message $m \in \mathcal{M}$, one can compute a ciphertext $\mathsf{AEnc}(pk, m)$, an encryption of m under pk.

ADec: Given the secret key sk and a ciphertext $\mathsf{AEnc}(pk, m)$, ADec is to recover a plaintext m.

ASum: Given the public key pk and a set of ciphertexts $\{\mathsf{AEnc}(pk, m_i)\}$ which are the encryption of messages $\{m_i\}$, one can homomorphically compute a ciphertext which is the encryption of the sum of the underlying messages: $\mathsf{AEnc}(pk, \sum_i m_i) = \mathsf{ASum}(\{\mathsf{AEnc}(pk, m_i)\}_i)$

ARefresh: One can randomize ciphertexts using a randomized procedure denoted as $\mathsf{ARefresh}$.

The commonly used additively homomorphic encryption schemes include Paillier encryption [27], Exponential ElGamal encryption [8], and Ring-LWE-based encryption schemes [2, 3, 9]. We depend on the standard concept of CPA security in encryption, which essentially implies that, without the private key sk, encrypted messages are computationally indistinguishable from one another.

B.2 Batch Private Information Retrieval

A BatchPIR scheme consists of three algorithms, all taking the computational security parameter κ as an implicit input:
$\mathsf{Query}(I) \to (qu, st)$: on input a set of distinct indexes $I = \{i_1, \cdots, i_b\} \subseteq \{1, \cdots, n\}$, outputs a query qu and a private state st including the index set.
$\mathsf{Answer}(DB, qu) \to ans$: on input the database DB and the query qu, outputs an answer ans.
$\mathsf{Recover}(st, ans) \to \{d_1, \cdots, d_b\}$: given the state st and the answer ans, outputs a batch of entries $\{d_1, \cdots, d_b\}$.

The BatchPIR is correct once for any dataset DB and all distinct inputs $I = \{i_1, \cdots, i_b\}$ and $\mathsf{Query}(I) \to (qu, st)$, $\mathsf{Recover}(st, \mathsf{Answer}(DB, qu)) = \{DB^{i_1}, \cdots, DB^{i_b}\}$.

The BatchPIR requires that the client's query qu reveals no information about the query indexes. Note that the BatchPIR does not aim to protect the privacy of the server's dataset DB. Formally, a BatchPIR scheme is secure once for all PPT adversaries $\mathcal{A}$, and all distinct batch query sets I_1, I_2 that $|I_1| = |I_2|$, $|\Pr[\mathcal{A}(qu = 1) \mid \mathsf{Query}(I_1) \to (qu, st)] - \Pr[\mathcal{A}(qu = 1) \mid \mathsf{Query}(I_2) \to (qu, st)]| \le \mathsf{negl}(\kappa)$.

B.3 Oblivious Key-Value Store

Definition 2. *An **Oblivious Key-Value Store (OKVS)** is parameterized by a set $\mathcal{K}$ of keys, a set $\mathcal{V}$ of values, and consists of two algorithms:*
Encode $\left(\{k_i, v_i\}_{i=\{1,\cdots,n\}}\right)$*: Input a set of (k_i, v_i) key-value pairs and outputs an object T (or, with statistically small probability, an error indicator $\perp$).*
Decode (T, k)*: Input an object T, a key k, outputs a value v.*

Correctness. For all $A \subseteq \mathcal{K} \times \mathcal{V}$ with distinct keys: $(k, v) \in A$ and $\perp \neq T \leftarrow \mathsf{Encode}(A) \Longrightarrow \mathsf{Decode}(T, k) = v$
Obliviousness. For all distinct $\{k_1^0, \ldots, k_n^0\}$ and all distinct $\{k_1^1, \ldots, k_n^1\}$, if Encode does not output $\perp$ for $(k_1^0, \ldots, k_n^0)$ or $(k_1^1, \ldots, k_n^1)$, then the output of $\mathsf{Exp}^{\mathcal{A}}\left(\mathcal{K} = (k_1^0, \ldots, k_n^0)\right)$ is computationally indistinguishable to that of $\mathsf{Exp}^{\mathcal{A}}\left(\mathcal{K} = (k_1^1, \ldots, k_n^1)\right)$, where:

$\mathsf{Exp}^{\mathcal{A}}(\mathcal{K} = (k_1, \ldots, k_n))$:
(1) for $i \in \{1, \cdots, n\}$: choose uniform $v_i \leftarrow \mathcal{V}$
(2) return $\mathcal{A}(\mathsf{Encode}(\{(k_1, v_1), \ldots (k_n, v_n)\}))$

In our construction, we also require the OKVS to meet the properties of randomness, linearity, and sparsity.
Randomness. For all sets of m distinct keys $\{k_1, \ldots, k_m\} \subseteq \mathcal{K}$, m random values $\{v_1, \ldots, v_m\} \subseteq \mathcal{V}$ and any $k \notin \{k_1, \ldots, k_m\}$, $\mathsf{Decode}(T, k)$ is statistically indistinguishable from a uniformly random element in $\mathcal{V}$ where $T \leftarrow \mathsf{Encode}(\{(k_1, v_1), \ldots, (k_m, v_m)\})$.
linearity. An OKVS is linear (over field $\mathbb{F}$) if $\mathcal{V} = \mathbb{F}$ ("values" are elements of $\mathbb{F}$), the output of Encode is a vector T in $\mathbb{F}^m$, and the Decode function is defined as : $\mathsf{Decode}(T, k) = \langle \mathrm{d}(k), T \rangle = \sum_{j=1}^{m} \mathrm{d}(k)_j T_j$ for some function $\mathrm{d} : \mathcal{K} \to \mathbb{F}^m$. Hence $\mathsf{Decode}(\cdot, k)$ is a linear map from $\mathbb{F}^m$ to $\mathbb{F}$.

sparsity. An OKVS is sparse once the output of the Encode function T can be divided into two parts $T = T_0 \| T_1$, where $T_0 \in \mathbb{F}^s$ and $T_1 \in \mathbb{F}^d$ and $d = o(s)$. There are two functions involving the Decode function, $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$. $\mathsf{Decode}(T,k) \stackrel{\text{def}}{=} \langle T_0, \mathsf{sparse}(k)\rangle + \langle T_1, \mathsf{dense}(k)\rangle$ for decoding the sparse OKVS T with any key $k \in \mathcal{K}$. The Hamming weight of the output of the sparse function is a fixed value w.

B.4 Multi-point Oblivious Pseudorandom Function (mp-OPRF)

Multi-point oblivious pseudorandom function (mp-OPRF) [19] allows the receiver with inputs $X = \{x_1, \ldots, x_n\}$ to learn the OPRF outputs $\{F(k,x_1), \ldots, F(k,x_n)\}$, where the key k is input by the sender. The detailed functionality of mp-OPRF is shown in Fig. 8.

Functionality 4. (mp-OPRF - $\mathcal{F}_{\text{mp-OPRF}}$)

- **Parameters:** Two parties receiver and sender. A PRF $F : \{0,1\}^\kappa \times \{0,1\}^l \to \{0,1\}^l$.
- **Inputs:** The sender inputs the PRF key $k \in \{0,1\}^\kappa$ and the receiver $\mathcal{R}$ inputs the set $X = \{x_1, \ldots, x_n\}$ where $x_i \in \{0,1\}^l$.
- **Outputs:** Output $\{F(k,x_1), \cdots, F(k,x_n)\}$ to the receiver.

Figure 8: Ideal functionality for $\mathcal{F}_{\text{mp-OPRF}}$.

B.5 Fuzzy Matching for L_∞ Distance

Functionality 5. (Fuzzy matching - $\mathcal{F}_{\text{match}}$)

- **Parameters:** Two parties receiver and sender. An input domain U^d. A distance threshold σ.
- **Inputs:** The receiver and sender input x and y from U^d,respectively.
- **Outputs:** Output 1 to the receiver once $\|x-y\|_\infty \le \sigma$ and 0 otherwise.

Figure 9: Ideal functionality for fuzzy matching.

C OPRF-based Private Value Sum Protocol

In this section, we propose a protocol that achieves functionality $\mathcal{F}_{PVS}$ from the OPRF technique. We suppose the receiver inputs m keys $x_1, \cdots, x_m \in U$ and the sender inputs m sets $Y_1, \cdots, Y_m$, each set contains n key-value pairs. At the beginning of the protocol, the sender initializes m empty key-value sets $Set_1, \cdots, Set_m = \emptyset$ and selects m random values $\hat{r}_1, \cdots, \hat{r}_m \in \mathbb{F}$ such that $\sum_{i=1}^m \hat{r}_i = 0$. For each $i \in \{1, \cdots, d\}$ and every key-value pair $\{k, v\} \in Y_i$, the sender adds the pair $\{k\|i, v + \hat{r}_i\}$ to the set Set.

The sender and receiver then invoke $\mathcal{F}_{\text{OPRF}}$, where the receiver inputs $x_1\|1, \cdots, x_m\|m$ and obtains m corresponding PRF outputs $F(k,x_1\|1), \cdots, F(k,x_m\|m)$, while the sender obtains the PRF key k. For every key-value pair $\{k_{ij}, v_{ij}\} \in Y_i$, the sender adds the pair $\{k_{ij}\|i, v_{ij} + F(k,k_{ij}\|i) + \hat{r}_i\}$ to Set_i. The sender encodes each set Set_i into an OKVS $T_i \longleftarrow \mathsf{Encode}(Set_i)$ and sends them to the receiver. The receiver decodes each OKVS T_i at the key $x_i\|i$, obtaining $c_i = \mathsf{Decode}(T_i, x_i\|i) - F(k,x_i\|i)$. Finally, the receiver computes $c = \sum_{i=1}^m c_i$. If $\{x_i, v_i\} \in Y_i$ for every $i \in \{1, \cdots, m\}$, then $c_i = v_i + \hat{r}_i$ and $\sum_{i=1}^m c_i = \sum_{i=1}^m v_i$. Otherwise, c is a random value in $\mathbb{F}$.

Notice that the receiver only needs to decode one key in every OKVS. We also apply similar BatchPIR and sparse OKVS techniques to avoid sending the OKVS directly and reduce the communication cost. The detailed OPRF-based private value sum protocol is shown in Fig. 10.

We define the security of our OPRF-based PVS protocol ($\Pi_{\text{PVS}_{\text{OPRF}}}$) in Theorem 4, and a detailed proof is provided in Appendix E.2.

D Probability of Separation under Uniform Distribution

In this section, we analyze the probabilities that the assumptions "set separated on at least one dimension" and "set separated on every dimension" hold in our protocol. We compute these probabilities under the assumption that elements are uniformly distributed. Since the domain length 2^u is sufficiently large, we approximate the discrete distribution over $\{0, 1, \ldots, 2^u\}$ by the continuous uniform distribution on the interval $[0, 2^u]$.

Theorem 1. *Let $U^d = [0,2^u]^d$ and $P = \{p_1, \ldots, p_n\} \subseteq U^d$ consist of n independent uniformly distributed points in U^d. Fix a distance threshold $\sigma > 0$. If $\frac{2\sigma(n-1)}{2^u} < 1$, then the probability that P is separated on every dimension is $\left(1 - \frac{2\sigma(n-1)}{2^u}\right)^{nd}$, and the probability that P is separated on at least one dimension is $1 - \left(1 - \left(1 - \frac{2\sigma(n-1)}{2^u}\right)^n\right)^d$.*

Proof. We begin with the one-dimensional case. Consider n i.i.d. uniform random variables $X_1, \ldots, X_n \sim \text{Unif}[0, L]$, with $L = 2^u$. Let $X_{(1)} < \cdots < X_{(n)}$ denote the order statistics. We say that the set is *separated* on this coordinate if $X_{(i+1)} - X_{(i)} > 2\sigma, \forall i = 1, \ldots, n-1$. The joint density of the order statistics is constant $n!$ over the simplex $0 < x_{(1)} < \cdots < x_{(n)} < L$.

The event that all gaps exceed 2σ is equivalent to $0 < x_{(1)} < x_{(2)} - 2\sigma < \cdots < x_{(n)} - (n-1)2\sigma < L - (n-1)2\sigma$. Define shifted variables $Z_i = x_{(i)} - (i-1)2\sigma$. Then the constraint becomes $0 < Z_1 < \cdots < Z_n < L - (n-1)2\sigma$. The Jacobian of this linear transformation is 1, so the volume of the feasible region is $\frac{(L-(n-1)\cdot 2\sigma)^n}{n!}$, provided $L > (n-1)2\sigma$. Since the den-

PROTOCOL 5. Private Value Sum Protocol from OPRF. $\Pi_{\mathsf{PVS}_{\mathsf{OPRF}}}$

- **Parameters:** Two parties: the receiver and the sender. Input domain U. A finite field $\mathbb{F}$. Set number m and set size n. A sparse OKVS scheme $(\mathsf{Encode}, \mathsf{Decode})$ with key space $\mathcal{K} = \{0,1\}^*$ and value space $\mathcal{V} = \mathbb{F}$. Let $\mathsf{sparse} : \{0,1\}^* \to \{0,1\}^s$ and $\mathsf{dense} : \{0,1\}^* \to \{0,1\}^d$ be the random functions used in Encode. Let w denote the Hamming weight of the sparse part. A BatchPIR scheme.
- **Inputs:**
  - The receiver inputs m keys $x_1, \cdots, x_m \in U$.
  - The sender inputs m sets of key-value pairs $Y_1, \cdots, Y_m$, where $Y_i = \{k_{ij}, v_{ij}\}_{j \in \{1,\cdots,n\}}$ that $k_{ij} \in U$ and $v_{ij} \in \mathbb{F}$.
- **Protocol:**

  Offline Phase

  1. The sender initializes m empty key-value sets $Set_1, \cdots, Set_m = \emptyset$ and generates $\hat{r}_1, \cdots, \hat{r}_m \in \mathbb{F}$ s.t. $\sum_{i=1}^m \hat{r}_i = 0$.

  Online Phase

  1. The sender and receiver invoke $\mathcal{F}_{\mathsf{OPRF}}$ that the receiver inputs $\{x_i \| i\}_{i=\{1,\cdots,m\}}$ and obtains $\{F(k, x_i \| i)\}_{i=\{1,\cdots,m\}}$ while the sender obtains the OPRF key k.
  2. For every $\{k_{ij}, v_{ij}\} \in Y_i$, the sender adds $\{k_{ij} \| i, v_{ij} + F(k, k_{ij}) + \hat{r}_i\}$ to Set_i, and computes $T_i = T_{0i} \| T_{i1} \longleftarrow \mathsf{Encode}(Set_i)$ for $i = \{1, \cdots, m\}$, where $T_{i0} \in \mathbb{F}^s$ and $T_{i1} \in \mathbb{F}^d$. The sender sends $T_{11}, \cdots, T_{m1}$ to the receiver.
  3. The receiver computes $r_i = \mathsf{sparse}(x_i \| i) \in \{0,1\}^s$ for $i \in \{1, \cdots, m\}$. Every r_i has a Hamming weight of w. The receiver defines a set $I_i = \{a_{i1}, \cdots, a_{iw}\}$ that $r_i^{a_{ij}} = 1$ for every $j \in \{1, \cdots, w\}$. The receiver obtains $c_{ia_{ij}} = T_{i0}^{a_{ij}}$ via batchPIR and computes $c_i = \langle T_{i1}, \mathsf{dense}(x_i \| i) \rangle + \sum_{j=1}^w c_{ia_{ij}} - F(k, x_i \| i)$ and $c = \sum_{i=1}^m c_i$.
  4. The receiver outputs c.

Figure 10: OPRF-based Private Value Sum Protocol.

sity of order statistics is $n!$, the probability that all gaps exceed 2σ is $\Pr(\text{separated in one dimension}) = \left(1 - \frac{2\sigma(n-1)}{L}\right)^n$.

Now extend to d dimensions. Since coordinates are independent, the probability that separation holds simultaneously in all d coordinates is $\left(1 - \frac{2\sigma(n-1)}{L}\right)^{nd}$. On the other hand, since events across dimensions are independent, the probability that separation occurs in at least one coordinate is $1 - \prod_{k=1}^d \left(1 - \left(1 - \frac{2\sigma(n-1)}{L}\right)^n\right) = 1 - \left(1 - \left(1 - \frac{2\sigma(n-1)}{L}\right)^n\right)^d$. This yields the claimed formulas. □

Let $p_1 := \left(1 - \frac{2\sigma(n-1)}{L}\right)^n$ denote the separation probability on a single dimension.

- The probability that P is separated on every dimension is then p_1^d, which under the condition $\frac{2\sigma(n-1)}{L} \ll 1$ can be approximated by $p_1^d \approx \exp\left(-\frac{2\sigma n(n-1)}{L} d\right)$.

- The probability that P is separated on at least one dimension is $1 - (1-p_1)^d$, which under the same condition can be approximated by $1 - (1-p_1)^d \approx 1 - \exp(-d\, p_1)$.

E Security Proofs

E.1 Security for Spatial Hashing

Theorem 1. *The non-interactive spatial hashing scheme in Sec. 3.1 realizes the functionality $\mathcal{F}_{SH}$.*

Proof. Since our scheme is non-interactive, we only need to prove the correctness. we first show that $|\mathcal{H}_1(x)| \le h_1 = h^d$ for every $x \in U^d$. This is because $2\sigma + 1$ side-length L_∞ ball will intersect with at most $\lceil \frac{2\sigma+1}{l} \rceil$ cells on every dimension when $l \le 2\sigma$ and 2 cells when $l > 2\sigma$, thus $\mathcal{H}_1(x)$ includes at most h^d distinct labels.

For the *Locally Overlapping* property, when $\|x - y\|_\infty \le \sigma$, $y \in [x_1 - \sigma, x_1 + \sigma] \times \cdots \times [x_d - \sigma, x_d + \sigma]$. Noticing that $\mathcal{H}_1(x)$ contains all the labels of the cells which the $2\sigma + 1$ side-length L_∞ ball centered at x intersects with, it derives directly that $cell_l(y) \in \mathcal{H}_1(x)$. $\mathcal{H}_1(x) \bigcap \mathcal{H}_2(y) \neq \emptyset$.

For the *Non-Conflicting* property, if there exist any two points x, y that $\|x - y\|_\infty \ge T$ and $\mathcal{H}_1(x) \bigcap \mathcal{H}_1(y) \neq \emptyset$, let $B_x = [x_1 - \sigma, x_1 + \sigma] \times \cdots \times [x_d - \sigma, x_d + \sigma]$, $B_y = [y_1 - \sigma, y_1 + \sigma] \times \cdots \times [y_d - \sigma, y_d + \sigma]$, then there will be at least one cell $C = [c_1, c_1 + l) \times \cdots \times [c_d, c_d + l)$ that $B_x \bigcap C \neq \emptyset$ and $B_y \bigcap C \neq \emptyset$. For every dimension $i \in \{1, \cdots, d\}$, $c_i - \sigma \le x_i < c_i + l + \sigma$ and $c_i - \sigma \le y_i < c_i + l + \sigma$, thus $|x_i - y_i| < l + 2\sigma = T$ for every dimension. In this case, $\|x - y\|_\infty < T$, which contradicts the assumption that $\|x - y\|_\infty \ge T$. Our scheme is also $\mathcal{H}_2$-**unique** since $\mathcal{H}_2(y)$ only output one item $cell_l(y)$. □

Theorem 2. *The protocol in Fig. 4 securely realizes the functionality $\mathcal{F}_{SH}$ in the $\mathcal{F}_{PVS}$ hybrid model, in the presence of semi-honest adversaries.*

Proof. **Correctness.** For the *Locally Overlapping* property, if there exist $x = (x^1, \ldots, x^d) \in X$ and $y = (y^1, \ldots, y^d) \in Y$ such that $\|x - y\|_\infty \le \sigma$, then for each $i \in [1, d]$, we have $y^i \in K_{Y_i}$. By the definition of $\mathcal{F}_{\mathrm{PVS}}$, it follows that $\mathcal{H}_1(x) = \sum_{j=1}^d Y_j[x^j] = \mathcal{H}_2(y)$.

For the *Non-Conflicting* property, since the set X is separated on at least one dimension, any distinct $p, q \in X$ differ in at least one coordinate of their associated random values. Hence, the probability of collision is negligible.

Privacy. In the protocol, the transcript consists solely of $\{c_i\}_{i\in[n_2]}$ obtained by P_2 from $\mathcal{F}_{\text{PVS}}$. Therefore, it suffices to prove security against a corrupted P_2. We construct a simulator that samples $\{c_i\}_{i\in[n_2]}$ uniformly at random and sends them to P_2, and then prove indistinguishability.

Specifically, since the set Y is separated on every dimension, in particular, for the dimension k on which the set X is separated, Y is also separated in dimension k. Hence, the values $\{y_i^k\}_{i\in[n_2]}$ correspond to distinct intervals in S_k (or, if they do not lie in any such interval, $\mathcal{F}_{\text{PVS}}$ outputs a random value). Therefore, $\{c_i\}_{i\in[n_2]}$ is computationally indistinguishable from a set of uniformly random samples. □

E.2 Security for PVS

Theorem 3. *Assume that the additively homomorphic encryption scheme is IND-CPA secure, then the protocol in Fig. 5 securely realizes the functionality $\mathcal{F}_{PVS}$ in the presence of semi-honest adversaries.*

Proof. **Correctness.** If, for all $i \in [m]$, we have $x_i \in K_{Y_i}$, then $t = \sum_{i=1}^{m} Y_i[x_i] + \mathsf{mask}$, and hence $o = \sum_{i=1}^{m} Y_i[x_i]$. If there exists some j such that $x_j \notin K_{Y_j}$, then by the randomness property of the OKVS, decoding using x_j yields a uniformly random value. Consequently, the final output is uniformly random.

Privacy. The sender's view consists only of the transcript from the PIR protocol and the ciphertext masked by the receiver in the final step. By the security of the PIR protocol and the randomness of the mask, security against a corrupted sender is guaranteed.

The receiver's view consists of the public key pk sent by the sender, the dense part of the OKVS sent by the sender, the transcript from the PIR protocol, and the value t obtained in the final step. It suffices to prove the security of the protocol in the case where the sender sends the complete OKVS $\{T_i\}_{i\in[m]}$. By the randomness of the AHE ciphertexts and the obliviousness of the OKVS, the OKVS sent by the sender is indistinguishable; $t = o + mask$ can be derived from the output and the receiver's random tape. Therefore, security against a corrupted receiver is guaranteed. □

Theorem 4. *Assume that the additively homomorphic encryption scheme is IND-CPA secure, the underlying OPRF protocol is secure in semi-honest model, then the protocol in Fig. 10 securely realizes the functionality $\mathcal{F}_{PVS}$ in the presence of semi-honest adversaries.*

Proof. It is easy to see that when $x_i \in K_{Y_i}$ for all $i \in [m]$, we have $o = \sum_{i=1}^{m} Y_i[x_i]$. Conversely, if there exists some j such that $x_j \notin K_{Y_j}$, then by the randomness of the OPRF and the randomness property of the OKVS, the final output is uniformly random. Thus, the correctness is guaranteed.

For a corrupted sender, its view consists of the OPRF key k and the transcript from the PIR protocol. Therefore, security follows from the security of the PIR protocol.

For a corrupted receiver, its view consists of: its OPRF values $\{F(k, x_i\|i)\}_{i\in[m]}$, the dense part of the OKVS sent by the sender, and the transcript from the PIR protocol. It suffices to prove the security of the protocol in the case where the sender sends the complete OKVS $\{T_i\}_{i\in[m]}$. By the security of the OPRF, $\{F(k, x_i\|i)\}_{i\in[m]}$ is pseudorandom from the receiver's perspective. Furthermore, by the security of the OPRF, the randomness of $\hat{r}_i$, and the obliviousness property of the OKVS, $\{T_i\}_{i\in[m]}$ is indistinguishable from a uniformly random set. Therefore, security is guaranteed. □

E.3 Security for Fuzzy PSI

Theorem 5. *Assume that the additively homomorphic encryption scheme is IND-CPA secure, the underlying spatial hash protocol is secure in semi-honest model, then the protocol in Fig. 6 securely realizes the functionality $\mathcal{F}_{\text{Fuzzy PSI(-CA)}}$ in the presence of semi-honest adversaries.*

Proof. **Correctness.** If there exist $w_i \in W$ and $q_j \in Q$ such that $\|w_i - q_j\| \le \sigma$, then by the locally overlapping property of Spatial Hashing we have $\mathcal{H}_1(w_i) \cap \mathcal{H}_2(q_j) \neq \perp$. Let the intersection element be w_{ik}. In this case, $\mathcal{R}$ encodes in the OKVS the key-value pairs $\{ w_{ik}\|q_j^m\|m, \mathsf{ARef}(c) \}_{m\in[d]}$. Therefore, when $\mathcal{S}$ decodes, it obtains the ciphertext of 0, so $e_j = 0$, and the PSI(-CA) protocol produces the correct output.

On the other hand, for any $q_j \in Q$ such that there does not exist $w_i \in W$ with $\|w_i - q_j\| \le \sigma$:

If $\mathcal{H}_2(q_j)$ has empty intersection with $\{\mathcal{H}_1(w_i)\}_{i\in[N]}$, then by the random decoding property of the OKVS, the probability that $\mathcal{R}$ obtains $e_j = 0$ is negligible.

If there exists some w_i such that $\mathcal{H}_1(w_i) \cap \mathcal{H}_2(q_j) \neq \perp$, then since $\|w_i - q_j\| > \sigma$, there exists a dimension $m \in [d]$ such that $\mathcal{H}_2(q_j)\|q_j^m\|m$ does not appear in the key-value pairs encoded in the OKVS. Hence, the probability that $e_j = 0$ is again negligible.

In conclusion, the correctness of the protocol is guaranteed.

Privacy. For a corrupted receiver, its view consists of: the transcript from the PIR protocol, the ciphertexts $c = \{c_1, \ldots, c_M\}$, and the transcript of the OT protocol used in [PSI]. The security of the PIR transcript is guaranteed by the PIR protocol. The ciphertexts $c = \{c_1, \ldots, c_M\}$ can be simulated from the intersection output (by generating ciphertexts of 0 for the number of elements in the intersection and padding the rest with random ciphertexts). The security of the OT transcript is guaranteed by the OT protocol.

For a corrupted sender, its view consists of: the public key pk, the dense part of the OKVS sent by the sender, the transcript from the PIR protocol (again, it suffices to prove

security in the case where the entire OKVS is sent), and the transcript of the OT protocol used in [PSI]. Since the values in the OKVS are additively homomorphic ciphertexts, by the security of the encryption scheme these values are pseudorandom, and hence the OKVS satisfies the obliviousness property. The security of the OT transcript is guaranteed by the OT protocol. □

E.4 Security for Fuzzy PSI-SP

Theorem 6. *Assume that the additively homomorphic encryption scheme is IND-CPA secure, the underlying spatial hash protocol is secure in semi-honest model, then the protocol in Fig. 7 securely realizes the functionality* $\mathcal{F}_{\text{Fuzzy PSI-SP}}$ *in the* $\mathcal{F}_{\text{match}}$ *hybrid model, in the presence of semi-honest adversaries.*

Proof. **Correctness.** If there exist $q_i \in Q$ and $w_j \in W$ such that $\|q_i - w_j\| \leq \sigma$, then by the locally overlapping property of Spatial Hashing we have $\mathcal{H}_1(q_i) \cap \mathcal{H}_2(w_j) \neq \perp$. In this case, $\mathcal{R}$ uses the hash intersection to decode and parse the ciphertext corresponding to $q_i = (q_i^1, \ldots, q_i^d)$. Thus, $\mathsf{ADec}(e_i)$ is within distance σ of w_i'. By the definition of $\mathcal{F}_{\mathsf{match}}$, the receiver obtains 1, and therefore the final output is the correct intersection.

On the other hand, for $w_i \in W$ such that no element of Q lies within distance σ: If the spatial hash output of Q does not contain $\mathcal{H}_2(w_i)$, then by the random decoding property of the OKVS, the probability that $\mathcal{F}_{\mathsf{match}}$ outputs a match is negligible. If the spatial hash output of Q does contain $\mathcal{H}_2(w_i)$, then since the distance is greater than σ, the output is 0, yielding the correct result.

Privacy. For a corrupted sender, its view consists of: the transcript from the PIR protocol and the masked ciphertexts $\{e_i\}_{i\in[N]}$. The security of the PIR transcript is guaranteed by the PIR protocol, while the security of the ciphertexts follows from the randomness of the mask.

For a corrupted receiver, its view consists of: the public key *pk*, the dense part of the OKVS sent by the sender (again, it suffices to prove security in the case where the entire OKVS is sent), the transcript from the PIR protocol, and the output of $\mathcal{F}_{\mathsf{match}}$. The security of the OKVS follows from the security of the encryption scheme and the obliviousness property, while the output of $\mathcal{F}_{\mathsf{match}}$ can be derived directly from the protocol output. Therefore, security is guaranteed. □

References

[1] Sebastian Angel, Hao Chen, Kim Laine, and Srinath Setty. Pir with compressed queries and amortized query processing. In *2018 IEEE symposium on security and privacy (SP)*, pages 962–979. IEEE, 2018.

[2] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[3] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.

[4] Anrin Chakraborti, Giulia Fanti, and Michael K Reiter. {Distance-Aware} private set intersection. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 319–336, 2023.

[5] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious prf. In *Annual International Cryptology Conference*, pages 34–63. Springer, 2020.

[6] Lukasz Chmielewski and Jaap-Henk Hoepman. Fuzzy private matching. In *Third International Conference on Availability, Reliability and Security*, pages 327–334. IEEE, 2008.

[7] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[8] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[9] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*, 2012.

[10] Michael J Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *International conference on the theory and applications of cryptographic techniques*, pages 1–19. Springer, 2004.

[11] Ying Gao, Lin Qi, Xiang Liu, Yuanchao Luo, and Longxin Wang. Efficient fuzzy private set intersection from fuzzy mapping. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 36–68. Springer, 2025.

[12] Gayathri Garimella, Benjamin Goff, and Peihan Miao. Computation efficient structure-aware psi from incremental function secret sharing. In *Annual International Cryptology Conference*, pages 309–345. Springer, 2024.

[13] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *Annual International Cryptology Conference*, pages 323–352. Springer, 2022.

[14] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In *Annual International Cryptology Conference*, pages 577–610. Springer, 2023.

[15] Meng Hao, Weiran Liu, Liqiang Peng, Hongwei Li, Cong Zhang, Hanxiao Chen, and Tianwei Zhang. Unbalanced {Circuit-PSI} from oblivious {Key-Value} retrieval. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6435–6451, 2024.

[16] Laura Hetz, Thomas Schneider, and Christian Weinert. Scaling mobile private contact discovery to billions of users. In *European Symposium on Research in Computer Security*, pages 455–476. Springer, 2023.

[17] Piotr Indyk and David Woodruff. Polylogarithmic private approximations and efficient matching. In *Theory of Cryptography: Third Theory of Cryptography Conference*, pages 245–264. Springer, 2006.

[18] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Batch codes and their applications. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 262–271, 2004.

[19] Stanisław Jarecki and Xiaomin Liu. Fast secure computation of set intersection. In *International Conference on Security and Cryptography for Networks*, pages 418–435. Springer, 2010.

[20] Yanxue Jia, Shi-Feng Sun, Hong-Sheng Zhou, and Dawu Gu. Scalable private set union, with stronger security. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6471–6488, 2024.

[21] Daniel Kales, Christian Rechberger, Thomas Schneider, Matthias Senker, and Christian Weinert. Mobile private contact discovery at scale. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1447–1464, 2019.

[22] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious prf with applications to private set intersection. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 818–829, 2016.

[23] Anunay Kulshrestha and Jonathan Mayer. Identifying harmful media in {End-to-End} encrypted communication: Efficient private membership computation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 893–910, 2021.

[24] Jian Liu, Jingyu Li, Di Wu, and Kui Ren. Pirana: Faster multi-query pir via constant-weight codes. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4315–4330. IEEE, 2024.

[25] Muhammad Haris Mughees and Ling Ren. Vectorized batch private information retrieval. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 437–452. IEEE, 2023.

[26] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, Dan Boneh, et al. Location privacy via private proximity testing. In *NDSS*, volume 11, 2011.

[27] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[28] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Spot-light: lightweight private set intersection from sparse ot extension. In *Annual International Cryptology Conference*, pages 401–431. Springer, 2019.

[29] Lucas Piske, Jaspal Singh, Ni Trieu, Vladimir Kolesnikov, and Vassilis Zikas. Distance-aware ot with application to fuzzy psi. *Cryptology ePrint Archive*, 2025.

[30] Srinivasan Raghuraman and Peter Rindal. Blazing fast psi from improved okvs and subfield vole. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pages 2505–2517, 2022.

[31] Peter Rindal and Phillipp Schoppmann. Vole-psi: fast oprf and circuit-psi from vector-ole. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 901–930. Springer, 2021.

[32] Emin Islam Tatlı. Cracking more password hashes with patterns. *IEEE Transactions on Information Forensics and Security*, 10(8):1656–1665, 2015.

[33] Erkam Uzun, Simon P Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. Fuzzy labeled private set intersection with applications to private {Real-Time} biometric search. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 911–928, 2021.

[34] Aron van Baarsen and Sihang Pu. Fuzzy private set intersection with large hyperballs. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 340–369. Springer, 2024.

[35] Qingsong Ye, Ron Steinfeld, Josef Pieprzyk, and Huaxiong Wang. Efficient fuzzy matching and intersection on private datasets. In *International Conference on Information Security and Cryptology*, pages 211–228. Springer, 2009.

[36] Cong Zhang, Yu Chen, Weiran Liu, Liqiang Peng, Meng Hao, Anyu Wang, and Xiaoyun Wang. Unbalanced private set union with reduced computation and communication. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1434–1447, 2024.