Assumption-Free Fuzzy PSI via Predicate Encryption

Erik-Oliver Blass
Airbus

Guevara Noubir
Northeastern University

Abstract

We present the first protocol for efficient Fuzzy Private Set Intersection (PSI) that achieves linear communication complexity, does not depend on restrictive assumptions on the distribution of party inputs, and abstains from inefficient fully homomorphic encryption. Specifically, our protocol enables two parties to compute all pairs of elements from their respective sets that are within a given Hamming distance, without constraints on how these sets are structured. Our key insight is that securely computing the (threshold) Hamming distance between two inputs can be reduced to securely computing their inner product. Leveraging this reduction, we construct a Fuzzy PSI protocol using recent techniques for inner-product predicate encryption. To enable the use of predicate encryption in our setting, we establish that these predicate encryption schemes only require a weak notion of simulation security. We also demonstrate how their internal key derivation can be efficiently distributed without a trusted third party.

As a result, our Fuzzy PSI on top of predicate encryption achieves optimal linear communication complexity for arbitrary input distributions. Our implementation validates its feasibility and demonstrates improved performance over the most closely related work.

1 Introduction

Private Set Intersection (PSI) is an increasingly popular approach to enable collaborations in a variety of data-driven tasks. Given two parties, each with a set of elements, PSI allows the computation of the intersection of the two sets without revealing any additional information to the parties. Introduced in seminal works such as Meadows [45] and Freedman et al. [27], PSI has since garnered significant interest both from the research community and industry. This has led to efficient PSI protocols as well as adoption and deployments by major companies including Google [39], Meta [12], and Microsoft [46]. It turns out that communication complexity between parties typically represents the primary performance bottleneck in real-world scenarios, as PSI computations often process data in batches rather than real-time [39]. So, current state-of-the-art in PSI features optimal linear communication complexity in the size of the parties' input sets, see [17, 31, 43, 56] for an overview.

While traditional PSI identifies exact matches between elements, many real-world applications require finding elements within a distance threshold. Examples include a client matching fingerprints against a biometric database, querying security logs with traffic features, analyzing GPS coordinates, or identifying DNA sequence variations.

The idea of relaxing the element equality constraint, generally referred to as *Fuzzy PSI* (FPSI) was initially also mentioned by Freedman et al. [27]. In FPSI, two parties (sender and receiver) hold their own set of vectors. A pair of vectors, one from each set, is considered to be in the intersection if the distance between them is below a predefined threshold. However, as naïve solutions result in exponential communication cost (in the data dimension), efficient solutions were left for future work. It is only lately that FPSI has seen a revival of interest due to the applications needs and the industry's general interest and adoption of PSI techniques.

Many solutions have recently been proposed that significantly improve the communication and computation complexity of FPSI protocols [21, 29, 32, 33, 57, 63, 65]. Yet, current solutions achieve linear communication complexity in the dataset size, the data dimension, and the distance threshold only by making strong assumptions about input data distributions. These structure-aware PSI approaches require specific data properties for input sets, such as minimum distance thresholds between elements or distinct element differences across dimensions. While current approaches demonstrate high effectiveness when applied to datasets that conform to their underlying structural assumptions, no existing solution achieves linear communication complexity for *arbitrarily* distributed input data. However, in cases where input data is unpredictable, deviates from idealized distributions, or fails to meet strict minimum distance thresholds, FPSI solutions for arbitrary input distributions are essential.

Table 1: Comparison of asymptotic communication and computation complexities. For protocols with multiple variants, we summarize lower bounds to highlight key parameters. S: FPSI sender, R: FPSI receiver, n_S, n_R: number of vectors from S and R (denoted as n when $n_S = n_R$), ℓ: vector length, t: threshold, B_1, B_2 : FHE parameters.

Protocol	Metric	Assumption	Communication	Computation
[64]	Hamming	FPR/FNR	$O(\ell n_S n_R B_1)$	$S: O(\ell n_S n_R B_2)$ $R: O(\binom{\ell}{t} n_R)$
[16]	Hamming & L_1	FPR	$O(n_S n_R t^2)$	$S: O((\ell+t^2) n_S n_R)$ $R: O((\ell+t) n_S n_R)$
[29]	Hamming	R. UniqC	$O(\ell^2 n_S + \ell t n_R)$	$S: O(\ell^2 n_S)$ $R: O(\ell^2 n_S + \ell t n_R)$
	L_∞	$R \wedge S$. disj. proj.	$O(\ell t (n_S + n_R))$	$S: O(\ell t n_S + n_R)$ $R: O(n_S + \ell t n_R)$
	L_p	$R \wedge S$. disj. proj.	$O((\ell t + p \log t) n_S + \ell t n_R)$	$S: O((\ell t + p \log t) n_S + n_R)$ $R: O(n_S + \ell t n_R)$
[21]	Hamming (generalizes)	$d(x,y) \le t$ or $d(x,y) \ge \delta t$, $\delta > 3$	$O(n^{1+\frac{1}{\delta-1}})$	$S/R: O(n^{1+\frac{1}{\delta-1}})$
[32]	L_∞	n_R receiver balls: radius t, separated $c \cdot t$, $c > 2$	$O((4 \log t)^\ell n_R + n_S)$	$S: O((2t)^\ell n_R)$ $R: O((2 \log t)^\ell n_S))$
		$c > 4$	$O(2^\ell \ell n_R \log t + n_S)$	$S: O((2t)^\ell n_R)$ $R: O(\ell n_S \log t)$
	L_∞	$\exists$ disj. proj.	$O(\ell n_R \log t + n_S)$	$S: O((2t)^\ell n_R)$ $R: O(\ell n_S \log t)$
[65]	L_p, L_∞	n_R receiver balls: radius t, separated $c \cdot t$, $c > 2$	$O(t \ell n_R + 2^\ell n_S)$	$S: O(2^\ell t n_S)$ $R: O(t \ell n_R + 2^\ell n_S))$
		$c > 4$	$O(t 2^\ell \ell n_R + n_S)$	$S: O(\ell n_S)$ $R: O(t 2^\ell \ell n_R + n_S))$
	L_∞	$\exists$ disj. proj.	$O((t\ell)^2 n_R + n_S)$	$S: O(\ell^2 n_S)$ $R: O((t\ell)^2 n_S + n_R)$
	L_p	$c > 2t(\ell^{\frac{1}{p}} + 1)$	$O(t^p n_S + t 2^\ell \ell n_R)$	$S: O((\ell + t^p) n_S)$ $R: O(n_S + t 2^\ell \ell n_R))$
[57]	L_1, L_2, L_∞	Disjoint Hash, $0 \le s \le \ell$	$\Omega(\ell(n_S 2^s + n_R 2^{\ell - s}))$	$S: \Omega(\ell n_S 2^s)$ $R: \Omega(\ell n_R 2^{\ell-s})$
[66]	L_∞	2δ-apart on all dim.	$O((n_S + n_R)(\log t)^\ell)$	$O((n_S + n_R)(\log t)^\ell)$
	L_p		$O(n_S(\ell + p \log t) + n_R(\ell t + p \log t))$	$O(n_S(\ell + p \log t) + n_R(\ell t + p \log t))$
Ours	Hamming	**None**	$O(\ell(n_S + t n_R))$	$S: O(\ell(n_S + t n_R))$ $R: O(\ell t n_S n_R)$

– FPR/FNR: assumes that receiver can tolerate non-negiligible false positive/negative rate.
– R. UniqC: assumes that for each vector of R there exist at least $t+1$ dimensions s.t. on each of these dimensions this vector has a unique value different from all other elements of R.
– R. disj. proj.: assumes that for each vector $\mathbf{y}$ of R there exists at least one dimension j on which $|\mathbf{y}[j] - \mathbf{y}'[j]| > 2t$ for all other elements $\mathbf{y}'$ of R.
– $R \wedge S$. disj. proj.: assumes that disj. proj. assumption holds for sender and receiver sets.

This paper introduces an efficient FPSI protocol achieving linear communication complexity for arbitrary input distributions. Our protocol computes the intersection of vectors from two parties within a Hamming distance threshold t. By leveraging inner-product predicate encryption, we reduce (private) fuzzy intersection computation to (private) testing whether vector inner-products match specific values. Pairwise testing to verify that one party's inputs are within a specified Hamming distance of the other party's inputs can be performed offline, eliminating the need for any communication. Instead, one party simply sends encryptions of their input vectors, and the other party obtains decryption keys for each of their input vectors, leading to both linear communication complexity and concrete practicality.

While we discuss related work in great detail later in Section 5, we compare the main features of our protocol to related work in Table 1. In summary, our **technical highlights** are:

- We present the first scheme to securely realize fuzzy private set intersection for Hamming distance, featuring linear communication complexity in the size of parties' input sets, the data dimension, and the distance threshold. Our techniques do not make any restrictive assumption on the structure or distribution of the parties' sets.

- Towards practicality, we show that inner-product predicate encryption, when used in our scheme, only needs to satisfy a weak notion of selective security to achieve full simulation security of fuzzy PSI. This weak notion covers both a game-based formulation (IND-WSS) and a simulation-based formulation (Sim-WSS). We prove, first, that IND-WSS $\Rightarrow$ Sim-WSS. Surprisingly, we then prove that today's concretely practical selectively secure schemes are already IND-WSS secure and thus usable as a building block, i.e., they are composable.

- We design a two-party, distributed, and concretely practical version of the key derivation scheme for the predicate encryption scheme by Park [55]. A two-party key derivation instead of relying on a trusted third party is an important building block in our main construction.

- To show concrete practicality of our techniques, we implement and benchmark them. Our code is openly available [6].

PARAMETERS: Number n_S of input vectors $\mathbf{x}_i$ from Sender S, number n_R of input vectors $\mathbf{y}_i$ from Receiver R where $\mathbf{x}_i, \mathbf{y}_i \in \{0,1\}^\ell$, vector length ℓ, threshold t

1. Wait for input $\mathsf{In}_S = (\mathbf{x}_1, \ldots, \mathbf{x}_{n_S})$ from sender S and $\mathsf{In}_R = (\mathbf{y}_1, \ldots, \mathbf{y}_{n_R})$ from receiver R.
2. Output $\mathsf{Out}_R = \{(\mathbf{x}_i, \mathbf{y}_j) | \mathbf{x}_i \in \mathsf{In}_S, \mathbf{y}_j \in \mathsf{In}_R \text{ s.t. } \mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j) < t\}$ to R.

Figure 1: Ideal fuzzy PSI functionality $\mathcal{F}_{\mathsf{FPSI}}$

Other metrics: We discuss potential extensions of Hamming distance to other metrics in Appendix E.

Computational complexity: A key design decision in our protocol is the trade-off between communication and computation. We explicitly prioritize achieving minimal, linear communication complexity for arbitrary data distributions which necessitates a quadratic $O(n_S n_R)$ computational cost at the receiver. This cost arises from the pairwise testing required when no structural assumptions can be leveraged to prune the search space. While this computational complexity is a known barrier when working on unstructured data [16, 64], we argue it is a justified trade-off in several important scenarios.

First, in many real-world deployments, particularly distributed systems, communication is the dominant performance bottleneck and cost driver [39]. Computation is local, parallelizable, and benefits from Moore's Law, while network bandwidth is constrained by physical infrastructure. Our protocol is designed for these communication-bound environments. Second, the protocol is practical for asymmetric use-cases, where a client with a small set queries a large server database. Examples include contact discovery on mobile devices, biometric authentication against a server, or checking a small list of threat indicators against a large security log. In these scenarios, the quadratic cost remains well within feasible limits. Our evaluation (Section 4) provides concrete evidence for the protocol's efficiency in these contexts. Our primary contribution is therefore a new, optimal design point for FPSI that prioritizes linear communication and assumption-free deployment, addressing a critical gap in the literature for communication-constrained and asymmetric applications.

1.1 Our results in a nutshell

This paper addresses the secure computation of Fuzzy Private Set Intersection (Fuzzy PSI), formalized as an ideal functionality in Figure 1. In this setting, a sender S holds an input set $\mathsf{In}_S = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_S}\}$, and a receiver R holds an input set $\mathsf{In}_R = \{\mathbf{y}_1, \ldots, \mathbf{y}_{n_R}\}$. Each element $\mathbf{x}_i$ and $\mathbf{y}_j$ is a binary vector of length ℓ, i.e., $\mathbf{x}_i, \mathbf{y}_j \in \{0,1\}^\ell$. The goal is to compute the fuzzy intersection of In_S and In_R, defined as all pairs $(\mathbf{x}_i \in \mathsf{In}_S, \mathbf{y}_j \in \mathsf{In}_R)$ such that their Hamming distance $\mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j)$ is below a threshold t. The crucial security requirement is that R learns only the fuzzy intersection, while S learns nothing about R's input.

We propose a new protocol Π_{FPSI} that securely realizes the ideal functionality $\mathcal{F}_{\mathsf{FPSI}}$ from Figure 1. Our construction follows two main steps: (1) we design Π_{FPSI} assuming the existence of a black-box inner product functionality, and (2) realizing this black-box functionality through *inner product predicate encryption* techniques.

Constructing Π_{FPSI} from Inner Products: Assume access to a black-box functionality that, given input vectors $\mathbf{x}$ from S and $\mathbf{y}$ from R, outputs to R whether the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ equals a threshold τ. Beyond this output, R does not learn anything about S' input, and S does not learn anything about R's input.

If we have such a black-box functionality, the idea is then to exploit a well-known relation between the Hamming distance of two vectors and their inner product [42]. Roughly speaking, sender S creates a new vector $\mathbf{x}'$ out of $\mathbf{x}$, and Receiver R a new vector $\mathbf{y}'$ out of $\mathbf{y}$ such that $\mathsf{HD}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}', \mathbf{y}' \rangle$ holds. So, one converts the problem of testing whether $\mathbf{x}$ and $\mathbf{y}$ have Hamming distance τ to the problem of testing whether the inner product of $\mathbf{x}'$ and $\mathbf{y}'$ equals τ.

Building on this reduction, we determine whether for the two vectors $\mathbf{x}$ and $\mathbf{y}$ their Hamming distance is less than a threshold t by computing for $\tau \in \{0, \ldots, t\}$ whether $\langle \mathbf{x}', \mathbf{y}' \rangle = \tau$. So in conclusion, we construct our fuzzy PSI protocol Π_{FPSI} by iterating over inner product tests.

At this point, we omit (crucial) subtleties about R learning the exact Hamming distance instead of only learning whether the Hamming distance is less than t and refer to Section 2 for full details on how we overcome them. Also, we relegate aspects such as the need for the black box inner product functionality revealing $\mathbf{x}$ in case $\langle \mathbf{x}', \mathbf{y}' \rangle = \tau$ to Section 2.

Secure, Efficient Two-Party Inner Product Computation: The second step is to actually build such a black-box inner product test functionality described above with the goal of achieving communication complexity linear in sizes n_S and n_R of the parties' input sets (as well as ℓ and t).

We employ a sub-type of functional encryption called predicate encryption for inner product predicates. This encryption allows one party to encrypt a message m under a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$ to obtain ciphertext c. Another party with vector $\mathbf{y} \in \mathbb{Z}_p^\ell$ and corresponding secret key $sk_{\mathbf{y}}$ can decrypt c to retrieve m if and only if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. If $\langle \mathbf{x}, \mathbf{y} \rangle \neq 0$, decryption fails, revealing no information about m or $\mathbf{x}$ beyond the inequality of the inner product. The idea is that S encrypts each $\mathbf{x}_i$ with itself ($m = \mathbf{x}_i$) and sends the resulting ciphertexts to R. Receiver R obtains decryption keys $sk_{\mathbf{y}_j}$ for each $\mathbf{y}_j$ and tests whether they can decrypt each ciphertext, yielding a simplified version of the black-box we want. The communication cost of these steps is linear in n_S, n_R, ℓ, and t.

However, using predicate encryption again presents two technical challenges which we overcome. First, practical predicate encryption schemes are only selectively secure under game-based definitions. This renders secure composition as part of our main construction Π_{FPSI} difficult. Second, as with

functional encryption also predicate encryption is typically run by a trusted third party that sets up the keys (public key, master secret key) and serves secret keys $sk_{\mathbf{y}}$ to recipients using a key derivation algorithm. In our fuzzy PSI scenario, there are only two parties, sender and receiver, who cannot resort to a trusted third party.

We address the first challenge by devising a new weak(er) selective security definition for predicate encryption for which we can show that it implies a weak notion of simulation-based security that is sufficiently strong to be useful for our purposes. As this new simulation-based security definition is implied by current selectively secure predicate encryption schemes, we can use these schemes as a simple hybrid in our constructions.

We solve the second challenge by letting the fuzzy PSI Sender S run the trusted third party and set up the system. Then, for each query for a decryption key sk_y from Receiver R, we propose a new two-party key derivation protocol, where the input from S is the master secret key, the input from R is $\mathbf{y}$, and the only information R learns is $sk_{\mathbf{y}}$. Sender S does not learn anything about $\mathbf{y}$. Although such a two-party key derivation protocol can be achieved by reverting to general 2PC techniques, the outcome is often impractical in terms of high communication or computation costs. Consequently, we design a new concretely practical OT-based protocol tailored to a recent inner product predicate encryption scheme, maintaining linear communication complexity in n_R.

Summary: By abstracting these two steps, we arrive at the following informal, simplified description of protocol Π_{FPSI}.

1) Sender S sets up a predicate encryption scheme for inner products, encrypts slightly modified versions of each input vector $\mathbf{x}_i$, and sends resulting ciphertexts $\mathbf{c}_i$ to Receiver R.

2) S and R engage in a two-party distributed key derivation protocol allowing R to obtain a secret key $sk_{\mathbf{y}_j}$ for a slight variation of each of R's input vectors $\mathbf{y}_j$.

3) For each combination of $\mathbf{c}_i$ and $sk_{\mathbf{y}_j}$, R tries to decrypt $\mathbf{c}_i$. A successful decryption reveals that the inner product of $\mathbf{x}_i$ and $\mathbf{y}_j$ satisfies a specific condition, implying that $\mathbf{x}_i$ and $\mathbf{y}_j$ are within a certain Hamming distance. Simultaneously, R recovers $\mathbf{x}_i$ and adds $(\mathbf{x}_i, \mathbf{y}_j)$ to the fuzzy intersection.

The resulting communication complexity is in $O(n_S + n_R)$ for sending all ciphertexts from S to R and obtaining secret keys. Computational complexity is in $O(n_S \cdot n_R)$ as R has to try all possible combinations of ciphertexts and secret keys.

1.2 Preliminaries

We briefly summarize the notation used throughout this paper.

To denote a length-n ordered sequence of elements x_i, we write $(x_1, \ldots, x_n)$. Vectors $\mathbf{x}$ are sequences of elements and written in bold fonts. For vector $\mathbf{x} = (x_1, \ldots, x_n)$ of length n, we write $\mathbf{x}[i]$ to denote the i^{th} element x_i. We use $i \in [n]$ as a shorthand for $i \in \{1, \ldots, n\}$ and $(x_i)_{i\in[n]}$ as a shorthand for sequence $(x_1, \ldots, x_n)$. We make use of predicates $[A \stackrel{?}{=} B]$ that can either evaluate to 1 (true) or 0 (false). If A equals B, then $[A \stackrel{?}{=} B]$ evaluates to 1, otherwise it evaluates to 0.

PARAMETERS: Prime p, vector length ℓ, number n_S of input vectors $\mathbf{x}_i \in \{-1,1\}^\ell$ from Sender S, number n_R of input vectors $\mathbf{y}_j \in \{-1,1\}^\ell$ from Receiver R, set $\mathcal{T} \subset \mathbb{N}$

1. Wait for vectors $(\mathbf{x}_i)_{i\in[n_S]}$ from S.
2. Wait for vectors $(\mathbf{y}_j)_{j\in[n_R]}$ from R.
3. Send $(b_{i,j,\tau}, \hat{\mathbf{x}}_{i,j,\tau})_{i\in[n_S],j\in[n_R],\tau\in\mathcal{T}}$ to R, where

$$b_{i,j,\tau} = [\langle \mathbf{x}_i, \mathbf{y}_j \rangle \stackrel{?}{=} \tau] \qquad \hat{\mathbf{x}}_{i,j,\tau} = \begin{cases} \mathbf{x}_i, \text{ if } b_{i,j,\tau} = 1 \\ \perp, \text{ otherwise.} \end{cases}$$

Figure 2: Ideal restricted inner-product predicate encryption functionality $\mathcal{F}_{\mathsf{PEI}}$

For Fuzzy PSI, the inputs of sender and receiver are sets, and each element in the set is a vector. As we will see later in Section 3, predicate encryption schemes are defined over *attribute* vectors over vector space $\mathbb{Z}_p^\ell$. At the same time, Fuzzy PSI requires binary vectors over $\{0,1\}$ as input, and other functionalities need vectors over $\{-1,1\}$ as input. If clear from the context, we will use terms *attribute vectors* and *vectors* interchangeably in this paper.

Security model: We operate in the standard semi-honest security model, where parties follow the protocol, but are curious to learn from the transcript. This approach is consistent with the vast majority of the Fuzzy PSI literature [13, 16, 20, 21, 29, 32, 35, 57, 63–66], allowing us to focus on the core challenge of achieving linear communication for unstructured data. We briefly discuss a potential path to malicious security in the full version of this paper [7].

2 Protocol Details

This section focuses on our main contribution, a protocol Π_{FPSI} securely realizing ideal fuzzy PSI functionality $\mathcal{F}_{\mathsf{FPSI}}$. To simplify exposition and ease understanding, we assume for now the existence of an ideal functionality $\mathcal{F}_{\mathsf{PEI}}$ as shown in Figure 2. We will use $\mathcal{F}_{\mathsf{PEI}}$ in the construction of protocol Π_{FPSI} as a building block. Later in Section 3, we then describe the actual protocol implementing building block $\mathcal{F}_{\mathsf{PEI}}$.

The main idea behind ideal functionality $\mathcal{F}_{\mathsf{PEI}}$ is that Sender S sends their input vectors $\mathbf{x}_i \in \{-1,1\}^\ell$ to a trusted third party (TTP), and also Receiver R sends their vectors $\mathbf{y}_j \in \{-1,1\}^\ell$ to the TTP. Observe that, for $\mathcal{F}_{\mathsf{PEI}}$, input vectors $\mathbf{x}_i$ and $\mathbf{y}_j$ are over $\{-1,1\}$ and *not* binary vectors. For set $\mathcal{T} \subset \mathbb{N}$, the TTP then sends back to R whether, for all $\mathbf{x}_i$, $\mathbf{y}_j$, and $\tau \in \mathcal{T}$, the inner products are equal to τ. That is, Receiver R learns all predicates $[\langle \mathbf{x}_i, \mathbf{y}_j \rangle \stackrel{?}{=} \tau]$. Moreover, in case $\langle \mathbf{x}_i, \mathbf{y}_j \rangle = \tau$, R also learns $\mathbf{x}_i$.

We call $\mathcal{F}_{\mathsf{PEI}}$ a *restricted inner-product predicate* encryption functionality, as it is close to regular predicate encryption for inner product predicates, but we are restricting to input vectors over $\{-1,1\}$ instead of $\mathbb{Z}_p$ and require inner products

equal to τ. We will clarify details in Section 3.

2.1 Building Fuzzy PSI with $\mathcal{F}_{\mathsf{PEI}}$

The key challenge to overcome when privately computing fuzzy PSI for the Hamming distance is to privately compute the Hamming distance itself. Our approach for privately computing the Hamming distance begins by exploiting a well-known relation between the Hamming distance $\mathsf{HD}(\mathbf{x},\mathbf{y})$ of two vectors $\mathbf{x}$ and $\mathbf{y}$ and their inner product $\langle \mathbf{x},\mathbf{y}\rangle$. However, instantiating this relation in a secure protocol turns out to be non-trivial, as it requires composable security and a distributed key setup.

In general, the Hamming distance of two binary vectors $\mathbf{x},\mathbf{y} \in \{0,1\}^\ell$ can be computed using the inner product with the following trick [42]. For vector $\mathbf{x} \in \{0,1\}^\ell$ (and similarly $\mathbf{y}$) construct vector $\mathbf{x}' \in \{-1,1\}^\ell$ (and similarly $\mathbf{y}'$) by setting $\mathbf{x}'[i] = -1$ if $\mathbf{x}[i] = 0$, and $\mathbf{x}'[i] = 1$ if $\mathbf{x}[i] = 1$. As a consequence, we have $\mathsf{HD}(\mathbf{x},\mathbf{y}) = \frac{\ell - \langle \mathbf{x}',\mathbf{y}'\rangle}{2}$.

To compute $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{=} t]$ for some $t \in \mathbb{N}$, we just have to check whether $[\langle \mathbf{x}',\mathbf{y}'\rangle \stackrel{?}{=} \ell - 2t]$. So, for $\tau = \ell - 2t$, we have $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{=} t] = [\langle \mathbf{x}',\mathbf{y}'\rangle \stackrel{?}{=} \tau]$, which we can compute using $\mathcal{F}_{\mathsf{PEI}}$. To check whether the Hamming distance of vectors $\mathbf{x}$ and $\mathbf{y}$ is less than some threshold t, $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{<} t] = 1$, our idea is to simply compute for $\theta \in (0,\ldots,t-1)$ whether $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{=} \theta] = 1$ by setting $\mathcal{T}$ in $\mathcal{F}_{\mathsf{PEI}}$ appropriately. Specifically, to compute $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{<} t]$, we compute $[\langle \mathbf{x}',\mathbf{y}'\rangle \stackrel{?}{=} \tau]$ for $\tau \in \mathcal{T} = \{\ell - 2t + 2,\ldots,\ell\}$ with $\mathcal{F}_{\mathsf{PEI}}$. As soon as $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{<} t] = 1$, this approach leaks t to the adversary which is more than an ideal functionality computing $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{<} t]$ would leak. However, in the specific context of Fuzzy PSI, this additional leakage is consistent with the target ideal functionality: for the case $\mathsf{HD}(\mathbf{x},\mathbf{y}) < t$, the Fuzzy PSI ideal functionality $\mathcal{F}_{\mathsf{FPSI}}$ of Figure 1 outputs $\mathbf{x}$ anyway in the clear to receiver R which allows R to also compute $\mathsf{HD}(\mathbf{x},\mathbf{y})$. We conclude by presenting an ideal functionality $\mathcal{F}_{\mathsf{HD}}^{<t}$ and a realizing protocol $\Pi_{\mathsf{HD}}^{<t}$ that for two sets of vectors $(\mathbf{x}_i)_{i\in[n_S]}$ and $(\mathbf{y}_j)_{j\in[n_R]}$

1. output $[\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j) \stackrel{?}{<} t]$ to R and
2. output $\mathbf{x}_i$ and $\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j)$ to R if $\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j) < t$.

Figure 3 shows $\mathcal{F}_{\mathsf{HD}}^{<t}$, and Figure 4 shows $\Pi_{\mathsf{HD}}^{<t}$. Protocol $\Pi_{\mathsf{HD}}^{<t}$ follows exactly the intuition we have described above.

Lemma 1 (Proof in Appendix B.1)**.** *Protocol $\Pi_{\mathsf{HD}}^{<t}$ securely realizes $\mathcal{F}_{\mathsf{HD}}^{<t}$ in the $\mathcal{F}_{\mathsf{PEI}}$-hybrid model with parameter $\mathcal{T} = \{\ell - 2t + 2,\ldots,\ell\}$.*

2.2 Fuzzy PSI Protocol Π_{FPSI}

With $\mathcal{F}_{\mathsf{HD}}^{<t}$ at hand, the construction of a fuzzy PSI protocol becomes straightforward. In our fuzzy PSI protocol Π_{FPSI} shown in Figure 5, Receiver R simply outputs each $\mathbf{y}_j$ and corresponding $\mathbf{x}_i = \mathbf{z}_{i,j}$ for which $b_{i,j}$ from $\mathcal{F}_{\mathsf{HD}}^{<t}$ equals 1. So, R outputs the $\mathbf{x}_i$ that are within Hamming distance less than t to $\mathbf{y}_j$, as indicated by $b_{i,j} = 1$.

PARAMETERS: Threshold t, vector length ℓ, number n_S of input vectors $\mathbf{x}_i \in \{0,1\}^\ell$ from S, number n_R of input vectors $\mathbf{y}_j \in \{0,1\}^\ell$ from R

1. Wait for vectors $(\mathbf{x}_i)_{i\in[n_S]}$ from Sender S.
2. Wait for vectors $(\mathbf{y}_j)_{j\in[n_R]}$ from Receiver R.
3. Send $(b_{i,j},\beta_{i,j},\mathbf{z}_{i,j})_{i\in[n_S],j\in[n_R]}$ to R where

$$b_{i,j} = [\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j) \stackrel{?}{<} t] \qquad \beta_{i,j} = \begin{cases} \langle \mathbf{x}_i,\mathbf{y}_j\rangle, & \text{if } b_{i,j} = 1 \\ \perp, & \text{otherwise.} \end{cases}$$

$$\mathbf{z}_{i,j} = \begin{cases} \mathbf{x}_i, & \text{if } b_{i,j} = 1 \\ \perp, & \text{otherwise.} \end{cases}$$

Figure 3: Ideal functionality $\mathcal{F}_{\mathsf{HD}}^{<t}$

INPUT OF S: $(\mathbf{x}_i)_{i\in[n_s]}, \mathbf{x}_i \in \{0,1\}^\ell$
INPUT OF R: $(\mathbf{y}_j)_{j\in[n_R]}, \mathbf{y}_j \in \{0,1\}^\ell$
PARAMETERS: Threshold t, an ideal functionality $\mathcal{F}_{\mathsf{PEI}}$ for length ℓ vectors and $\tau \in \mathcal{T} = \{\ell - 2t + 2,\ldots,\ell\}$
PROTOCOL:

1. For $i \in [n_S]$,
   (a) S creates vector $\mathbf{x}'_i$ by replacing each 0 in $\mathbf{x}_i$ by a -1.
   (b) S sends $\mathbf{x}'_i$ to $\mathcal{F}_{\mathsf{PEI}}$.
2. For $j \in [n_R]$
   (a) R creates vector $\mathbf{y}'_j$ by replacing all 0 elements of $\mathbf{y}_j$ by -1.
   (b) R sends $\mathbf{y}'_j$ to $\mathcal{F}_{\mathsf{PEI}}$.
3. For $i \in [n_S]$, $j \in [n_R]$, and $\tau \in \mathcal{T} = \{\ell - 2t + 2,\ldots,\ell\}$ $\mathcal{F}_{\mathsf{PEI}}$ sends $(u_{i,j,\tau},\mathbf{v}_{i,j,\tau})$ back to R.
4. For $i \in [n_S]$, $j \in [n_R]$,
   - if $\exists (i,j,\tau)$ such that $u_{i,j,\tau} = 1$, R outputs $(b_{i,j} = 1, \beta_{i,j} = \frac{\ell-\tau}{2}, \mathbf{z}_{i,j} = \mathbf{v}_{i,j,\tau})$, where every -1 element of $\mathbf{v}_{i,j,\tau}$ is replaced by a 0.
   - otherwise, if $\nexists u_{i,j,\tau} = 1$, R outputs $(b_{i,j} = 0, \beta_{i,j} = \mathbf{z}_{i,j} = \perp)$.

Figure 4: Protocol $\Pi_{\mathsf{HD}}^{<t}$ in the $\mathcal{F}_{\mathsf{PEI}}$-hybrid model

Theorem 1 (Proof in Appendix B.2)**.** *Protocol Π_{FPSI} securely realizes $\mathcal{F}_{\mathsf{FPSI}}$ in the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid model.*

3 Realizing $\mathcal{F}_{\mathsf{PEI}}$

After presenting protocol Π_{FPSI} for Fuzzy Private Set Intersection, we now turn to the construction of its core component, a protocol for ideal functionality hybrid $\mathcal{F}_{\mathsf{PEI}}$. Our approach is based on predicate encryption techniques for inner product predicates, which, as we will demonstrate, already achieve a functionality closely aligned with $\mathcal{F}_{\mathsf{PEI}}$. However, embedding predicate encryption as a building block within a more complex protocol introduces additional technical challenges. We begin with an introduction to relevant predicate encryption schemes, their challenges, and how to realize $\mathcal{F}_{\mathsf{PEI}}$ with them.

INPUT OF S: Input vectors $(\mathbf{x}_i)_{i\in[n_S]}$, $\mathbf{x}_i \in \{0,1\}^\ell$
INPUT OF R: Input vectors $(\mathbf{y}_j)_{j\in[n_R]}$, $\mathbf{y}_j \in \{0,1\}^\ell$
PARAMETERS: Number n_s of input vectors from S, number n_R of input vectors from R, vector length ℓ, threshold t
PROTOCOL:
1. S sends $(\mathbf{x}_i)_{i\in[n_S]}$ in shuffled order to $\mathcal{F}_{\mathsf{HD}}^{<t}$, R sends $(\mathbf{y}_j)_{j\in[n_R]}$ to $\mathcal{F}_{\mathsf{HD}}^{<t}$.
2. R receives back $(b_{i,j}, \beta_{i,j}, \mathbf{z}_{i,j})_{i\in[n_R],j\in[n_S]}$ from $\mathcal{F}_{\mathsf{HD}}^{<t}$.
3. For each $b_{i,j} = 1$, R outputs $(\mathbf{z}_{i,j}, \mathbf{y}_j)$.

Figure 5: Fuzzy PSI Protocol Π_{FPSI} in the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid model

3.1 Predicate Encryption

Informally, predicate encryption is a sub-class of functional encryption where the decryption of a ciphertext is possible only if a predicate function f over private key and ciphertext evaluates to 1, see [10, 11, 42, 53, 62] for an overview. More specifically, a predicate encryption scheme for function f encrypts plaintext m under *attribute* x to ciphertext c. A receiver holding the private key for an attribute y can decrypt c back to m if and only if function $f_y(x)$ evaluates to 1.

Standard examples for predicate functions f include identity-based encryption [8, 9, 59, 61], where attributes x and y could be identities (such as bit strings). In this case, $f_y(x)$ outputs 1 if and only if $x = y$. In attribute-based encryption [5, 37, 68], attributes x and y come from different attribute spaces. Here, x can be a Boolean formula in n variables, and y is an assignment for the n variables. Predicate function $f_y(x)$ evaluates to 1 if and only if Boolean formula x evaluates to true for assignment y.

As with functional encryption, predicate encryption is typically applied in scenarios where the receiver of ciphertext c has to ask a third trusted party for private keys corresponding to attribute y. For example, in ID-based encryption, the receiver would need to show valid credentials to the TTP to get back the private key that allows decryption of all ciphertexts encrypted under their ID y.

The above examples of predicate encryption are called *payload hiding*. A ciphertext encrypting a payload (plaintext m) under attribute x can only be decrypted by a private key for attribute y, if $f_y(x)$ evaluates to 1. In this paper, we require a simplified variation of payload-hiding predicate encryption where the payload m is the actual attribute x. So, after decryption, the receiver does not only learn that $f_y(x) = 1$, but they also learn x in the clear. We will show later in Appendix A that any predicate encryption scheme trivially realizes simplified predicate encryption by setting plaintext $m = x$.

We now formalize the intuition behind this simplified predicate encryption and then define its security properties.

Definition 1. *For attribute space Σ, let predicate $f : \Sigma \times \Sigma \to \{0,1\}$ be a function mapping attributes x and y from Σ to 1 or 0. Let λ be the security parameter and $\mathcal{C}$ the ciphertext space. A* simplified *predicate encryption scheme* $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KDer}, \mathsf{Enc}, \mathsf{Dec})$ *for predicate f is defined as*

- $(pk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$*: generates a public key pk and a master secret key msk.*
- $sk_y \leftarrow \mathsf{KDer}(msk, y)$*: on input master secret key msk and an attribute $y \in \Sigma$, this algorithm outputs a secret key sk_y.*
- $c \leftarrow \mathsf{Enc}_{pk}(x)$*: using public key pk, this algorithm takes attribute $x \in \Sigma$ to output a ciphertext $c \in \mathcal{C}$.*
- $\{x, \perp\} \leftarrow \mathsf{Dec}_{sk_y}(c)$*: for secret key sk_y and a ciphertext c, this algorithm outputs either $x \in \Sigma$ or $\perp$.*

For correctness, we require that, for all $x, y \in \Sigma$ such that $f_y(x) = 1$,

$$Pr[\mathsf{Dec}_{sk_y}(c) = x : (pk, msk) \leftarrow \mathsf{Setup}(1^\lambda), sk_y \leftarrow \mathsf{KDer}(msk, y), c \leftarrow \mathsf{Enc}_{pk}(x)] = 1.$$

Discussion: In the definition above, we limit expressiveness and present the simplified version of predicate encryption only to suit our application's specific needs and to ease notation. For completeness sake, note that, in the general case of predicate encryption, f could also be defined over two different input spaces. There also exist *predicate-only* predicate encryption schemes that do not encrypt a plaintext, but only output if $f_y(x) = 1$, i.e., $[f_y(x) \stackrel{?}{=} 1]$. Also, we only consider so called *(strongly) attribute-hiding, payload-hiding* predicate encryption where x and m are both hidden in case the receiver uses a secret key y where $f_y(x) = 0$. We stress that predicate-only predicate encryption does not give the same properties as what we target with our simplified predicate encryption, and we discuss differences in the full version of this paper [7]. We point out that several other types of predicate encryption schemes with different security guarantees exist. For a more in-depth introduction, we refer to [10, 42, 53]. We discuss these variations and their use for our main construction in the full version of this paper [7].

Of specific interest in this paper are predicate encryption schemes for the prominent *inner-product* predicate [1, 19, 23, 36, 40, 42, 49–52, 55, 69]. There, attributes $\mathbf{x}$ and $\mathbf{y}$ are length-ℓ vectors from vector space $\Sigma = \mathbb{Z}_p^\ell$ for a prime p with $|p| = \lambda$, and $f_{\mathbf{y}}(\mathbf{x}) = 1$ if and only if they are orthogonal, so their inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ is 0.

As simplified predicate encryption for the inner-product predicate over $\mathbb{Z}_p^\ell$ is at the core of this work, we write predicate encryption as a shorthand from now on if obvious from the context. We will also use vector notation $\mathbf{x}$ for attributes.

3.2 Security of Predicate Encryption

The standard, strong security definition for predicate encryption is *adaptive security*. The idea is that the adversary learns, first, the public key and then gets oracle access to KDer before specifying their challenge attribute(s). For predicate encryption of general predicates as well as for functional encryption of general functions, it has been shown difficult to find

simulation-based security definitions and prove security in the standard model [10, 15, 53]. Similar to adaptive simulation-based security for public key encryption [48], the main challenge for the simulator is to send a ciphertext to the adversary without knowing which decryption keys the adversary will request in the future, so which information the adversary will be able to compute from the underlying plaintext.

Recent works have introduced sophisticated predicate encryption schemes that achieve adaptive simulation security for inner product predicates [1, 23, 36, 69]. While these schemes could theoretically serve as hybrid functionalities in our main Fuzzy PSI protocol and its security proof, the resulting scheme would not necessarily be practical or even implementable. The concrete practicality of these recent schemes remains uncertain due (I) their use of fully-homomorphic encryption as a building block, (II) use of complexity leveraging in their security argument or (III) ciphertext and key sizes being linear in the number of plaintexts. Concrete practicality of recent theoretical advances has yet to be validated through implementations and parameter evaluations, as no concrete implementations or performance assessments currently exist.

Game-Based Security: An alternative line of work has presented predicate encryption schemes that are proven secure for a game-based security definition [19, 40, 42, 49–52, 55]. Some of these schemes offer only selective security (see discussion below), work in impractical groups of composite order or have large key sizes. Yet, there exist other schemes that are not only asymptotically efficient, but also concretely practical with implementations available [47, 55]. Unfortunately, using a predicate encryption primitive secure under a game-based definition as a black-box to prove simulation-based security of a more complex protocol realizing $\mathcal{F}_{\mathsf{PEI}}$ (and ultimately Fuzzy PSI) is involved. There are no composability guarantees implied by this type of security definition, and the security proof would need to include a cumbersome reduction to the predicate encryption scheme.

A way to remedy this problem, and our strategy in this section, is to show that a game-based definition implies a similar simulation-based definition. As a result, any scheme providing the game-based security can be used as a hybrid functionality in the more complex protocol, offering the corresponding simulation-based security. There has been only limited exploration of the relationship between game-based and simulation-based security in predicate encryption so far. O'Neill [53] was able to show for general functional encryption that a special type of security called token-non-adaptive (TNA) security implies a corresponding simulation-based definition. Unfortunately, current concretely practical predicate encryption schemes [19, 40, 42, 49–52, 55] offer selective security, a notion that is different from TNA security, and there is no simulation-based security definition implied by selective security for predicate encryption schemes.

Roadmap for the remainder of this section: Surprisingly, we observe and prove that $\mathcal{F}_{\mathsf{PEI}}$ can be implemented by any predicate encryption primitive that meets a weak(er) notion of simulation-based security (Sim-WSS) that we introduce. We also show that this weaker notion of simulation security is implied by a weaker notion of game-based security (IND-WSS) which, in turn, is already achieved by existing predicate encryption schemes that are so far proven secure only using a game-based, selective security definition. That is, we prove that existing predicate encryption schemes can serve as simulation-secure building blocks to securely realize $\mathcal{F}_{\mathsf{PEI}}$ under their respective hardness assumptions. Finally, we eliminate the need for a TTP during the key derivation of predicate encryption schemes. For the general case, we use 2PC. For the scheme of Park [55], we develop a more efficient solution by a careful modification of its KDer algorithm.

Experiment $\mathsf{Exp}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$

$b \xleftarrow{\$} \{0,1\}$
$((\mathbf{x}_{i,0},\mathbf{x}_{i,1})_{i\in[n]},st) \leftarrow \mathcal{A}_1(1^\lambda)$
$(pk,msk) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathbf{c}_i \leftarrow \mathsf{Enc}_{pk}(\mathbf{x}_{i,b}))_{i\in[n]}$
$b' \leftarrow \mathcal{A}_2^{\mathsf{KDer}(msk,\cdot)}(pk,(\mathbf{c}_i)_{i\in[n]},st)$
If $b=b'$ output 1, else output 0.

Experiment $\mathsf{Exp}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$

$b \xleftarrow{\$} \{0,1\}$
$((\mathbf{x}_{i,0},\mathbf{x}_{i,1})_{i\in[n]},(\mathbf{y}_j)_{j\in[n']},st) \leftarrow \mathcal{A}_1(1^\lambda)$
$(pk,msk) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathcal{K} = (\mathsf{KDer}(msk,\mathbf{y}_j))_{j\in[n']}$
$(\mathbf{c}_i \leftarrow \mathsf{Enc}_{pk}(\mathbf{x}_{i,b}))_{i\in[n]}$
$b' \leftarrow \mathcal{A}_2(pk,\mathcal{K},(\mathbf{c}_i)_{i\in[n]},st)$
If $b=b'$ output 1, else output 0.

Figure 6: (Weak) selective game-based definitions

3.3 Selective Security

We follow the approach suggested by Boneh et al. [10] and O'Neill [53]. We define a weak game-based definition for which we can show that it implies a weak simulation-based security definition for predicate encryption. Our weak simulation-based security definition might not have much utility for general predicate encryption scenarios in other applications and other contexts, but it is sufficiently strong to be useful as a building block in the special case of Fuzzy PSI and securely realizing $\mathcal{F}_{\mathsf{PEI}}$.

We start with a simplified game-based notion for *selective security*, matching our simplified predicate encryption where the plaintext m equals the attribute x under which it is encrypted. To avoid cumbersome notation, we adopt this simplification without loss of generality, as it does not affect the validity of our results. More precisely, the only difference between this simplified selective security below and regular selective security is that the adversary cannot output plaintexts m as part of their challenge. Appendix A shows that any predicate encryption scheme with standard selective security from related work [42, 49, 50, 55] also trivially realizes simplified predicate encryption in the random oracle model.

The main idea of selective security [3, 14, 34] in general is that the adversary has to commit to the attributes they want to be challenged on *before* receiving the public key. Below is the formal simplified selective security game-based definition IND-SS ("IND-Selective Security") for predicate encryption.

Definition 2 (IND-SS)**.** *Let* λ *be the security parameter,* $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KDer}, \mathsf{Enc}, \mathsf{Dec})$ *be a predicate encryption scheme, and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary.*

Consider security experiment $\mathsf{Exp}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *in Figure 6 where* $\mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \Sigma$*. For each of* $\mathcal{A}$*'s inputs* $\mathbf{y} \in \Sigma$ *to an oracle call* $\mathsf{KDer}(msk, \cdot)$*, it must hold that*
1. $(f_{\mathbf{y}}(\mathbf{x}_{i,0}) = f_{\mathbf{y}}(\mathbf{x}_{i,1}))_{i\in[n]}$ *and*
2. for each i: if $f_{\mathbf{y}}(\mathbf{x}_{i,0}) = f_{\mathbf{y}}(\mathbf{x}_{i,1}) = 1$*, then* $\mathbf{x}_{i,0} = \mathbf{x}_{i,1}$.
The probability that experiment $\mathsf{Exp}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *outputs* 1 *is* $Pr[\mathsf{Exp}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1]$.

A predicate encryption scheme PE *is* IND-SS *secure* iff *for all PPT*(λ) *adversaries* $\mathcal{A}$*, the following is negligible in* λ*:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 2 \cdot Pr[\mathsf{Exp}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1] - 1.$$

As standard, Definition 2 requires equality of predicate evaluations and equality of attributes in case an attribute can be decrypted, so that $\mathcal{A}$ cannot trivially derive b.

In the IND-SS security definition above as well as in all following definitions, we specify security for multiple encryptions (Definition 12.5 of Katz and Lindell [41]). The adversary can send n pairs of attribute vectors $(\mathbf{x}_{i,0}, \mathbf{x}_{i,1})_{i\in[n]}$ instead of a single pair of attributes $(\mathbf{x}_0, \mathbf{x}_1)$. As with regular public key encryption, also selectively secure predicate encryption schemes secure for one encryption are secure for multiple encryptions using a standard hybrid argument, see, e.g., Lemma 6 in [34].

3.4 Weak Selective Security

Yet, even for this selective security setting, it is unclear how to derive a simulation-based definition, amenable for composition to prove the security of our fuzzy PSI scheme, and that could be reduced to Definition 2. In the reduction, a simulator Sim would receive an $\mathbf{x}$ from the adversary in the beginning and need to generate an $\mathbf{x}'$ such that $f_{\mathbf{y}_i}(\mathbf{x}) = f_{\mathbf{y}_i}(\mathbf{x}')$ for all $\mathbf{y}_i$ that Sim would not have at this step. Attributes $\mathbf{y}_i$ become available to Sim only later during key derivation. Our insight is that, for the specific case of Fuzzy PSI, the following weaker definition of selective security for predicate encryption is sufficient. In our weaker definition, the adversary specifies challenge attribute $\mathbf{x}$ and all $\mathbf{y}_i$ they will query for during key derivation up front. The weaker selective game-based security implies a simulation-based security that we use in our proof of Fuzzy PSI. In the full version of this paper [7], we further discuss real-world implications and use cases for predicate encryption schemes that meet our weaker security definition.

Note that our weak selective security resembles the one for arbitrary functional encryption by Garg and Srinivasan [30] and the "very selective" security by Agrawal [1].

3.4.1 Game-Based Security

We will now present our weak game-based security definition IND-WSS (IND-"Weak Selective Security"). Both the game-based IND-WSS definition as well as our simulation-based definition Sim-WSS later follow the game- and simulation-based template definitions for adaptive security of O'Neill [53]. As with selective security, the difference to these templates is that the adversary essentially commits to both the challenge attributes $(\mathbf{x}_{i,0}, \mathbf{x}_{i,1})$ and the $\mathbf{y}_j$ before Setup is called, and the adversary can only get keys for the $\mathbf{y}_j$ they have initially committed to. We will obtain two interesting (sub-)results: first, the weaker IND-WSS and Sim-WSS security definitions prove sufficient for our fuzzy PSI construction. Second, there are already concretely practical schemes that satisfy them. Our new security notions also hold independent value and potential for other applications, since they are satisfied by a wider range of predicate encryption schemes

Definition 3 (IND-WSS)**.** *Let* λ *be the security parameter,* $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KDer}, \mathsf{Enc}, \mathsf{Dec})$ *be a simple predicate encryption scheme, and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary. Consider security experiment* $\mathsf{Exp}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *in Figure 6. All* $\mathbf{x}_{i,0}$*,* $\mathbf{x}_{i,1}$*, and* $\mathbf{y}_j$ *output by* $\mathcal{A}_1$ *must be such that*
1. $f_{\mathbf{y}_j}(\mathbf{x}_{i,0}) = f_{\mathbf{y}_j}(\mathbf{x}_{i,1})$.
2. if $f_{\mathbf{y}_j}(\mathbf{x}_{i,0}) = f_{\mathbf{y}_j}(\mathbf{x}_{i,1}) = 1$*, then* $\mathbf{x}_{i,0} = \mathbf{x}_{i,1}$.
The probability that experiment $\mathsf{Exp}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *outputs* 1 *is* $Pr[\mathsf{Exp}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1]$.

Predicate encryption scheme PE *is* IND-WSS *secure* iff *for all PPT*(λ) *adversaries* $\mathcal{A}$*, the following is negligible in* λ*:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 2 \cdot Pr[\mathsf{Exp}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1] - 1.$$

Before presenting the simulation-based security definition implied by IND-WSS, we briefly show that IND-SS security implies IND-WSS security. With IND-WSS being weaker than IND-SS, we can then use any concretely practical selectively secure predicate encryption scheme for inner-products in our implementation and evaluation. It will automatically satisfy our weak simulation-based security definition, too.

Lemma 2 (Proof in Appendix B.3)**.** *Let* PE *be* IND-SS *secure predicate encryption. Then,* PE *is also* IND-WSS *secure.*

3.4.2 Preimage Sampleability

Before completing the transition from game-based to simulation-based security, we need one final ingredient. Predicate f for which our predicate encryption scheme is defined for must be *preimage sampleable* [53]. Preimage sampleability for f means that, given a sequence of $f_{\mathbf{y}_j}(\mathbf{x})$ for unknown $\mathbf{x}$, you can efficiently compute an $\mathbf{x}'$ such that $f_{\mathbf{y}_j}(\mathbf{x}) = f_{\mathbf{y}_j}(\mathbf{x}')$ for all j. As preimage sampleability is not new [53] and for space reasons, we defer its definition and the proof that inner-product predicate is preimage sampleable to Appendix C.

3.4.3 Simulation-based Security

Finally, we present our simulation-based security definition Sim-WSS ("Sim-weak selective security").

$\underline{\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)}$
$((\mathbf{x}_i)_{i\in[n]},(\mathbf{y}_j)_{j\in[n']},\sigma) \leftarrow \mathcal{A}_1(1^\lambda)$
$(msk,pk) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathcal{K} = (\mathsf{KDer}(msk,\mathbf{y}_j))_{j\in[n]}$
$\mathbf{c}_r \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_i))_{i\in[n]}$
$\sigma' \leftarrow \mathcal{A}_2(pk,\mathbf{c}_r,\mathcal{K})$
If $\sigma=\sigma'$ output 1 else 0.

$\underline{\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda)}$
$((\mathbf{x}_i)_{i\in[n]},(\mathbf{y}_j)_{j\in[n']},\sigma) \leftarrow \mathcal{A}_1(1^\lambda)$
$(msk,pk) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathcal{K} = (\mathsf{KDer}(msk,\mathbf{y}_j))_{j\in[n]}$
$\mathbf{c}_s \leftarrow \mathsf{Sim}(pk,(\mathbf{y}_j,f_{\mathbf{y}_j}(\mathbf{x}_i))_{i\in[n],j\in[n']},\mathcal{K})$
$\sigma' \leftarrow \mathcal{A}_2(pk,\mathbf{c}_s,\mathcal{K})$
If $\sigma=\sigma'$ output 1 else 0.

Figure 7: Simulation-based security experiments

Definition 4 (Sim-WSS)**.** *Let* λ *be the security parameter,* $\mathsf{PE}=(\mathsf{Setup},\mathsf{KDer},\mathsf{Enc},\mathsf{Dec})$ *be a predicate encryption scheme, and* $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ *be an adversary. Consider the two security experiments* $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *and* $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda)$ *in Figure 7. Let the probabilities that experiment* $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *and experiment* $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *output* 1 *be*

$$Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)=1] \text{ and } Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda)=1].$$

Predicate encryption scheme PE *is* Sim-WSS *secure* iff *there exists a PPT* (λ) *simulator* Sim *such that for all PPT* (λ) *adversaries* $\mathcal{A}$*, the following is negligible in* λ*:*

$$\mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda) = Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)=1] - Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda)=1].$$

The security intuition behind this definition is that $\mathcal{A}_1$ outputs vectors $(\mathbf{x}_i)_{i\in[n]}$ and $(\mathbf{y}_j)_{i\in[n']}$, but also some value σ. Only after this step is the public/master key setup. In the real experiment, $(\mathbf{x}_i)_{i\in[n]}$ is encrypted into a ciphertext $\mathbf{c}_r$. A scheme is simulation secure if there exists a simulator Sim that can generate a ciphertext $\mathbf{c}_s$ using only the output of the ideal functionality, the public key pk and $\mathcal{K}$ (the set of private keys for $(\mathbf{y}_j)_{i\in[n']}$), such that allowing any adversary $\mathcal{A}_2$ access to the ciphertext $\mathbf{c}_r$, along with the public key pk and $\mathcal{K}$ does not give it a non-negligible advantage, to guess σ, over a run using $\mathbf{c}_s$ the ciphertext output by the Sim. The intuition is that the ciphertext does not reveal anything about the $(\mathbf{x}_i)_{i\in[n]}$ that cannot be simulated from the output of the ideal functionality.

Lemma 3 (Proof in Appendix B.4)**.** *Let* PE *be an* IND-WSS *secure predicate encryption scheme for a preimage sampleable predicate function. Then,* PE *is also* Sim-WSS *secure.*

3.5 Two-Party distributed KDer

So far, we have silently ignored two important issues. First, we have assumed that Receiver R can somehow obtain secret keys $\mathcal{K}$ for each of their input vectors $\mathbf{y}$. In the standard setting of predicate encryption, it is typically a TTP that runs Setup, derives master secret key msk, and then answers KDer queries by clients. However, in our two-party setting where Sender S sets up the encryption, and S and R are mutually untrusted, we need a distributed two-party KDer. Essentially,

PARAMETERS: Prime p, predicate encryption scheme PE set up by Sender S for vectors of length $(\ell+1)$, number n_R of input vectors $\mathbf{y}_j \in \{0,1\}^\ell$ from Receiver R, set $\mathcal{T} \subset \mathbb{N}$

1. Wait for master secret key msk from S.
2. Wait for vectors $(\mathbf{y}_j)_{j\in[n_R]}$ from R.
3. For each $j \in [n_R], \tau \in \mathcal{T}$,
   (a) Create length-$(\ell+1)$ vector $\mathbf{y}_{j,\tau}$ by setting
   $$\mathbf{y}_{j,\tau}[1]=\mathbf{y}_j[1],\ldots,\mathbf{y}_{j,\tau}[\ell]=\mathbf{y}_j[\ell],\mathbf{y}_{j,\tau}[\ell+1]=-\tau.$$
   (b) Send $sk_{\mathbf{y}_{j,\tau}} \leftarrow \mathsf{KDer}(msk,\mathbf{y}_{j,\tau})$ to R.

Figure 8: Ideal functionality $\mathcal{F}_{\mathsf{KDer}}$

S and R engage in a two-party KDer such that S does not learn anything about R's input $\mathbf{y}$, R does not learn anything about msk, but R still obtains secret key $sk_{\mathbf{y}} \leftarrow \mathsf{KDer}(msk,\mathbf{y})$.

The second issue that we have ignored is that standard predicate encryption only tests whether the inner product of vectors equals 0, i.e., $[\langle\mathbf{x},\mathbf{y}\rangle \stackrel{?}{=} 0]$. However, for functionality $\mathcal{F}_{\mathsf{PEI}}$, we need to test whether the inner product equals any $\tau \in \mathbb{N}$, so $[\langle\mathbf{x},\mathbf{y}\rangle \stackrel{?}{=} \tau]$.

We address both issues in this section in a combined way.

Support for arbitrary inner products: The second issue of privately testing for arbitrary inner products can be easily addressed. There exists a well-known transformation [42] that allows to check whether $[\langle\mathbf{x},\mathbf{y}\rangle \stackrel{?}{=} t]$ for $t \in \mathbb{N}$ by just using the regular functionality for predicate $[\langle\mathbf{x},\mathbf{y}\rangle \stackrel{?}{=} 0]$ as a sub-routine. Specifically, to check whether, for two length-ℓ vectors $\mathbf{x}$ and $\mathbf{y}$, their inner product equals t instead of 0, we create two vectors $\mathbf{x}',\mathbf{y}'$ of length $(\ell+1)$. For $i \in \{1,\ldots,\ell\}$, we set $\mathbf{x}'[i]=\mathbf{x}[i]$ and $\mathbf{y}'[i]=\mathbf{y}[i]$. At position $\ell+1$, we set $\mathbf{x}'[\ell+1]=1$ and $\mathbf{y}'[\ell+1]=-t$. Evaluating the inner product predicate on $\mathbf{x}'$ and $\mathbf{y}'$ as input allows deriving if the inner product of $\mathbf{x}$ and $\mathbf{y}$ is t, i.e., $[\langle\mathbf{x}',\mathbf{y}'\rangle \stackrel{?}{=} 0] = [\langle\mathbf{x},\mathbf{y}\rangle \stackrel{?}{=} t]$.

So, to support checking for arbitrary products of length-ℓ vectors, we instantiate a predicate encryption scheme for length-$(\ell+1)$ vectors and run the above transformation.

Secure KDer *Computation:* The transformation of working on length-$(\ell+1)$ vectors leads to the ideal functionality $\mathcal{F}_{\mathsf{KDer}}$ shown in Figure 8. To be able to test whether the inner product of two length-ℓ vectors is τ, R needs to retrieve secret key $sk_{\mathbf{y}'}$ for corresponding length-$(\ell+1)$ vector $\mathbf{y}'$.

There are several ways one can realize such an $\mathcal{F}_{\mathsf{KDer}}$ functionality, and we present two approaches. One is a black-box technique based on 2PC (such as garbled circuits), and one is modifying the actual real-world KDer algorithm of the predicate encryption scheme used. While both techniques are asymptotically efficient with computation and communication complexity polynomial in the security parameter, the second approach is also concretely practical for the scheme we will be using (and others) in our implementation later in Section 4.

INPUT OF S: Master secret key msk of predicate encryption scheme PE for length-$(\ell+1)$ vectors, msk includes G, $G_{j,i}, H_{j,i} \in \mathbb{G}, f_{j,i} \in \mathbb{Z}_p$, for $j \in [4], i \in [\ell+1]$

INPUT OF R: Vectors $(\mathbf{y}_j)_{j\in[n_R]}, \mathbf{y}_j \in \{-1,1\}^\ell$

PARAMETERS: Pairing group $\mathbb{G}$ used in PE, prime p, number n_R of R's input vectors, length ℓ, set $\mathcal{T} \subset \mathbb{N}$

PROTOCOL:

For each $\mathbf{y}_j$, for $\tau \in \mathcal{T}$,

1. let length-$(\ell+1)$ vector
$$\mathbf{y}'[1] = \mathbf{y}_j[1], \ldots, \mathbf{y}'[\ell] = \mathbf{y}_j[\ell], \mathbf{y}'[\ell+1] = -\tau.$$
2. for $i \in [\ell]$,
   (a) for $u \in [4]$, S chooses random $M_{u,i} \xleftarrow{\$} \mathbb{G}$ and computes
   $$K^{-1}_{u,i} = G_{u,i} - H_{u,i} \qquad K^{1}_{u,i} = G_{u,i} + H_{u,i}$$
   $$\kappa^{-1}_{u,i} = M_{u,i} + f_{u,i} K^{-1}_{u,i} \qquad \kappa^{1}_{u,i} = M_{u,i} + f_{u,i} K^{1}_{u,i}.$$
   (b) S and R engage in 1-out-of-2 OT where S is the OT sender with input $(K^{-1}_{u,i} || \kappa^{-1}_{u,i})_{u\in[4]}$ and $(K^{1}_{u,i} || \kappa^{1}_{u,i})_{u\in[4]}$, and R is the OT receiver with input $\mathbf{y}'[i]$ receiving $K^{\mathbf{y}'[i]}$. R extracts the $K^{\mathbf{y}'[i]}_{u,i}$ which are part of secret key $sk_{\mathbf{y}'}$ and the $(\kappa^{\mathbf{y}'[i]}_{u,i})_{u\in[4]}$.
3. For $u \in [4]$, S chooses random $M_{u,\ell+1} \xleftarrow{\$} \mathbb{Z}_p$ and sets
$$K_{u,\ell+1} = G_{u,\ell+1} + \mathbf{y}'[\ell+1] \cdot H_{u,\ell+1} \text{ and}$$
$$\kappa_{u,\ell+1} = M_{u,\ell+1} + f_{u,\ell+1} K_{u,\ell+1}.$$
4. S sends $(K_{u,\ell+1}, \kappa_{u,\ell+1})_{u\in[4]}$ and $\gamma = G - \sum_{i\in[\ell+1],u\in[4]} M_{u,i}$ to R.
5. R computes $K_A = \gamma + \sum_{i\in[\ell],u\in[4]} \kappa^{\mathbf{y}[i]}_{u,i} + \sum_{u\in[4]} \kappa_{u,\ell+1}$ which is part of $sk_{\mathbf{y}'}$.
6. S computes and sends $K_B = G'$ completing $sk_{\mathbf{y}'}$.

Figure 9: Protocol Π_{KDer} in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model

3.5.1 Using 2PC

General two- or multi-party computation techniques such as garbled circuits allow parties to compute any functionality or circuit in a way that both parties only see the output of that computation, but learn nothing else about the other parties' input, see Evans et al. [26] for an overview.

Consequently, for any specific predicate encryption scheme PE, let $\mathcal{C}_{\mathsf{KDer}}(msk, \mathbf{y})$ be a circuit representation implementing PE's key derivation algorithm $\mathsf{KDer}(msk, \mathbf{y})$ with master secret key msk and attribute $\mathbf{y}$ being its input. Let 2PC be a two-party secure circuit computation mechanism such as garbled circuits where $(o_1, o_2) \leftarrow \mathsf{2PC}(\mathcal{C}, i_1, i_2)$ securely evaluates circuit $\mathcal{C}$ on Party 1's input i_1, Party 2's input i_2 and outputs o_1 to Party 1 and o_2 to Party 2. We can just plug circuit $\mathcal{C}_{\mathsf{KDer}}$, msk, and $\mathbf{y}$ into this mechanism, so S and R jointly run $\mathsf{2PC}(\mathcal{C}_{\mathsf{KDer}}, msk, \mathbf{y})$ to obtain $o_1 = \perp$ for S and $o_2 = sk_y$ for R. The 2PC evaluation of $\mathcal{C}_{\mathsf{KDer}}$ is asymptotically efficient and securely realizes $\mathcal{F}_{\mathsf{KDer}}$ by definition.

3.5.2 Concretely practical construction for Park [55]

For the concrete case of the predicate encryption scheme by Park [55] used in our evaluation, there exists a more practical version of two-party KDer without reverting to general 2PC.

Intuition: In Park's scheme, secret keys comprise ℓ elements K_i from some pairing group $\mathbb{G}$, essentially one for each component of attribute vector $\mathbf{y}$. The main idea for a two-party KDer is that the sender prepares two different version of each K_i: $K_{-1,i}$ for the case that $\mathbf{y}[i] = -1$ and $K_{1,i}$ for $\mathbf{y}[i] = 1$. Sender S and receiver R then run ℓ 1-out-of-2 OTs, where in the i^{th} OT, S inputs $(K_{-1,i}, K_{1,i})$, R inputs bit $\mathbf{y}[i]$, and R receives $K_{\mathbf{y}[i],i}$.

Technical details: As the exact details require some understanding of Park [55]'s scheme, we summarize the key derivation (Section 4.1 in [55]). The scheme works for attributes $\mathbf{y} \in \mathbb{Z}_p^\ell$. For an attribute $\mathbf{y}$, the TTP computes secret key $sk_{\mathbf{y}}$ consisting of $4\ell+2$ elements $sk_{\mathbf{y}} = ((K_{1,i}, K_{2,i}, K_{3,i}, K_{4,i})_{i\in[\ell]}, K_A, K_B) \in \mathbb{G}^{4\ell+2}$. Specifically,

- the first 4ℓ elements $K_{j,i}$ are computed as $K_{j,i} = G_{j,i} + \mathbf{y}[i] \cdot H_{j,i}$, where $G_{i,j}, H_{i,j} \in \mathbb{G}$ do not depend on $\mathbf{y}$, but only on master secret key msk and independent randomness.
- $K_A = G + \sum_{i=1}^{\ell} (f_{1,i} K_{1,i} + f_{2,i} K_{2,i} + f_{3,i} K_{3,i} + f_{4,i} K_{4,i})$ where $G \in \mathbb{G}$ and $f_{j,i} \in \mathbb{Z}_p$ come from msk.
- $K_B = G' \in \mathbb{G}$ does not depend on $\mathbf{y}$, but only independent randomness.

We now convert the above KDer into a concretely practical, secure two-party KDer protocol where Sender S inputs master secret key msk, and Receiver R inputs $\mathbf{y}$. Recall that in our case length-ℓ vectors are transformed to length $\ell+1$ vectors $\mathbf{y}'$, where the first ℓ elements are either -1 or 1, and the last element is always set to $-t$. For the first ℓ elements, we let S compute the two possible versions for each $K_{j,i}$ that R could obtain (for either -1 or 1) and mask the $K^{f_{j,i}}_{j,i}$ by a random factor M such that R cannot learn more than K_A. Then, R can fetch the $K_{j,i}$ with OT and compute K_A by peeling off random factors M. For the $(\ell+1)^{\text{th}}$ element of $\mathbf{y}'$, S sends the $K_{j,\ell+1}$ in the clear. Figure 9 presents protocol Π_{KDer} in full detail.

Lemma 4 (Proof in in Appendix B.5)**.** *Protocol Π_{KDer} securely realizes $\mathcal{F}_{\mathsf{KDer}}$ from Figure 8 in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model.*

Discussion: We point out that several other predicate encryption schemes for the inner product predicate use key derivation techniques similar to the one by Park [55], and we conjecture that our efficient two party KDer technique from above also applies in their cases [49–52, 69].

There exists a trivial optimization for Π_{PEI} that we have omitted from Figure 9 to keep our exposition simple: instead of running one separate 1-out-of-2 OTs for each $\tau \in \mathcal{T}$, observe that R's choices do not change for the same $\mathbf{y}'$. Thus, we can run a single OT for the combination of $(K^{-1}_{u,i} || \kappa^{1}_{u,i})$ for all τ of the same $\mathbf{y}'$. Our implementation in Section 4 uses this optimization to reduce the number of OTs by a factor of $|\mathcal{T}|$.

3.6 Π_{PEI} from Sim-WSS and KDer

Finally, we complete the construction of a new protocol to securely realize ideal functionality $\mathcal{F}_{\mathsf{PEI}}$ (Figure 2) with the presentation of protocol Π_{PEI} in Figure 10. It combines

INPUT OF S: $(\mathbf{x}_i)_{i\in[n_S]}, \mathbf{x}_i \in \{-1,1\}^\ell$
INPUT OF R: $(\mathbf{y}_j)_{j\in[n_R]}, \mathbf{y}_j \in \{-1,1\}^\ell$
PARAMETERS: Set $\mathcal{T} \subset \mathbb{N}$, predicate encryption scheme PE for attribute vectors of length $\ell+1$ and matching $\mathcal{F}_{\mathsf{KDer}}$-hybrid for attribute vectors of length ℓ and parameter $\mathcal{T}$, security parameter λ, numbers of input n_S, n_R
PROTOCOL:
1. S invokes $\mathsf{Setup}(1^\lambda)$ and gets (pk, msk). S sends msk to $\mathcal{F}_{\mathsf{KDer}}$.
2. R sends $(\mathbf{y}_j)_{j\in[n_R]}$ to $\mathcal{F}_{\mathsf{KDer}}$ and gets back secret keys $\mathcal{K}$.
3. For each $\mathbf{x}_i$, let length-$(\ell+1)$ vector $\mathbf{x'}_i$ be such that $\mathbf{x'}_i[1] = \mathbf{x}_i[1], \ldots, \mathbf{x'}_i[\ell] = \mathbf{x}_i[\ell], \mathbf{x'}_i[\ell+1] = 1$.
   S computes $(\mathbf{c}_i \leftarrow \mathsf{Enc}_{pk}(\mathbf{x'}_i))_{i\in[n_S]}$ and sends the $\mathbf{c}_i$ to R.
4. For $i \in [n_S]$, $sk_{j,\tau} \in \mathcal{K}$,
   - R computes $\mathbf{z}_{i,j,\tau} = \mathsf{Dec}_{sk_j}(\mathbf{c}_i)$
   - if $\mathbf{z}_{i,j,\tau} = \perp$, R outputs $(0, \perp)$, else R outputs $(1, \mathbf{z}_{i,j,\tau})$.

Figure 10: Protocol Π_{PEI} in the $\mathcal{F}_{\mathsf{KDer}}$-hybrid model

Sim-WSS-secure predicate encryption and distributed KDer in the now obvious way.

Theorem 2 (Proof in Appendix B.6)**.** *Let* PE *be a* Sim-WSS*-secure predicate encryption scheme (*simplified predicate encryption for inner-product predicate over $\mathbb{Z}_p^\ell$*). Then,* Π_{PEI} *securely realizes functionality* $\mathcal{F}_{\mathsf{PEI}}$ *in the* $\mathcal{F}_{\mathsf{KDer}}$*-hybrid model.*

3.7 Discussion

One might argue that basing our construction of protocol Π_{PEI} and thus also Π_{FPSI} on a strong attribute-hiding predicate encryption scheme is unnecessarily restrictive, hinders performance, and weaker *predicate-only* predicate encryption schemes could be sufficient. However, the current state of the art suggests otherwise. For space reasons, we defer the detailed discussion to the full version of this paper [7].

Sim-WSS-secure predicate encryption is of independent interest beyond FPSI. We analyze its broader applicability in the full version [7]. Additionally, Appendix D formalizes an extension to *fuzzy labeled PSI*.

4 Evaluation

We have implemented protocol Π_{FPSI} and evaluated its performance through benchmarks across various combinations of parameters $n_S = n_R$, t, and ℓ. The goal of our evaluation is to investigate the concrete performance of protocol Π_{FPSI} across various parameters and to understand practical trade-offs of achieving linear communication fuzzy PSI without data structure assumptions.

We report on the concrete performance of Π_{FPSI} **without** directly **comparing** it to existing protocols. Related work considering Hamming distance relies on strong assumptions about the input data structure to optimize performance. For instance, Chongchitmate et al. [21] assume a significant gap δt between elements not within threshold t and evaluate for large values of δ (e.g., 8). Gao et al. [29] require the "R. UniqC" assumption, where for each vector of Receiver R there exist at least $t+1$ dimensions such that on each of these dimensions this vector has a unique value different from all other elements of R. Uzun et al. [64] and Chakraborti et al. [16] offer probabilistic security, permitting a false positive rate, while we provide strict, deterministic security guarantees.

In conclusion, our work is free from requiring such assumptions, as our techniques support arbitrary input conditions. So, any direct performance comparison would be both uninformative and inherently unfair. Only to provide context and put our benchmarks into perspective, we also present time and communication cost of the work by Chakraborti et al. [16] (USENIX'23, an improvement of Uzun et al. [64]) alongside our measurements. This line of work is closest to ours regarding the distribution of inputs in their main protocols. However, we stress that [16] is computing only an *approximate* fuzzy PSI when a non-negligible false-positive rate is acceptable.

FHE Strawman: To establish another cost baseline and demonstrate the computational inefficiency and high cost of straightforward FHE-based solutions, we have also implemented and benchmarked an FHE strawman protocol. There, Receiver R encrypts inputs $\mathbf{y}_j$ and sends them to Sender S. For each $\mathbf{x}_i$, S computes whether any $\mathbf{y}_j$ is within the Hamming distance t. Specifically, S homomorphically computes the encryption of $\mathbf{x}_i + r_i \cdot \Pi_{j=0}^{n_R} \Pi_{\tau=0}^{t-1} (\sum_{s=0}^{\ell} \mathbf{x}_i[s] \oplus \mathbf{y}_j[s] - \tau)$ for randomly chosen r_i. These ciphertexts are sent back to R, who decrypts them. If $\mathbf{x}_i$ is in distance t of any $\mathbf{y}_i$, decryption reveals a $\mathbf{x}_i$. We have implemented this approach using Microsoft's SEAL library [60]. To avoid FHE parameter explosion and improve performance, we have also performed optimizations, e.g., we have computed XORs relying only on additions, exploiting the fact that S knows $\mathbf{x}_i[s]$. We have also reduced the number of ciphertext multiplications to $n_S \cdot (\log n_R + \log t)$ by structuring terms as a binary tree.

Our implementation is written in C++ and available for download [6]. At its core, we have re-implemented the predicate encryption scheme by Park [55]. This predicate encryption scheme is selectively secure for a game-based definition, it is designed for the inner-product predicate over $\mathbb{Z}_p^\ell$, so it offers preimage sampleability and is Sim-WSS secure. In contrast to its previous implementation [47], we have ported Park's scheme to the popular MCL library [44] which has allowed easy adoption of the original KDer algorithm to our distributed setting (Figure 9). Cryptographic operations are performed over the Type-3 BN-254 curve using the optimal Ate pairing. To realize the OT functionality $\mathcal{F}_{\mathsf{OT}}$ in Π_{KDer}, we borrow the Ferret-OT implementation from EMP-OT [67]. Ferret realizes random OT, so S encrypts the possible two choices inside each key with the random values output by the random OT and sends the result to R. We use the hash-based KEM-hybrid transformation described in Appendix A to encrypt vectors $\mathbf{x}$ as plaintexts m in the underlying predi-

Table 2: Benchmarks result. Communication: total data between sender and receiver, Time: total runtime, Cost: monetary cost for one run on `t2.xlarge` instances, n: number of input vectors from sender and receiver ($n_S = n_R$), ℓ: vector length, t: threshold for Hamming distance, *: provides weaker security, values for [16] estimated from their paper (see text)

Communication (MByte)

	$n=128$				$n=512$				$n=1024$				$n=4096$			
	$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$
[16]*	3316	13264	3316	13264	53054	212217	53054	212217	212217	848867	212217	848867	$3.4\cdot10^6$	$1.4\cdot10^7$	$3.4\cdot10^6$	$1.4\cdot10^7$
FHE	446	446	446	446	1783	1783	1783	1783	3566	3566	3566	3566	14266	14266	14266	14266
Ours	17	32	33	64	64	127	129	256	128	253	258	511	509	1009	1029	2041

Time (s)

	$n=128$				$n=512$				$n=1024$				$n=4096$			
	$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$
[16]* WAN	837	1611	837	1611	13396	25777	13396	25777	53586	103109	53586	103109	857369	$1.6\cdot10^6$	857369	$1.6\cdot10^6$
FHE WAN	659	1159	768	1276	10238	18320	11834	19960	40791	73212	47206	79676	638014	$1.1\cdot10^6$	746429	$1.3\cdot10^6$
Ours WAN	87	180	176	350	1393	2789	2647	5298	5397	10781	10538	21160	85450	171476	167762	339133
[16]* LAN	577	571	577	571	9235	9132	9235	9132	36940	36526	36940	36526	591036	584418	591036	584418
FHE LAN	624	1124	733	1241	10098	18180	11694	19820	40511	72932	46926	79396	636895	$1.1\cdot10^6$	745310	$1.3\cdot10^6$
Ours LAN	86	177	172	343	1388	2774	2639	5286	5384	10756	10566	21069	85799	168381	166305	333309

Cost (US$)

	$n=128$				$n=512$				$n=1024$				$n=4096$			
	$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$
[16]* WAN	0.33	1.25	0.33	1.25	5.35	19.98	5.35	19.98	21.41	79.92	21.41	79.92	343	1279	343	1279
FHE WAN	0.07	0.1	0.08	0.1	0.68	1.1	0.77	1.19	2.42	4.09	2.75	4.42	34.15	60.51	39.74	66.64
Ours WAN	0.01	0.01	0.01	0.02	0.08	0.15	0.15	0.3	0.29	0.58	0.57	1.14	4.45	8.93	8.74	17.66
[16]* LAN	0.32	1.2	0.32	1.2	5.14	19.12	5.14	19.12	20.56	76.49	20.56	76.49	329	1224	329	1224
FHE LAN	0.07	0.1	0.08	0.1	0.68	1.09	0.76	1.18	2.4	4.07	2.73	4.41	34.09	60.45	39.68	66.58
Ours LAN	0.01	0.01	0.01	0.02	0.08	0.15	0.15	0.29	0.29	0.58	0.57	1.13	4.47	8.77	8.66	17.36

cate encryption scheme. We use AES-based hash Blake2 and AES-based PRG from cryptoTools [58].

Table 2 summarizes our main benchmarks. All benchmarks, Π_{FPSI} and the FHE strawman, were performed on a single Intel Xeon W-1290 CPU. To precisely control the connection speed between sender and receiver, we employ `tc` and `Wondershaper` [38] to emulate two different network environments. First, we set RTT to 70 ms and a bandwidth limit of 100 MBit/s, corresponding to a typical intra-continental WAN setting. Second, we set RTT to 2 ms and a bandwidth limit of 5 GBit/s, corresponding to a typical LAN setup. As the source code of [16] is not available, and their evaluation is only for one single symmetric set size of $n = n_s = n_R = 100$, Table 2 presents only symmetric set sizes and interpolates Chakraborti et al.'s measurements (Section 6 of [16]) for other parameters. They report timings for two `t2.xlarge` Amazon EC2 instances, roughly comparable to our setup in the LAN setting. As our CPU offers 4 more threads (20 instead of 16), we reduce *in their favor* their timings by 25%, assuming perfect parallelization of their scheme. Timings for [16], the FHE strawman, and Π_{FPSI} in Table 2 are dominated by quadratic computation time. In Π_{FPSI} due to the quadratic complexity of Receiver R trying to decrypt all ciphertexts with all secret keys. The quadratic decryption in Π_{FPSI}, as well as in the FHE strawman, are easily parallelizable, so the total runtime can be significantly reduced on multi-core architectures. The timings in Table 2 are total end-to-end times for full protocol runs of Π_{FPSI} and interpolations for [16] (based on their evaluation) and the FHE strawman (based on micro-benchmarks). For Π_{FPSI}, this includes the time to encrypt all sender inputs, perform OT-based distributed key derivation, decryption by R, and all network transmission time. Communication cost includes all n ciphertexts and distributed key derivation (OT and sending the two encrypted choices). For FHE, timings include encryption, decryption, and network transmission time.

Table 2 also presents estimates of the monetary costs for each of the three schemes. We assume sender and receiver executing in cloud environments where CPU time and network communication cost money. To estimate these costs, we adopt pricing again from an Amazon `t2.xlarge` AWS instance [2]. Our cost estimations assume inter-data center communication in the WAN and LAN settings from above.

Discussion: Our construction significantly outperforms both the most closely related work by Chakraborti et al. [16] (USENIX'23) and the FHE-based strawman. We expect the latter to be also limited by poor scalability, as FHE parameters must be adapted to accommodate increasing noise levels.

Our evaluation confirms the theoretical complexity: the quadratic computational cost at the receiver becomes the primary performance driver for large, *symmetric* input sets. However, this cost is highly amenable to parallelization. The $O(n_S n_R)$ decryption checks are independent and can be distributed across multiple cores or machines, leading to near-linear speedups. To demonstrate this, we have also

benchmarked Π_{FPSI} on a `c7a.metal-48xl` instance with 192 threads, see Table 3 (appendix). While this instance is more expensive than a `t2.xlarge`, it reduces the 6 hour runtime for $n = 1024$ to 26 min. More powerful cloud instance offer even more parallelization and are in easy reach for commercial deployment scenarios. More importantly, Π_{FPSI}'s value is most evident in its target scenarios: asymmetric workloads in communication-constrained environments. For example, a client can check a set of 128 elements against a server database of 131k elements in 90 min ($\ell = 16, t = 8$, Table 4 in appendix). This turnaround is practical for offline or batch-processing applications, including biometric matching or threat intelligence queries where assumption-based protocols are inapplicable. Our experiments also validate that the time cost of communication is negligible compared to computation, especially for larger set sizes (e.g., WAN communication time is only 0.2% of total time for $n = 1024, \ell = 32, t = 8$). So, the WAN and LAN settings show little performance differences. We even observed cases where the total time for WAN was slightly smaller than for LAN (e.g., by 0.2% for above setup), most likely due to operating system jitter, running in a cloud environment, and other effects such as thermal throttling that impact CPU speed. By supporting unstructured data, we establish a new baseline for assumption-free FPSI minimizing communication, often the bottleneck in distributed systems.

5 Related Work

Fuzzy PSI: Several schemes aim to achieve linear communication or computation complexity. A common strategy is clustering input data to reduce complexity. However, these techniques make strong assumptions about data distribution.

Uzun et al. [64] propose a Fuzzy Labeled PSI scheme (FLPSI) with Locality Sensitive Hashing and noise removal techniques. They map samples from an Euclidean space to bitstrings amenable to Hamming distance comparisons. The paper uses a combination of subsampling and 2PC computation to derive smaller sets of inputs to be fed into existing exact Labeled PSI schemes. FLPSI is defined only for a closeness functions with probabilistic guarantees distinguishing close (matching) and far (non-matching) elements. It is not defined for elements that are neither close nor far and does not provide formal guarantees for distance thresholds.

Garimella et al. [32] introduce structure-aware PSI (sa-PSI) with communication complexity linear in the sender's set description size, not its cardinality. They exploit the structure of parties' elements and introduce a paradigm for structure-aware PSI using weak boolean function secret-sharing (FSS). This is applied to FPSI, when the sender's set is defined by balls of radius t, and the metric is L_∞ in an ℓ-dimensional space (and additional constraints). For globally-axis-disjoint structures (the projection of the balls onto every axis is disjoint), communication becomes linear in the dimension. Follow-up works introduce a maliciously secure protocol [33] and eliminate λ in the solution's complexity [13].

van Baarsen and Pu [65] combine an oblivious key-value store with additive homomorphic encryption (AHE) to construct a FPSI supporting the L_∞ distance. In follow-up work [66], they replace an AHE-based OPRF with a VOLE-based construction, extending support to general L_p distances. However, in the general case, the cost is exponential in the dimension ℓ. This exponential overhead is eliminated assuming input points are at least $2t$-apart on every coordinate.

Chakraborti et al. [16] propose FPSI schemes for Hamming and integer distances. They represent each input element as a set and formulate the condition for revealing an element to the receiver as the size of the sets' difference exceeding $\ell - t$. To cope with leakage when elements are within the $(t, 2t)$ interval, [16, 35] propose two solutions: (I) homomorphically computing the Hamming distance between elements and filtering elements beyond the threshold, or (II) a sub-sampling technique. In addition to a quadratic communication complexity, this approach has a non-negligible false positive rate.

Son et al. [63] present a FPSI scheme for cosine similarity with computation and communication linear in the dimension of each set element. Thee paper builds on fully homomorphic encryption, optimizing it for approximate sign function evaluation. Unfortunately, the approach requires the sender's elements to be separated by at least twice the threshold t.

Richardson et al. [57] generalize the PSI scheme by Cho et al. [20] to provide FPSI for Euclidean distances L_1, L_2, and L_∞. This scheme uses conditionally overlapping hashing of sender and receiver inputs to execute a PSI over a small set of bins. Although it offers the possibility of trade-offs, the complexity remains exponential in the dimension of the data, limiting its applicability to low-dimentional setups.

Gao et al. [29] present Fuzzy mapping, an abstraction of previous approaches using coarse mapping to cluster sender and receiver elements to reduce the number of PSI executions. The underlying assumption is that, for R's elements, on at least $t + 1$ dimensions, each element has a unique attribute relatively to all other elements. Under this assumption, they design a solution for Hamming distance and L_∞ norm.

Chongchitmate et al. [21] propose FPSI for structured data assuming that elements are either "close" (distance $\leq t$) or sufficiently "far" (distance $\geq 3t$). The scheme achieves near-linear computation and communication complexity for Hamming distance and generalizes to other distances using low distortion embeddings to Hamming distance.

In conclusion, recent schemes demonstrate reductions in complexity when input data aligns with their structural assumptions. Still, the inherent dependence on structure limits general applicability, the limitation addressed by our work.

Functional&predicate encryption are active research areas [1, 10, 19, 23, 36, 40, 42, 49–53, 55, 69]. We build FPSI from IND-WSS predicate encryption and instantiate with Park's scheme, but support any selectively-secure scheme.

Ethical Considerations

This work presents a protocol for efficient Fuzzy Private Set Intersection (FPSI). While our research methodology was primarily theoretical and computational, involving no human subjects or personal data during benchmarks, the application of this technology might have ethical implications for privacy and surveillance. Below, we carefully analyze the stakeholders involved, the potential impacts of deploying this technology, and our rationale for publishing this work.

Stakeholder Identification We identified the following stakeholders who could be impacted by the availability of this protocol.

Data Subjects The individuals whose personal information (biometrics, location history, genetic data) is contained within the datasets being compared. This group includes vulnerable populations whose data might be sensitive or stigmatized.

Data Custodians Organizations or individuals holding the datasets (service providers, government agencies, individual users) who wish to compute intersections without revealing their full databases.

Adversarial Actors Entities seeking to identify specific targets within large, leaked or public datasets.

The Broader Public Society at large, which benefits from privacy-preserving technologies, but is also harmed by the proliferation of efficient surveillance tools.

Impact Analysis and Dual Use The publication of this protocol facilitates data matching, where a party with a (small) set of queries can efficiently check against a large database. This capability could have dual-use impacts:

Positive Impact: Privacy & Security The protocol benefits Data Subjects and Custodians by enabling necessary security checks without exposing raw data. For example, a user can check if their biometric data appears in a database without revealing their identity to the server. Without this protocol, such checks require trusting the server with cleartext data. Thus, the proposed techniques make legitimate data sharing with strong privacy guarantees more efficient, preventing unrestricted mass-sharing of private data.

Negative Impact: Surveillance The linear communication complexity we achieve might lower one barrier to mass surveillance. While FPSI is unlikely to impact nation-state adversaries surveilling citizens' raw private data, it might provide increased efficiency to actors operating in a legal context that prevents them from mass surveillance. They might claim to retrieve information about specific targets as authorized by laws while not learning anything about other users. Previously, the communication cost of such assumption-free FPSI might have made this prohibitive in messy, unstructured data.

Mitigations To mitigate the risks to data subjects, first our protocol design focuses on strict data minimization and provable security. The protocol is formally proven to reveal only the intersection and the Hamming distance of intersecting elements. No auxiliary information about non-matching elements is revealed, preventing attacks where an adversary tries to learn about the database structure or infer attributes of non-targets. Second, our FPSI technique allows the data provider to restrict how many entries can be retrieved . This limits the information gain to exactly what is necessary for the intersection task for a rate limited number of entries.

Conclusion We have carefully debated whether the risk of enabling efficient surveillance outweighs the benefit of providing a privacy-preserving tool. We concluded that publishing is the ethical choice, mainly for two reasons:

The Alternative is Worse In the absence of efficient PSI, organizations currently solve the matching problem by sharing full datasets in clear or using weak anonymization. This status quo causes massive, definite privacy harms to data subjects every day. Our protocol offers a secure alternative that stops this leakage.

Assumption-Free Security When data distribution assumptions of existing works fail in real-world scenarios, those protocols can leak information or fail silently. By providing an assumption-free baseline, we ensure that privacy guarantees hold even for the messy, unpredictable data distributions found in the real world.

Ultimately, we believe that democratizing access to efficient, provably secure matching tools empowers defenders and privacy advocates more than it aids adversaries, who often already possess the resources to perform less efficient, but equally invasive matching.

Open Science

We release the artifacts needed to build and reproduce our results via Zenodo [6]. They include: (1) our C++ implementation and build files, (2) scripts to reproduce experiments and collect logs, (3) the SEAL-integrated micro-benchmark for FHE timings, (4) LAN/WAN network emulation scripts, and (5) third-party dependencies with small patches needed to compile. The top-level README lists the directory structure and maps artifacts to the corresponding results.

References

[1] S. Agrawal. Stronger Security for Reusable Garbled Circuits, General Definitions and Attacks. In *CRYPTO*, 2017.

[2] Amazon. Amazon EC2 On-Demand Pricing, 2022. https://aws.amazon.com/ec2/pricing/on-demand/.

[3] P. Ananth, Z. Brakerski, G. Segev, and V. Vaikuntanathan. From Selective to Adaptive Security in Functional Encryption. In *CRYPTO*, 2015.

[4] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *CCS*, 1993.

[5] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-Policy Attribute-Based Encryption. In *Symposium on Security and Privacy*, 2007.

[6] E.-O. Blass and G. Noubir. Source code, 2025. https://zenodo.org/records/17936990.

[7] Erik-Oliver Blass and Guevara Noubir. Assumption-free fuzzy PSI via predicate encryption. Cryptology ePrint Archive, Paper 2025/217, 2025. URL https://eprint.iacr.org/2025/217.

[8] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, 2004.

[9] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3), 2003.

[10] D. Boneh, A. Sahai, and B. Waters. Functional Encryption: Definitions and Challenges. In *TCC*, 2011.

[11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11), 2012.

[12] P. Buddhavarapu, A. Knox, P. Mohassel, S. Sengupta, E. Taubeneck, and V. Vlaskin. Private matching for compute. Cryptology ePrint Archive, Paper 2020/599, 2020. URL https://eprint.iacr.org/2020/599.

[13] D. Bui, G. Garimella, P. Miao, and P. Van Long Pham. New Framework for Structure-Aware PSI From Distributed Function Secret Sharing. Cryptology ePrint Archive, Paper 2025/907, 2025. URL https://eprint.iacr.org/2025/907.

[14] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *EUROCRYPT*, 2003.

[15] A. De Caro, V. Iovino, A. Jain, A. O'Neill, O. Paneth, and G. Persiano. On the Achievability of Simulation-Based Security for Functional Encryption. In *CRYPTO*, 2013.

[16] A. Chakraborti, G. Fanti, and M. K. Reiter. Distance-Aware private set intersection. In *USENIX*, 2023.

[17] M. Chase and P. Miao. Private set intersection in the internet setting from lightweight oblivious PRF. In *(CRYPTO)*, 2020.

[18] H. Chen, Z. Huang, K. Laine, and P. Rindal. Labeled PSI from Fully Homomorphic Encryption with Malicious Security. In *CCS*, 2018.

[19] J. Chen, J. Gong, and H. Wee. Improved Inner-Product Encryption with Adaptive Security and Full Attribute-Hiding. In *ASIACRYPT*, 2018.

[20] C. Cho, D. Dachman-Soled, and S. Jarecki. Efficient concurrent covert computation of string equality and set intersection. In *CT-RSA*, 2016.

[21] W. Chongchitmate, S. Lu, and R. Ostrovsky. Approximate PSI with near-linear communication. Cryptology ePrint Archive, Paper 2024/682, 2024. URL https://eprint.iacr.org/2024/682.

[22] R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM J. Comput.*, 33(1), 2003.

[23] P. Datta, T. Okamoto, and K. Takashima. Adaptively Simulation-Secure Attribute-Hiding Predicate Encryption. In *ASIACRYPT*, 2018.

[24] S. Dirksen and S. Mendelson. Fast binary embeddings with gaussian circulant matrices: improved bounds. *Discrete & Computational Geometry*, 60(3), 2018.

[25] S. Dirksen and S. Mendelson. Non-gaussian hyperplane tessellations and robust one-bit compressed sensing. *Journal of the European Mathematical Society*, 23 (9), 2021.

[26] D. Evans, V. Kolesnikov, and M. Rosulek. *A Pragmatic Introduction to Secure Multi-Party Computation*. Now Publishers Inc, 2018.

[27] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, 2004.

[28] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *J. Cryptol.*, 26(1), 2013.

[29] Y. Gao, L. Qi, X. Liu, Y. Luo, and L. Wang. Efficient fuzzy private set intersection from fuzzy mapping. Cryptology ePrint Archive, Paper 2024/1462, 2024. URL https://eprint.iacr.org/2024/1462.

[30] S. Garg and A. Srinivasan. Single-Key to Multi-Key Functional Encryption with Polynomial Loss. In *TCC*, 2016.

[31] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai. Oblivious key-value stores and amplification for private set intersection. In *CRYPTO*, 2021.

[32] G. Garimella, M. Rosulek, and J. Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *CRYPTO*, 2022.

[33] G. Garimella, M. Rosulek, and J. Singh. Malicious secure, structure-aware private set intersection. In *CRYPTO*, 2023.

[34] Romain Gay. *Public-Key Encryption, Revisited: Tight Security and Richer Functionalities*. PhD thesis, 2019. https://www.di.ens.fr/~rgay/thesis.pdf.

[35] S. Ghosh and M. Simkin. The communication complexity of threshold private set intersection. In *CRYPTO*, 2019.

[36] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate Encryption for Circuits from LWE. In *CRYPTO*, 2015.

[37] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*, 2006.

[38] B. Hubert, J. Geul, and S. Sehier. Wondershaper, a command-line utility for limiting an adapter's bandwidth, 2021. https://github.com/magnific0/wondershaper.

[39] M. Ion, B. Kreuter, A. E. Nergiz, S. Patel, S. Saxena, K. Seth, M. Raykova, D. Shanahan, and M. Yung. On deploying secure computing: private intersection-sum-with-cardinality. In *EuroS&P*, 2020.

[40] S. Katsumata, R. Nishimaki, S. Yamada, and T. Yamakawa. Adaptively Secure Inner Product Encryption from LWE. In *ASIACRYPT*, 2020.

[41] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Third Edition*. Chapman & Hall, 2020.

[42] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, 2008.

[43] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu. Efficient batched oblivious PRF with applications to private set intersection. In *CCS*, 2016.

[44] Shigeo M. MCL – A portable and fast pairing-based cryptography library, 2025. https://github.com/herumi/mcl.

[45] C. Meadows. A More Efficient Cryptographic Matchmaking Protocol for Use in the Absence of a Continuously Available Third Party. In *Symposium on Security and Privacy*, 1986.

[46] Microsoft. Password Monitor: Safeguarding passwords in Microsoft Edge, 2021. https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/.

[47] MIRACL. Multiprecision Integer and Rational Arithmetic Cryptographic Library, 2023. https://github.com/miracl/MIRACL/blob/master/source/curve/pairing/ipe.cpp.

[48] J. Buus Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case. In *CRYPTO*, 2002.

[49] T. Okamoto and K. Takashima. Hierarchical Predicate Encryption for Inner-Products. In *ASIACRYPT*, 2009.

[50] T. Okamoto and K. Takashima. Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In *ASIACRYPT*, 2012.

[51] T. Okamoto and K. Takashima. Efficient (Hierarchical) Inner-Product Encryption Tightly Reduced from the Decisional Linear Assumption. *Trans. Fundam. Electron. Commun. Comput. Sci.*, 2013.

[52] T. Okamoto and K. Takashima. Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. *Trans. Fundam. Electron. Commun. Comput. Sci.*, 99-A(1), 2016.

[53] A. O'Neill. Definitional Issues in Functional Encryption. Cryptology ePrint Archive, Paper 2010/556, 2010. URL https://eprint.iacr.org/2010/556.

[54] R. Ostrovsky and Y. Rabani. Low distortion embeddings for edit distance. *Journal of the ACM*, 54(5), 2007.

[55] J. Hwan Park. Inner-product encryption under standard assumptions. In *Designs, Codes and Cryptography*, 2011.

[56] S. Raghuraman and P. Rindal. Blazing Fast PSI from Improved OKVS and Subfield VOLE. In *CCS*, 2022.

[57] D. Richardson, M. Rosulek, and J. Xu. Fuzzy PSI via oblivious protocol routing. Cryptology ePrint Archive, Paper 2024/1642, 2024. URL https://eprint.iacr.org/2024/1642.

[58] P. Rindal. cryptoTools library, 2025. https://github.com/ladnir/cryptoTools.

[59] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT*, 2005.

[60] SEAL. Microsoft SEAL (release 4.1). https://github.com/Microsoft/SEAL, January 2023. Microsoft Research, Redmond, WA.

[61] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, 1984.

[62] E. Shen, E. Shi, and B. Waters. Predicate Privacy in Encryption Systems. In *TCC*, 2009.

[63] H. Son, S. Paik, Y. Kim, S. Kim, H. Chung, and J. H. Seo. Doubly efficient fuzzy private set intersection for high-dimensional data with cosine similarity. Cryptology ePrint Archive, Paper 2025/054, 2025. URL https://eprint.iacr.org/2025/054.

[64] E. Uzun, S. P. Chung, V. Kolesnikov, A. Boldyreva, and W. Lee. Fuzzy labeled private set intersection with applications to private Real-Time biometric search. In *USENIX*, 2021.

[65] A. van Baarsen and S. Pu. Fuzzy private set intersection with large hyperballs. In *EUROCRYPT*, 2024.

[66] A. van Baarsen and S. Pu. Fuzzy Private Set Intersection from VOLE. Cryptology ePrint Archive, Paper 2025/911, 2025. URL https://eprint.iacr.org/2025/911.

[67] X. Wang, A. J. Malozemoff, and J. Katz. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, 2016.

[68] B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC*, 2011.

[69] H. Wee. Attribute-Hiding Predicate Encryption in Bilinear Groups, Revisited. In *TCC*, 2017.

A Simplified Predicate Encryption

In our exposition in Section 3.1, we have defined a simplified predicate encryption scheme that directly encrypts a vector $\mathbf{x} \in \Sigma$ to ciphertext $\mathbf{c}$, i.e., $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(\mathbf{x})$. Decryption under secret key $sk_{\mathbf{y}}$ in our simplified definition directly yields $\mathbf{x}$ if and only if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. Yet, standard predicate encryption schemes allow a more powerful setup where a plaintext m from plaintext space $\mathcal{M}$ is encrypted under $\mathbf{x}$, i.e., $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(m, \mathbf{x})$. Decryption yields m if and only if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. If decryption fails, nothing is revealed about m. Moreover, in any case, nothing about $\mathbf{x}$ is revealed besides whether $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. The security definition for both simplified and standard predicate encryption is selective security where the adversary has to output up front the vectors $(\mathbf{x}_{i,0}, \mathbf{x}_{i,1})$ they want to be challenged upon.

We now show that any standard predicate encryption scheme can be transformed into a simplified predicate encryption scheme. While there are various ways how to perform such a transform, we apply the typical approach of *hybrid enryption*. There, the predicate encryption scheme is used as a Key Encapsulation Mechanism (KEM) to encrypt a symmetric key which is then used with symmetric key encryption to encrypt whatever input should be encrypted, see [4, 22, 28] for an overview.

First, let $\mathcal{PE} = (\mathcal{SETUP}, \mathcal{KDER}, \mathcal{ENC}, \mathcal{DEC})$ be a standard predicate encryption scheme for predicate f. We construct simplified predicate encryption scheme $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KDer}, \mathsf{Enc}, \mathsf{Dec})$ for predicate f in the following way.

For PE, we set Setup to be exactly like $\mathcal{SETUP}$, and KDer to be exactly like $\mathcal{KDER}$. We only change encryption and decryption in the following straightforward way.

1. $\mathsf{Enc}_{pk}(\mathbf{x})$: To encrypt $\mathbf{x}$ in the simplified encryption scheme, choose a random $m \xleftarrow{\$} \mathcal{M}$ and use a cryptographic hash function H (modeled as a random oracle) to hash it to a key $k = H(m)$ for a semantically secure encryption (E, D). Then encrypt m under $\mathbf{x}$ using $\mathcal{ENC}$, i.e., $\mathbf{c}_1 \leftarrow \mathcal{ENC}_{pk}(m, \mathbf{x})$. Use the semantically secure encryption to encrypt the bit-representation of $\mathbf{x}$ and key k to ciphertext $c_2 \leftarrow E_k(\mathbf{x})$. Send $(\mathbf{c}_1, c_2)$ to the other party.

2. $\mathsf{Dec}_{sk_{\mathbf{y}}}(\mathbf{c}_1, c_2)$: To decrypt $(\mathbf{c}_1, c_2)$ with $sk_{\mathbf{y}}$, run $\mathcal{DEC}_{sk_{\mathbf{y}}}(\mathbf{c}_1)$. If decryption is successful, not returning $\perp$ but returning m', compute $k' = H(m')$ and decrypt c_2 to $\mathbf{x}'$ using the semantically secure encryption with key k', i.e., $\mathbf{x}' = D_{k'}(c_2)$.

Note that one can also use a PRG G instead of semantically secure encryption (E, D) with k serving as its seed to produce a one-time pad. The security of this KEM-style hybrid encryption scheme PE in the random oracle model follows directly from the security of $\mathcal{PE}$ and the underlying encryption scheme (E, D) (or PRG G), analogous to the argument by Bellare and Rogaway [4].

B Security Proofs

B.1 Lemma 1

Proof. Regarding correctness, recall, first, our conversion between vectors over $\{0,1\}$ and $\{-1,1\}$. More importantly, observe that the way we construct vectors $\mathbf{x'}_i$ and $\mathbf{y'}_j$ leads to

$$[\langle \mathbf{x'}_i, \mathbf{y'}_j \rangle \stackrel{?}{=} \tau] = [\mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j) \stackrel{?}{=} \frac{\ell - \tau}{2}].$$

So, for each $\theta \in \{0, \ldots, t-1\}$, R learns $\mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j) \stackrel{?}{=} \theta$ which allows them to correctly compute and output both $b_{i,j}$, $\beta_{i,j}$, and $\mathbf{z}_{i,j}$ in the last step of Protocol $\Pi^{\leq t}_{\mathsf{HD}}$.

For security, we construct simulators Sim_S and Sim_R for the views of S and R.

$\mathsf{Sim}_S((\mathbf{x}_i)_{i \in [n_S]})$: This simulator is trivial, as S does not receive any message or output. It simply runs the simulator for the sender in the $\mathcal{F}_{\mathsf{PEI}}$-hybrid using arbitrary input to create the view for S.

$\mathsf{Sim}_R((\mathbf{y}_j)_{j \in [n_R]}, (b_{i,j}, \beta_{i,j}, \mathbf{z}_{i,j})_{i \in [n_S], j \in [n_R]})$: Again, the only messages that Sim_R has to generate for R are the responses from $\mathcal{F}_{\mathsf{PEI}}$. For this, Sim_R calls the receiver's simulator of the $\mathcal{F}_{\mathsf{PEI}}$-hybrid. As input to this simulator, Sim_R uses the

Table 3: Benchmark results on AWS `c7a.metal-48xl` instance (AMD Epyc 9R14 CPU, 192 threads), comparison to Intel Xeon W-1290 CPU (20 threads), monetary cost for one run on `c7a.metal-48xl` instance.

Time (s)

	$n=128$				$n=512$				$n=1024$				$n=4096$			
	$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$		$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$	$t=8$	$t=16$
Xeon WAN	87	180	176	350	1393	2789	2647	5298	5397	10781	10538	21160	85450	171476	167762	339133
Xeon LAN	86	177	172	343	1388	2774	2639	5286	5384	10756	10566	21069	85799	168381	166305	333309
c7a.48xl WAN	13	23	23	43	110	215	210	417	409	815	794	1587	5700	11344	11079	22088
c7a.48xl LAN	9	18	17	38	103	205	198	400	399	799	771	1542	5708	11204	11280	21928

Cost (US$)

c7a.48xl WAN	0.04	0.07	0.07	0.13	0.33	0.64	0.63	1.24	1.21	2.41	2.35	4.7	16.75	33.33	32.55	64.9
c7a.48xl LAN	0.03	0.06	0.05	0.12	0.31	0.61	0.59	1.19	1.18	2.36	2.28	4.56	16.77	32.92	33.14	64.43

Table 4: Benchmark results and monetary cost for Π_{FPSI} with **asymmetric** set size $n_S = 2^{17} = 131072$, $n_R = 2^7 = 128$ on AWS `c7a.metal-48xl` instance.

Communication (MByte)

$\ell=16$		$\ell=32$	
$t=8$	$t=16$	$t=8$	$t=16$
285	301	557	589

Time (s)

	$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$
WAN	5654	11490	11107	22074
LAN	5678	11412	10940	22628

Cost (US$)

	$\ell=16$		$\ell=32$	
	$t=8$	$t=16$	$t=8$	$t=16$
WAN	16.59	33.69	32.59	64.73
LAN	16.66	33.46	32.1	66.36

$(\mathbf{y'}_j)_{j\in[n_R]}$. For its output part, the $\mathcal{F}_{\mathsf{PEI}}$ simulator requires $n_S n_R \cdot |\mathcal{T}|$ pairs $(b_{i,j,\tau}, \hat{\mathbf{x}}_{i,j,\tau})$. For each $b_{i,j}=1$ in its own input, Sim_R sets the output pair for the $\mathcal{F}_{\mathsf{PEI}}$ simulator to $(b_{i,j,\frac{\ell-\beta_{i,j}}{2}} = 1, \hat{\mathbf{x}}_{i,j,\frac{\ell-\beta_{i,j}}{2}} = \mathbf{z}_{i,j})$ and all other pairs to $(0,\perp)$. □

B.2 Theorem 1

Proof. Correctness of $\mathcal{F}_{\mathsf{FPSI}}$ follows immediately from the correctness of the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid: R outputs the set of $(\mathbf{x}_i, \mathbf{y}_j)$ that have Hamming distance less than t which is the definition of Fuzzy PSI output.

For security, we construct simulators Sim_S for S and Sim_R for R. Note that S shuffles their input using a random permutation π before sending it to the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid. This is a standard trick, so that R does not learn the real indices of S's input in the intersection. Not to overload notation in the following, we will just write $\mathbf{x}_i$ to denote the i^{th} input of S to Π_{FPSI} even though it is actually the $\pi(i)^{\text{th}}$ input.

$\mathsf{Sim}_S((\mathbf{x}_i)_{i\in[n_S]})$: Sender S does not receive any message or produce any output, so Sim_S just runs the sender's simulator of the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid with arbitrary input to generate S' view.

$\mathsf{Sim}_R(\mathsf{In}_R = (\mathbf{y}_j)_{j\in[n_R]}, \mathsf{Out}_R = \{(\mathbf{x}_i,\mathbf{y}_j) | \mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j) < t\})$: To generate the view of Receiver R, Sim_R runs the receiver's simulator of the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid with the following input and output. The input for the $\mathcal{F}_{\mathsf{HD}}^{<t}$ simulator is simply $\mathsf{In}_R = (\mathbf{y}_j)_{j\in[n_R]}$. For the output $(b_{i,j}, \beta_{i,j}, \mathbf{z}_{i,j})$ of the $\mathcal{F}_{\mathsf{HD}}^{<t}$ simulator, Sim_R sets:

- for each $(\mathbf{x}_i,\mathbf{y}_j) \in \mathsf{Out}_R$, $b_{i,j}=1$, $\beta_{i,j} = \langle \mathbf{x}_i, \mathbf{y}_j \rangle$, $\mathbf{z}_{i,j} = \mathbf{x}_i$.
- for all $i \in [n_S]$ and $j \in [n_R]$ such that $(\mathbf{x}_i,\mathbf{y}_j) \notin \mathsf{Out}_R$, $b_{i,j}=0$, $\beta_{i,j} = \perp$, $\mathbf{z}_{i,j} = \perp$.

□

B.3 Lemma 2

Proof. Assume PE is not IND-WSS secure, so there exists adversary $\mathcal{A}^* = (\mathcal{A}_1^*, \mathcal{A}_2^*)$ in the IND-WSS game such that $\mathsf{Adv}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}^*}(\lambda)$ is non-negligible in λ. We construct adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ for the IND-SS game in Figure 6 that uses this adversary $\mathcal{A}^*$ as a sub-routine. We show that $\mathsf{Adv}^{\mathsf{IND\text{-}SS}}_{\mathsf{PE},\mathcal{B}}(\lambda) = \mathsf{Adv}^{\mathsf{IND\text{-}WSS}}_{\mathsf{PE},\mathcal{A}^*}(\lambda)$.

$\mathcal{B}_1$ runs $\mathcal{A}_1^*$ to get the $\mathbf{x}_{i,0}, \mathbf{x}_{i,1}$ and the $(\mathbf{y}_j)$. $\mathcal{B}_1$ forwards the $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$ to the IND-SS challenger. After receiving public key pk and ciphertexts $\mathbf{c}_i$ back, $\mathcal{B}_2$ asks key derivation oracle KDer for the private keys corresponding to attributes $(\mathbf{y}_j)_{j\in[n']}$. Let the sequence of these private keys returned by the oracle be $\mathcal{K}$. Finally, $\mathcal{B}_2$ calls $\mathcal{A}_2^*$ with pk, $\mathcal{K}$, and the $\mathbf{c}_i$ as input and outputs whatever $\mathcal{A}_2^*$ outputs.

Our reduction is tight, as $\mathcal{B}$ has the same runtime and success probability as $\mathcal{A}^*$. □

B.4 Lemma 3

Proof. Assume PE is not Sim-WSS secure. Consequently, for any simulator Sim, there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ from the Real and Ideal experiments of Definition 4 such that, the advantage $\mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda)$ is not negligible (intuitively

distinguishing the real ciphertext $\mathbf{c}_r$ from the simulator ciphertext $\mathbf{c}_s$). We will use the preimage samplability property to build a specific simulator Sim^*. Given our assumption that PE is not Sim-WSS secure, it means that there exists a corresponding adversary $\mathcal{A}^* = (\mathcal{A}_1^*, \mathcal{A}_2^*)$ from the Real and Ideal experiments of Definition 4 such that, $\mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A}^*,\mathsf{Sim}^*}(\lambda)$ is not negligible. We will use Sim^* and $\mathcal{A}^*$ to construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ for the IND-WSS experiment of Definition 3.

Constructing $\mathcal{B}$: Adversary $\mathcal{B}_1$ starts by running $\mathcal{A}_1^*$. It obtains $((\mathbf{x}_{i,0})_{i\in[n]}, (\mathbf{y}_j)_{j\in[n']}, \sigma)$. As predicate functions $f_{\mathbf{y}_j}$ are preimage sampleable, $\mathcal{B}_1$ uses the $\mathbf{y}_j$ to compute an $(\mathbf{x}_{i,1})_{i\in[n]}$ such that, for all $\mathbf{y}_{j\in[n']}$, predicates $f_{\mathbf{y}_j}$ are the same for $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,1}$, so $f_{\mathbf{y}_j}(\mathbf{x}_{i,0}) = f_{\mathbf{y}_j}(\mathbf{x}_{i,1})$.

Observe that, with non-negligible probability, there exists an $i \in [n]$ such that $\mathbf{x}_{i,1} \neq \mathbf{x}_{i,0}$ and for all $j \in [n']$: $f_{y_j}(\mathbf{x}_{i,0}) = f_{y_j}(\mathbf{x}_{i,1}) = 0$. Otherwise, PE would already be Sim-WSS secure, because the $f_{\mathbf{y}_i}(\mathbf{x}_{i,0})$ would automatically reveal $\mathbf{x}_{i,0}$ to the adversary by preimage sampling. Specifically, if for the computed $(\mathbf{x}_{i,1})_{i\in[n]}$ it would hold that $(\mathbf{x}_{i,0} = \mathbf{x}_{i,1})_{i\in[n]}$ or $f_{\mathbf{y}_j}(\mathbf{x}_{i,0}) = 1$ (which would reveal $\mathbf{x}_{i,0}$), with probability $1 - negl(\lambda)$, then it is possible to create the following simulator Sim' for Definition 4. Simulator Sim' computes inputs $(\mathbf{x}_{i,0})_{i\in[n]}$ using preimage sampling and encrypts $\mathbf{c}_s \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_{i,0}))_{i\in[n]}$. No adversary $\mathcal{A}_2$'s output can be distinguished using input $(pk, \mathbf{c}_r = (\mathsf{Enc}_{pk}(\mathbf{x}_{i,0}))_{i\in[n]}, \mathcal{K})$ or $(pk, \mathbf{c}_s = (\mathsf{Enc}_{pk}(\mathbf{x}_{i,0}))_{i\in[n]}, \mathcal{K})$. This would contradict our assumption that PE is not Sim-WSS secure.

Constructing Sim^*: Consequently, consider $(\mathbf{x}_{i,1} \neq \mathbf{x}_{i,0})_{i\in[n]}$ in the following. For every i such that there exists $j \in [n']$ with $f_{\mathbf{y}_j}(\mathbf{x}_{i,0}) = 1$, set $\mathbf{x}_{i,1}$ to $\mathbf{x}_{i,0}$. From the argument above, there will remain at least one i for which $\mathbf{x}_{i,1} \neq \mathbf{x}_{i,0}$. $\mathcal{B}_1$ builds the following simulator Sim^*. First, $\mathcal{B}_1$ submits $((\mathbf{x}_{i,0})_{i\in[n]}, (\mathbf{x}_{i,1})_{i\in[n]}, (y_j)_{j\in[n']}, \sigma)$ as its first output in the IND-WSS game. Note that we use σ from $\mathcal{A}_1$'s output as $\mathcal{B}_1$'s state. After running Setup, the IND-WSS challenger computes $\mathcal{K}$ and either $\mathbf{c} \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_{i,0}))_{i\in[n]}$ or $\mathbf{c} \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_{i,1}))_{i\in[n]}$. $\mathcal{B}_2$ gets $(pk, \mathcal{K}, \mathbf{c}, t)$. The output of Sim^* is defined as the ciphertext $\mathbf{c}$ obtained from the challenger in the IND-WSS game. At this stage, $\mathcal{B}_2$ calls $\mathcal{A}_2^*(pk, c, \mathcal{K})$ and receives σ'. If $\sigma = \sigma'$ then $\mathcal{B}_2$ outputs $b' = 0$ to the IND-WSS challenger, otherwise they output $b' = 1$. The intuition is that if the challenger has chosen $b = 1$, then $\mathbf{c} \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_{i,1}))_{i\in[n]}$ leads to a σ' that is different from σ with a non-negligible probability compared to a σ' derived from $\mathbf{c} \leftarrow (\mathsf{Enc}_{pk}(\mathbf{x}_{i,0}))_{i\in[n]}$.

Analysis: First, we note the following two properties. (1) $\Pr[b' = b | b = 0] = \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1] = \mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda) + \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1]$ and (2) $\Pr[b' = b | b = 1] = 1 - \Pr[b' = 0 | b = 1]\Pr[b' = b | b = 1] = 1 - \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1]$.

Therefore, $\Pr[b' = b] = \Pr[b' = b | b = 0] \cdot \Pr[b = 0] + \Pr[b' = b | b = 1] \cdot \Pr[b = 1] = (\mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda) + \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1])\frac{1}{2} + (1 - \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1])\frac{1}{2} = \frac{1}{2} + \frac{\mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda)}{2}$.

As a result, adversary $\mathcal{B}$, would win the indistinguishability game IND-WSS with a non-negligible advantage half of the advantage of the adversary in the Sim-WSS experiment. □

B.5 Lemma 4

Proof. Observe that Π_{KDer} is correct, as R, first, retrieves all $K_{j,i}$ corresponding to input $\mathbf{y}$. Second, γ removes all randomness added to the $\kappa_{j,i}$ such that R correctly computes K_A, too.

For security, we show existence of simulators Sim_S for S and Sim_R for R.

$\mathsf{Sim}_S(\mathsf{PE}, msk)$: This simulator for Sender S is trivial, as it only has to run the $\mathcal{F}_{\mathsf{OT}}$-simulator for the OT sender with arbitrary input.

$\mathsf{Sim}_R((\mathbf{y}_j)_{j\in[n_R]}, (sk_{\mathbf{y}_j,\tau})_{j\in[n_R],\tau\in\mathcal{T}})$: Simulator Sim_R for R starts by running the $\mathcal{F}_{\mathsf{OT}}$ simulator for the OT receiver (for each $\mathbf{y}_j, \tau, i$). From its input $sk_{\mathbf{y}_j,\tau}$, Sim_R takes the ℓ values $K_{u,i}$ as input to the OT simulator. To simulate the $\kappa_{u,i}$, it chooses random values $\rho_{u,i} \in \mathbb{G}$ as input to the OT simulator. Observe that the $\rho_{u,i}$ are indistinguishable from the $\kappa_{u,i}$ sent in the real protocol execution. Then S sends $K_{u,\ell+1}$ and another random $\rho_{u,\ell+1} \in \mathbb{G}$ to R to simulate Message (4) from Π_{KDer}.

With K_A being part of $sk_{\mathbf{y}_j,\tau}$ coming from the ideal functionality, Sim_R sends $\gamma = K_A - \sum_{u\in[4],i\in[\ell+1]} \rho_{u,i}$ to R which is indistinguishable from the message sent in the real protocol execution. Finally, Sim_R sends K_B from the ideal functionality's key $sk_{\mathbf{y}_j}$ to R. □

B.6 Theorem 2

Proof. Observe the correctness of Π_{PEI} from the protocol description. Let PE be a Sim-WSS secure predicate encryption scheme for the simplified predicate encryption for the inner-product predicate over $\mathbb{Z}_p^\ell$. In the $\mathcal{F}_{\mathsf{KDer}}$-hybrid model, let PE support an ideal $\mathcal{F}_{\mathsf{KDer}}$ functionality. We need to show the existence of simulators Sim_S and Sim_R capable of generating respective views for S and R that are indistinguishable from real protocol executions.

$\mathsf{Sim}_S((\mathbf{x}_i)_{i\in[n_S]})$*:* We first note that Sender S does not receive any message or output. Hence, its view is trivial to simulate: Sim_S sends random input vectors $\mathbf{y}_j$ to $\mathcal{F}_{\mathsf{KDer}}$. The simulator for $\mathcal{F}_{\mathsf{KDer}}$ generates the corresponding view for S.

$\mathsf{Sim}_R((\mathbf{y}_j)_{j\in[n_R]}, (b_{i,j,\tau}, \hat{\mathbf{x}}_{i,j,\tau})_{i\in[n_S],j\in[n_R],\tau\in\mathcal{T}})$*:* The view of R comprises the view for $\mathcal{F}_{\mathsf{KDer}}$ and ciphertexts $\mathbf{c}_i$. First, Sim_R runs $\mathsf{Setup}(1^\lambda)$, obtains pk and msk, and sends msk to $\mathcal{F}_{\mathsf{KDer}}$ to generate the key derivation view for R. With access to msk, Sim_R can also re-compute keys $\mathcal{K}$ for inputs $\mathbf{y}_j$.

Finally, to simulate the ciphertexts, recall Lemma 3. For any Sim-WSS scheme, there exists a simulator Sim^* that, given input $(pk, (\mathbf{y}_j)_{j\in[n_R]}, (f_{\mathbf{y}_j}(\mathbf{x}_i))_{i\in[n_S],j\in[nR]}, \mathcal{K})$, outputs ciphertexts $\mathbf{c}_S$ such that no adversary can distinguish with a

Experiment $\mathsf{Exp}^{PS}_{\mathsf{PE},\mathcal{A},\mathcal{B}}(\lambda)$

$((\mathbf{x}_i)_{i\in[n]},(\mathbf{y}_j)_{j\in[n']}) \xleftarrow{\$} \mathcal{B}(1^\lambda)$

$(\mathbf{x}'_i)_{i\in[n]} \xleftarrow{\$} \mathcal{A}(1^\lambda,(\mathbf{y}_j)_{j\in[n']},(f_{\mathbf{y}_j}(\mathbf{x}_i)_{i\in[n],j\in[n']}))$

If $(f_{\mathbf{y}_j}(\mathbf{x}'_i)=f_{\mathbf{y}_j}(\mathbf{x}_i))_{i\in[n],j\in[n']}$ output 1, else output 0

Figure 11: Preimage sampleability

non-negligible advantage $(pk,\mathcal{K},\mathbf{c}\leftarrow(\mathsf{Enc}_{pk}(\mathbf{x}_i))_{i\in[n_S]})$ from $(pk,\mathcal{K},\mathbf{c}_S)$. So, Sim_R employs Sim^* to compute the ciphertexts for R as follows.

For all combinations of i and j where there exists a τ such that $b_{i,j,\tau}=1$, Sim_R sends $\mathsf{Enc}_{pk}(\hat{\mathbf{x}}_{i,j,\tau})$ to R.

For all other combinations of i and j, Sim_R uses one of the ciphertexts output by Sim^* when run with input $(pk,(\mathbf{y}_j)_{j\in[n_R]},(b_{i,j,0})_{i\in[n_S],j\in[n_R]},\mathcal{K})$. □

C Preimage Sampleability

Definition 5. *Consider Experiment* $\mathsf{Exp}^{PS}_{\mathsf{PE},\mathcal{A},\mathcal{B}}(\lambda)$ *in Figure 11. A predicate* f *is* preimage sampleable iff *there exists a PPT algorithm* $\mathcal{A}$ *such that, for every PPT algorithm* $\mathcal{B}$*, the probability that* $\mathsf{Exp}^{PS}_{\mathsf{PE},\mathcal{A},\mathcal{B}}(\lambda)$ *outputs* 0 *is negligible in* λ.

Lemma 5. *The inner-product predicate* $f_{\mathbf{y}}(\mathbf{x})=[\langle\mathbf{x},\mathbf{y}\rangle\stackrel{?}{=}0]$ *with* $\mathbf{x},\mathbf{y}\in\mathbb{Z}_p^\ell$ *is preimage sampleable.*

Proof (Sketch, also see O'Neill [53], Proposition 5.1). For each $\mathbf{x}_i$, $\mathcal{A}$ uses the inner-product predicate results to set up a separate system of linear equations. Consider the equations E_0 for which the inner-product is 0 and E_1 the equations for which the inner-product is non-zero. $\mathcal{A}$ computes a base $(\mathbf{b}_k)_{k\in[s]}$ for E_0's kernel using Gaussian elimination. The kernel's dimension s is at least 1 because we already know that at least one solution exists (i.e., $\mathbf{x}$). $\mathcal{A}$ outputs $\mathbf{x}'=\sum_{k=1}^{s}r_kb_k$, a random linear combination of the kernel's base vectors. By construction, $\mathbf{x}'$ satisfies E_0. It also satisfies E_1 with probability at least $1-\frac{n}{p}$. □

D Extension to (Weak) Fuzzy Labeled PSI

While not central to our contribution, we briefly outline a simple extension of Π_{FPSI} for *weak fuzzy labeled PSI*. In labeled PSI [18, 64], sender S inputs tuples $(\mathbf{x}_i,L_i)$, where L_i is a label. Given receiver inputs $\mathbf{y}_j$, the output is $\{L_i \mid \exists(i,j)\text{ s.t. }\mathbf{x}_i=\mathbf{y}_j\}$. Similarly, fuzzy labeled PSI outputs $\{L_i \mid \exists(i,j)\text{ s.t. }\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j)<t\}$. By encrypting L_i instead of $\mathbf{x}_i$ in Π_{FPSI}, we can achieve this functionality with minor adjustments to the security arguments. This approach yields a *weak* fuzzy labeled PSI, as the receiver learns t when $\mathsf{HD}(\mathbf{x}_i,\mathbf{y}_j)<t$, but it might be useful in many real-world scenarios.

E Discussion: Other Metrics

We briefly note that structure-aware FPSI has also been considered for additional metrics, including L_1, L_2, L_n, and L_∞ norms. Chongchitmate et al. [21] demonstrate that their Approx-PSI for Hamming distance can be combined with known low-distortion embeddings for Levenshtein distance [54], Euclidean distance [25], and angular distance [24] to obtain Approx-PSI over these metrics. While these extensions are outside our scope, our FPSI construction can similarly operate on transformed inputs, although the resulting functionality must be adapted.

In the Euclidean case, the functionality becomes: for parameters $0<\delta_0<\delta_1$, the receiver learns all sender elements within distance δ_0, learns nothing about elements at distance at least δ_1, and the behavior for distances in (δ_0,δ_1) is undefined. To extend our FPSI, one might use the additive embedding stated by Chongchitmate et al. [21, Thm. F.4], building on Dirksen and Mendelson [25]. For vectors in a ball $B(R)\subset\mathbb{R}^N$ and any $0<\delta<(\delta_1-\delta_0)/2$, there exists a map $\phi:\mathbb{R}^N\to\{0,1\}^\ell$ such that

$$\left|\alpha\cdot\mathsf{HD}(\phi(x),\phi(y))-\|x-y\|_2\right|\le\delta$$

for all input pairs, where $\alpha>0$ is a scaling factor and $\ell=O\big(\rho^2(\log n+\lambda')/\delta^2\big)$ with $\rho=O\Big(R\sqrt{\log(R/\delta)}\Big)$ and λ' a statistical security parameter. Selecting $\delta=\Theta(\delta_1-\delta_0)$ yields Hamming thresholds $D_{\text{close}}<\tau<D_{\text{far}}$ such that $\|x-y\|_2\le\delta_0$ implies $H(\phi(x),\phi(y))<\tau$, while $\|x-y\|_2\ge\delta_1$ implies $H(\phi(x),\phi(y))>\tau$. Running our FPSI on the embedded inputs with threshold τ therefore enforces the Euclidean fuzzy semantics. Since $\ell=\tilde{O}\big((\log n+\lambda')/(\delta_1-\delta_0)^2\big)$, communication and computation grow by the same factor. A more refined analysis of resulting constants is left for future work.

The embedding itself is computationally lightweight: it samples random hyperplanes and maps each vector to a bit indicating the side on which the vector lies.

Similar conclusions hold for other metrics. For example, applying the Ostrovsky–Rabani embedding for edit distance [54] or the random-hyperplane embedding for angular distance [24] yields analogous Hamming thresholds and thus the same pattern of a guaranteed-reveal region, a guaranteed-hide region, and an undefined gap determined by the embedding's distortion parameters.