Assumption-Free Fuzzy PSI via Predicate Encryption

Erik-Oliver Blass[1] Guevara Noubir[2]

[1]Airbus
Munich, Germany
erik-oliver.blass@airbus.com

[2]Northeastern University
Boston, USA
g.noubir@northeastern.edu

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 satisfy a weak notion of simulation security and 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 features not only asymptotically optimal linear communication complexity but is also concretely practical.

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 [38] and Freedman et al. [22], 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 [33], Meta [9], and Microsoft [39]. 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 [33]. So, current state-of-the-art in PSI features optimal linear communication complexity in the size of the parties' input sets, see [13, 26, 37, 49] for an overview.

While traditional PSI identifies exact matches between elements, many real-world applications require finding elements within a distance threshold. Examples include matching network traffic features in security logs, comparing biometric data like fingerprints, analyzing GPS coordinates, and 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. [22]. In Fuzzy private set intersection (FPSI), two parties 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 [18, 24, 27, 28, 50, 55, 57]. Yet, current solutions achieve linear communication complexity in the dataset size 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.

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, the **technical highlights** of this paper are:

– We present the first scheme to securely realize fuzzy private set intersection for Hamming distance, featuring linear communication complexity in the size

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
[56]	Hamming	FPR/FNR	$O(\ell n_S n_R B_1)$	$S : O(\ell n_S n_R B_2)$ $R : O(\binom{\ell}{t} n_R)$
[12]	Hamming and L_1	FPR	$O(n_S n_R t^2)$	$S : O((\ell + t^2) n_S n_R)$ $R : O((\ell + t) n_S n_R)$
[24]	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.	$O((\ell t + p \log t) n_S + \ell t n_R)$	$S : O((\ell t + p \log t) n_S + n_R)$ $R : O(n_S + \ell t n_R)$
[18]	Hamming (generalizes to other)	$d(x, y) \leq t$ or $d(x, y) \geq \delta t$, $\delta > 3$	$O(n^{1+\frac{1}{\delta-1}})$	$S/R : O(n^{1+\frac{1}{\delta-1}})$
[27]	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.	$O(\ell n_R \log t + n_S)$	$S : O((2t)^\ell n_R)$ $R : O(\ell n_S \log t)$
[57]	L_p, L_∞	n_R receiver balls: radius t, separated $c \cdot t$, $c > 2$	$O(t \ell n_R + 2^\ell n_S)$	$S : O(2^\ell t n_S)$ $R : O(t \ell n_R + 2^\ell n_S))$
		$c > 4$	$O(t 2^d \ell n_R + n_S)$	$S : O(\ell n_S)$ $R : O(t 2^\ell \ell n_R + n_S))$
	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	$c > 2t(\ell^{\frac{1}{p}} + 1)$	$O(t^p n_S + t 2^\ell \ell n_R)$	$S : O((\ell + t^p) n_S)$ $R : O(n_S + t 2^d \ell n_R))$
[50]	L_1, L_2, L_∞	Disjoint Hash, $0 \leq s \leq \ell$	$\Omega(\ell(n_S 2^s + n_R 2^{\ell - s}))$	$S : \Omega(\ell n_S 2^s)$ $R : \Omega(\ell n_R 2^{\ell - s})$
Ours	Hamming	**None**	$O(\ell t(n_S + n_R))$	$S : O(\ell t(n_S + n_R))$ $R : O(\ell t n_S n_R)$

- FPR/FNR: assumes that receiver can tolerate non-negiligible false positive/negative rate.
- R. UniqC: assumes that for each vector of R there exists 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.
- R. disj. proj.: assumes that for each vector $\mathbf{y}$ of R there exists at least one dimension j on which we have $|\mathbf{y}[j] - \mathbf{y}'[j]| > 2t$ for all other elements $\mathbf{y}'$ of R.
- $R \wedge S$. disj. proj.: assumes that disj. proj. assumption holds for sender and receiver sets.

of parties' input sets. Our techniques do not make any restrictive assumption on the structure or distribution of the parties' sets.

- To be able to employ current schemes for inner-product predicate encryption, we introduce notions of weak selective security for preimage sampleable predicate encryption, both as an indistinguishability game-based formulation (IND-WSS) and as a simulation-based formulation (Sim-WSS). We prove, first, that IND-WSS ⇒ Sim-WSS and that current selectively secure predicate encryption schemes are IND-WSS secure.
- We design a two-party, distributed, and concretely practical version of the key derivation scheme for the predicate encryption scheme by Park [48]. A

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.

Fig. 1. Ideal fuzzy PSI functionality $\mathcal{F}_{\mathsf{FPSI}}$

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. Upon publication of the paper, the source code will be made available for download.

For completeness sake, we mention that recent structure-aware FPSI research has addressed various distance metrics, not only Hamming distance, but also L_1, L_2, L_n, and L_∞ norms. Also, Chongchitmate et al. [18] demonstrated how low-distortion embeddings can extend Hamming distance methods to other metrics like Levenshtein (edit) distance, Euclidean distance, and angular distance.

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 the input of S, 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 relation between the Hamming distance of two vectors and their inner product. 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 [36]. So, we convert 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. To compute whether for the two vectors $\mathbf{x}$ and $\mathbf{y}$ their Hamming distance is less than a threshold t, we compute 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 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. 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.

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 and n_R.

However, using predicate encryption again presents two technical challenges. 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 $\mathit{\Pi}_{\mathsf{PEI}}$.

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 the resulting ciphertext $\mathbf{c}_i$ 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 the pair $(\mathbf{x}_i, \mathbf{y}_j)$ 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

Before presenting technical details of our main protocol for fuzzy PSI, 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 $(x_1, \ldots, x_n)$.

Finally, we make use of predicates $[A \overset{?}{=} B]$ that can either evaluate to 1 (true) or 0 (false). If A equals B, then $[A \overset{?}{=} B]$ evaluates to 1, otherwise it evaluates to 0.

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 within the semi-honest security model which assumes that sender or receiver may act as passive adversaries. As a result, our security proofs will each demonstrate the existence of two simulators, one to simulate the sender and one to simulate the receiver. Each will be shown to reproduce

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

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

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

the respective party's view of a real protocol execution, given only inputs and outputs derived from the ideal functionality.

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 $[\langle \mathbf{x}_i, \mathbf{y}_j \rangle \overset{?}{=} \tau]$. 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 exploits a relation between the Hamming distance $\mathsf{HD}(\mathbf{x}, \mathbf{y})$ of two vectors $\mathbf{x}$ and $\mathbf{y}$ and their inner product $\langle \mathbf{x}, \mathbf{y} \rangle$.

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 standard trick [36]. 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 $[\langle \mathbf{x}', \mathbf{y}' \rangle \stackrel{?}{=} \ell - 2t]$. That is, 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, $[\mathsf{HD}(\mathbf{x},\mathbf{y}) \stackrel{?}{<} t] = 1$, we iteratively compute for $\theta \in (0, \dots, t-1)$ 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 $\tau \in \mathcal{T} = \{\ell - 2t + 2, \dots, \ell\}$ 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 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 both an ideal functionality $\mathcal{F}_{\mathsf{HD}}^{<t}$ and a realizing protocol $\mathit{\Pi}_{\mathsf{HD}}^{<t}$ 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 $\mathit{\Pi}_{\mathsf{HD}}^{<t}$. Protocol $\mathit{\Pi}_{\mathsf{HD}}^{<t}$ follows exactly the intuition we have described above.

Lemma 1. *Protocol $\mathit{\Pi}_{\mathsf{HD}}^{<t}$ securely realizes $\mathcal{F}_{\mathsf{HD}}^{<t}$ in the $\mathcal{F}_{\mathsf{PEI}}$-hybrid model with parameter $\mathcal{T} = \{\ell - 2t + 2, \dots, \ell\}$.*

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 $\theta \in \{0, \dots, t-1\}$, 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 $\mathit{\Pi}_{\mathsf{HD}}^{<t}$.

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

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

$$b_{i,j} = [\mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j) \overset{?}{<} t]$$

$$\beta_{i,j} = \begin{cases} \langle \mathbf{x}_i, \mathbf{y}_j \rangle, & \text{if } b_{i,j} = 1 \\ \perp, & \text{otherwise.} \end{cases}$$

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

Fig. 3. Ideal functionality $\mathcal{F}_{\mathsf{HD}}^{<t}$

$\mathsf{Sim}_S((\mathbf{x}_i)_{i \in [n_S]})$: 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 $(\mathbf{y}'_j)_{j \in [n_R]}$. 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 $(0, \perp)$. □

2.2 Fuzzy PSI Protocol $\mathit{\Pi}_{\mathsf{FPSI}}$

With $\mathcal{F}_{\mathsf{HD}}^{<t}$ at hand, the construction of a fuzzy PSI protocol becomes straightforward. In our fuzzy PSI protocol $\mathit{\Pi}_{\mathsf{FPSI}}$ 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 $b_{i,j} = 1$.

Theorem 1. *Protocol $\mathit{\Pi}_{\mathsf{FPSI}}$ securely realizes $\mathcal{F}_{\mathsf{FPSI}}$ in the $\mathcal{F}_{\mathsf{HD}}^{<t}$-hybrid model.*

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 $\mathit{\Pi}_{\mathsf{FPSI}}$ even though it is actually the $\pi(i)^{\text{th}}$ input.

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) Sender 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) Receiver 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 $(b_{i,j} = 0, \beta_{i,j} = \mathbf{z}_{i,j} = \perp)$.

Fig. 4. Protocol $\Pi^{<t}_{\mathsf{HD}}$ realizing $\mathcal{F}^{<t}_{\mathsf{HD}}$ in the $\mathcal{F}_{\mathsf{PEI}}$-hybrid model

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 $\mathcal{F}^{<t}_{\mathsf{HD}}$, and R sends $(\mathbf{y}_j)_{j\in[n_R]}$ to $\mathcal{F}^{<t}_{\mathsf{HD}}$.
2. R receives back $(b_{i,j}, \beta_{i,j}, \mathbf{z}_{i,j})_{i\in[n_R], j\in[n_S]}$ from $\mathcal{F}^{<t}_{\mathsf{HD}}$.
3. For each $b_{i,j} = 1$, R outputs $(\mathbf{z}_{i,j}, \mathbf{y}_j)$.

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

$\mathsf{Sim}_S((\mathbf{x}_i)_{i\in[n_S]})$: Sender S does not receive any message or produce any output, so Sim_S just runs the sender's simulator of the $\mathcal{F}^{<t}_{\mathsf{HD}}$-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 $\mathcal{F}^{<t}_{\mathsf{HD}}$-hybrid with the following input and output. The input for the $\mathcal{F}^{<t}_{\mathsf{HD}}$ simulator is simply $\mathsf{In}_{\mathsf{R}} = (\mathbf{y}_j)_{j\in[n_R]}$. For the output $(b_{i,j}, \beta_{i,j}, \mathbf{z}_{i,j})$ of the $\mathcal{F}^{<t}_{\mathsf{HD}}$ simulator, Sim_R sets

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

□

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. Below, we begin with an introduction to relevant predicate encryption schemes, their challenges, and how to realize $\mathcal{F}_{\mathsf{PEI}}$ with them.

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 [7, 8, 36, 47, 54] 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 [5, 6, 52, 53], 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 [4, 32, 59], 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 two attributes x and y from Σ to either* 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* $f_y(x) = 1$*,*

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

must hold.

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 whether $f_y(x) = 1$, i.e., $[f_y(x) \overset{?}{=} 1]$. Also in this paper, 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 us the same properties as what we target with our simplified predicate encryption, and we discuss differences in Section 3.7. We also point out that several other types of predicate encryption schemes with different security guarantees exist. For a more in-depth introduction, we refer to [7, 36, 47]. We discuss these variations and their use for our main construction also later in Section 3.7.

Of specific interest in this paper are predicate encryption schemes for the prominent *inner-product* predicate [1, 16, 20, 31, 34, 36, 43–46, 48, 60]. 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 will just write predicate encryption as a shorthand for brevity from now on if obvious from the context. We will also stick to vector notation $\mathbf{x}$ for attributes from now on.

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 previously been shown that is difficult to find simulation-based security definitions and prove security in the standard model [7, 11, 47]. Similar to adaptive simulation-based security for public key encryption [42], the main challenge is that the simulator would have 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, 20, 31, 60]. While these schemes could theoretically serve as hybrid functionalities within 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) individual 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 [16, 34, 36, 43–46, 48]. 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 [40, 48]. 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 [47] 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 [16, 34, 36, 43–46, 48] 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 ($\mathsf{Sim\text{-}WSS}$) that we introduce. We

```
**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.
```

Fig. 6. Selective security game-based definition

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 achievable 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 show how to replace the TTP required in predicate encryption schemes to derive keys by either a standard application of 2PC in the general case or even more efficiently by a careful modification of the KDer algorithm in the scheme by Park [48].

3.3 Selective Security

We follow the approach suggested by Boneh et al. [7] and O'Neill [47]. 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 [36, 43, 44, 48] also trivially realizes simplified predicate encryption in the random oracle model.

The main idea of selective security [2, 10, 29] 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 called* IND-SS *secure* iff *for all $PPT(\lambda)$ adversaries $\mathcal{A}$*

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

is negligible in λ.

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 [35]). 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 of Gay [29].

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 commits to both the challenge attribute $\mathbf{x}$ as well as 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 finally use in our proof of Fuzzy PSI. In Section 3.7.3, 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 ones for arbitrary functional encryption discussed by Garg and Srinivasan [25] and the "very selective" security by Agrawal [1].

```
**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.
```

Fig. 7. Weak selective game-based definition

3.4.1 Game-Based Security

We 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 [47]. As with selective security, the difference to these templates is that the adversary has to commit 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. Interestingly, this weaker requirement proves sufficient for our fuzzy PSI construction.

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 7. 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

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

A predicate encryption scheme PE is called IND-WSS secure iff *for all $PPT(\lambda)$ adversaries $\mathcal{A}$,*

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

is negligible in λ.

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 later 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.

```
**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
```

Fig. 8. Preimage sampleability

Lemma 2. *Let* PE *be an* $\mathsf{IND\text{-}SS}$ *secure predicate encryption scheme. Then,* PE *is also* $\mathsf{IND\text{-}WSS}$ *secure.*

Proof. Assume PE is not $\mathsf{IND\text{-}WSS}$ secure, so there exists adversary $\mathcal{A}^* = (\mathcal{A}^*_1, \mathcal{A}^*_2)$ in the $\mathsf{IND\text{-}WSS}$ 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 $\mathsf{IND\text{-}SS}$ 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 $\mathsf{IND\text{-}SS}$ 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 $\mathcal{A}^*$. □

3.4.2 Preimage Sampleability

Before completing the transition from game-based to simulation-based security, we need one final ingredient. The predicate f for which our predicate encryption scheme is defined for must be *preimage sampleable* [47]. Preimage sampleability for f means that, given a sequence of $f_{\mathbf{y}_j}(\mathbf{x})$ for some 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.

Definition 4. *Consider Experiment* $\mathsf{Exp}^{PS}_{\mathsf{PE},\mathcal{A},\mathcal{B}}(\lambda)$ *in Figure 8. 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 3. *The inner-product predicate* $f_{\mathbf{y}}(\mathbf{x}) = [\langle \mathbf{x}, \mathbf{y}\rangle \overset{?}{=} 0]$ *with* $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^\ell$ *is preimage sampleable.*

Proof (Sketch, also see O'Neill [47], 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 $\mathbf{x}' = \sum_{k=1}^{s} r_k b_k$, 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 $1-\frac{n}{p}$. □

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

Fig. 9. Simulation-based security experiments

3.4.3 Simulation-based Security

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

Definition 5 (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 9.*

Let the probability that experiment $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *outputs* 1 *be*

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

Let the probability that experiment $\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda)$ *outputs* 1 *be*

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

A predicate encryption scheme PE *is called* Sim-WSS *secure* iff *there exists a $PPT(\lambda)$ simulator* Sim *such that for all $PPT(\lambda)$ adversaries $\mathcal{A}$,*

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

is negligible in λ.

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 4. *Let* PE *be an* IND-WSS *secure predicate encryption scheme for a preimage sampleable predicate function. Then,* PE *is also* Sim-WSS *secure.*

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 5 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 5 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 5. 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.

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.

Fig. 10. Ideal functionality $\mathcal{F}_{\mathsf{KDer}}$

$$\begin{aligned}
\Pr[b' = b|b = 0] =& \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}real}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1] \\
=& \mathsf{Adv}^{\mathsf{Sim\text{-}WSS}}_{\mathsf{PE},\mathcal{A},\mathsf{Sim}}(\lambda) + \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1] \\
&\text{and} \\
\Pr[b' = b|b = 1] =& 1 - \Pr[b' = 0|b = 1] \\
\Pr[b' = b|b = 1] =& 1 - \Pr[\mathsf{Exp}^{\mathsf{Sim\text{-}WSS\text{-}ideal}}_{\mathsf{PE},\mathcal{A}}(\lambda) = 1].
\end{aligned}$$

Therefore,

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

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. □

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, 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., $[\langle \mathbf{x}, \mathbf{y} \rangle \overset{?}{=} 0]$. However, for functionality $\mathcal{F}_{\mathsf{PEI}}$, we need to test whether the inner product equals any $\tau \in \mathbb{N}$, so $[\langle \mathbf{x}, \mathbf{y} \rangle \overset{?}{=} \tau]$.

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 [36] that allows to check whether $[\langle \mathbf{x}, \mathbf{y} \rangle \overset{?}{=} t]$ for $t \in \mathbb{N}$ by just using the regular functionality for predicate $[\langle \mathbf{x}, \mathbf{y} \rangle \overset{?}{=} 0]$ 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 $(\ell + 1)$ and set them to

$$\mathbf{x}'[1] = \mathbf{x}[1], \ldots, \mathbf{x}'[\ell] = \mathbf{x}[\ell], \mathbf{x}'[\ell + 1] = 1 \text{ and}$$
$$\mathbf{y}'[1] = \mathbf{y}[1], \ldots, \mathbf{y}'[\ell] = \mathbf{y}[\ell], \mathbf{y}'[\ell + 1] = -t.$$

Evaluating the inner product predicate on $\mathbf{x}'$ and $\mathbf{y}'$ as input allows deriving whether the inner product of $\mathbf{x}$ and $\mathbf{y}$ is t, i.e.,

$$[\langle \mathbf{x}', \mathbf{y}' \rangle \overset{?}{=} 0] = [\langle \mathbf{x}, \mathbf{y} \rangle \overset{?}{=} t].$$

So, to support checking for arbitrary products of length-ℓ vectors, our approach is to 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 10. 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 in the following. 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 that is 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.

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. [21] 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 $\mathsf{2PC}$ 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 $\mathsf{2PC}$ evaluation of $\mathcal{C}_{\mathsf{KDer}}$ is efficient and securely realizes $\mathcal{F}_{\mathsf{KDer}}$ by definition.

3.5.2 Concretely practical construction for Park [48]

For the concrete case of the predicate encryption scheme by Park [48] that we employ in our evaluation, there exists a more efficient 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 [48]'s scheme, we briefly summarize the key derivation (Section 4.1 in [48]). 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

INPUT OF S: Master secret key msk of predicate encryption scheme PE for length-(ℓ+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:

1. For each $\mathbf{y}_j$, for $\tau \in \mathcal{T}$,
   (a) let length-($\ell + 1$) vector

   $$\mathbf{y}'[1] = \mathbf{y}_j[1], \ldots, \mathbf{y}'[\ell] = \mathbf{y}_j[\ell], \mathbf{y}'[\ell+1] = -\tau.$$

   (b) for $i \in [\ell]$,
      i. for $u \in [4]$, S chooses random $M_{u,i} \xleftarrow{\$} \mathbb{G}$ and computes

      $$K_{u,i}^{-1} = G_{u,i} - H_{u,i} \qquad K_{u,i}^{1} = G_{u,i} + H_{u,i}$$
      $$\kappa_{u,i}^{-1} = M_{u,i} + f_{u,i} K_{u,i}^{-1} \qquad \kappa_{u,i}^{1} = M_{u,i} + f_{u,i} K_{u,i}^{1}.$$

      ii. S and R engage in 1-out-of-2 OT where S is the OT sender with input $(K_{u,i}^{-1} || \kappa_{u,i}^{-1})_{u \in [4]}$ and $(K_{u,i}^{1} || \kappa_{u,i}^{1})_{u \in [4]}$, and R is the OT receiver with input $\mathbf{y}'[i]$ receiving $K^{\mathbf{y}'[i]}$. R extracts the $K_{u,i}^{\mathbf{y}'[i]}$ which are part of secret key $sk_{\mathbf{y}'}$ and the $(\kappa_{u,i}^{\mathbf{y}'[i]})_{u \in [4]}$.

   (c) For $u \in [4]$, S chooses random $M_{u,\ell+1} \xleftarrow{\$} \mathbb{Z}_p$ and sets

   $$K_{u,\ell+1} = G_{u,\ell+1} + \mathbf{y}'[\ell+1] \cdot H_{u,\ell+1} \text{ and}$$
   $$\kappa_{u,\ell+1} = M_{u,\ell+1} + f_{u,\ell+1} K_{u,\ell+1}.$$

   (d) 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.
   (e) R computes $K_A = \gamma + \sum_{i \in [\ell], u \in [4]} \kappa_{u,i}^{\mathbf{y}[i]} + \sum_{u \in [4]} \kappa_{u,\ell+1}$ which is part of $sk_{\mathbf{y}'}$.
   (f) S computes and sends $K_B = G'$ completing $sk_{\mathbf{y}'}$.

Fig. 11. Protocol $\mathit{\Pi}_{\mathsf{KDer}}$ realizing $\mathcal{F}_{\mathsf{KDer}}$ in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model

$K_{j,i}^{f_{j,i}}$ 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 11 presents protocol $\mathit{\Pi}_{\mathsf{KDer}}$ in full detail.

Lemma 5. *Protocol $\mathit{\Pi}_{\mathsf{KDer}}$ securely realizes $\mathcal{F}_{\mathsf{KDer}}$ from Figure 10 in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model.*

Proof. Observe that $\mathit{\Pi}_{\mathsf{KDer}}$ 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.

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. Sender S invokes $\mathsf{Setup}(1^\lambda)$ and gets (pk, msk). S sends msk to $\mathcal{F}_{\mathsf{KDer}}$.
2. Receiver 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 $(1, \mathbf{z}_{i,j,\tau})$.

Fig. 12. Protocol $\mathit{\Pi}_{\mathsf{PEI}}$ realizing $\mathcal{F}_{\mathsf{PEI}}$ in the $\mathcal{F}_{\mathsf{KDer}}$-hybrid model

$\mathsf{Sim}_R((\mathbf{y}_j)_{j\in[n_R]}, (sk_{\mathbf{y}_{j,\tau}})_{j\in[n_R],\tau\in\mathcal{T}})$: 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 (1d) from $\mathit{\Pi}_{\mathsf{KDer}}$.

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. □

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 [48], and we conjecture that our efficient two party KDer technique from above also applies in their cases [43–46, 60].

There exists a trivial optimization for $\mathit{\Pi}_{\mathsf{PEI}}$ that we have omitted from Figure 11 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 $(K_{u,i}^{-1}||\kappa_{u,i}^{1})$ 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 $\mathit{\Pi}_{\mathsf{PEI}}$ 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 $\mathit{\Pi}_{\mathsf{PEI}}$ in Figure 12.

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

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

Proof. Observe the correctness of $\mathit{\Pi}_{\mathsf{PEI}}$ 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 4. 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 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 $(pk$, $(\mathbf{y}_j)_{j\in[n_R]}$, $(b_{i,j,0})_{i\in[n_S],j\in[n_R]}$, $\mathcal{K})$.

□

3.7 Discussion: Weaker Predicate Encryption

One might argue that basing our construction of protocol $\mathit{\Pi}_{\mathsf{PEI}}$ and thus also $\mathit{\Pi}_{\mathsf{FPSI}}$ 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.

3.7.1 Using Strong Predicate-Only Predicate Encryption

A strong attribute-hiding *predicate-only* predicate encryption scheme is sufficient for $\mathit{\Pi}_{\mathsf{FPSI}}$. In such a scheme [36], the only information the receiver R obtains is $[f_{\mathbf{y}_j}(\mathbf{x}_i) \overset{?}{=} 1]$. Naturally, for each index i for which R has learned that $f_{\mathbf{y}_j}(\mathbf{x}_i) = 1$, R could then use 1-out-of-n_S Oblivious Transfer or symmetric PIR and privately fetch $\mathbf{x}_i$ in S' input set. As we are in the semi-honest security model, this approach would be secure, because R will only ask for the indices they are supposed to. Yet, as the size of the intersection might be equal to n_S, R would need to query for a total of n_S elements (many dummy elements) not to leak the actual size of the intersection to S. Moreover, 1-out-of-n_S OT or symmetric PIR also incur a significant overhead in either communication, computation, or both. In contrast, the additional cost of payload-hiding predicate encryption schemes over predicate-only schemes is surprisingly small with the current state of the art. Their main approach for payload-hiding is to first design a predicate-only scheme that realizes a key encapsulation mechanism (KEM). Then, in case $f_{\mathbf{y}_j}(\mathbf{x}_i) = 1$, R learns a (symmetric) key that can be used to decrypt another ciphertext back to m. For example, in the works by Katz et al. [36] or Park [48], the only additional operation required to achieve payload-hiding over predicate-only encryption is a cheap hash performed during decryption.

3.7.2 Using Weak Predicate-Only Predicate Encryption

Another weaker type of predicate encryption could be a predicate-only predicate encryption scheme that automatically outputs attribute $\mathbf{x}$ in the clear as soon as $f_{\mathbf{y}}(\mathbf{x}) = 1$. This could be sufficient to realize $\mathcal{F}_{\mathsf{PEI}}$. Instead of the extra encryption of $\mathbf{x}$ and decryption later, receiver R would get $\mathbf{x}$ already for free in case $f_{\mathbf{y}}(\mathbf{x}) = 1$. However, we are not aware of an efficient predicate-only predicate encryption scheme for the inner-product predicate that provides this type of security. We stress that this type of security, revealing the attribute "for free" in case $f_{\mathbf{y}}(\mathbf{x}) = 1$ is very different from the (established) notion of *weakly* attribute-hiding predicate encryption, see, e.g., Gorbunov et al. [31]. In weakly attribute-hiding schemes, only some information about $\mathbf{x}$ is potentially leaked to R in addition to $[f_{\mathbf{y}}(\mathbf{x}) \overset{?}{=} 1]$, but not $\mathbf{x}$ as a whole. So, the current state of the art imposes that *weak attribute-hiding predicate-only encryption* is not sufficient, and the additional encryption of $\mathbf{x}$ is required.

The current state of the art in predicate encryption for the inner product predicate does not incur a significant drawback over weaker predicate-only constructions, and we leave basing Fuzzy PSI on weaker variations of predicate encryption to future work.

3.7.3 Sim-WSS-Secure Predicate Encryption outside of $\mathcal{F}_{\mathsf{PEI}}$

While we do not claim any specific utility for Sim-WSS-secure predicate encryption outside our fuzzy private set intersection scenario, we briefly discuss the implications and meaning of this security notion for real-world scenarios. Recall that Sim-WSS denotes a selective security notion in which the adversary must commit to both the challenge inputs $\mathbf{x}_i$ and the inputs $\mathbf{y}_j$ for KDer queries before receiving the public key.

Consequently, $\mathsf{Sim\text{-}WSS}$-secure predicate encryption schemes have limited general applicability [14]. Specifically, such a scheme is only secure for the encryption of data that was committed to or fixed before publishing the public key. While the adversary can always influence the distribution of new plaintexts, selective security only guarantees protection of inputs committed to before the public key is released. Thus, once the public key is published, the adversary's control over plaintext selection compromises security. Similarly, the scheme is secure only if parties commit to the secret keys they wish to obtain before the public key is made available. Interestingly, this proves sufficient to achieve a secure $\mathcal{F}_{\mathsf{FPSI}}$ with linear communication complexity as the parties' one-time input set to predicate encryption is fixed before running Setup and then submitted in a batch.

If parties want to perform predicate encryption and key derivation of input chosen after publication of the public key, one would have to reset the whole system, let parties commit to their input, and then choose a new public key.

3.8 Extension to Fuzzy Labeled-PSI

While not at the core of our contribution, we briefly highlight that Π_{FPSI} can be modified in a straightforward manner to also support a fuzzy PSI extension one might call fuzzy labeled-PSI.

In standard labeled PSI [15, 56], the sender's input is a set of tuples (x_i, L_i), where L_i is called a *label.* For the receiver set of inputs y_j, The output of labeled PSI is the set $\{L_i | \exists (i,j) \text{ s.t. } x_i = y_j\}$. Along the same lines, we can define the output of fuzzy labeled PSI for sender input set $(\mathbf{x}_i, L_i)$ and receiver input set $\mathbf{y}_j$ to be set $\{L_i | \exists (i,j) \text{ s.t. } \mathsf{HD}(\mathbf{x}_i, \mathbf{y}_j) < t\}$. Using our predicate encryption approach, we can easily implement this functionality by encrypting L_i instead of $\mathbf{x}_i$ and making minor adjustments to our security arguments.

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 show concrete practicality of Π_{FPSI}. We report on the concrete performance of Π_{FPSI} **without** directly **comparing** it to existing protocols. Related work relies on strong assumptions about data structure to optimize performance. Such assumptions are not required for this work, as our techniques support arbitrary input conditions. Thus, any direct comparison would be both uninformative and inherently unfair.

Our implementation is written in C++ and will be made available upon publication of the paper. At its core, we have re-implemented the predicate encryption scheme by Park [48]. 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 $\mathsf{Sim\text{-}WSS}$ secure.

Table 2. Benchmark results for protocol Π_{FPSI}. Comm: total data exchanged between sender and receiver, Time: total runtime, n_S and n_R: number of input vectors from sender and receiver, ℓ: length of each vector, t: threshold for Hamming distance

		$\ell = 16$		$\ell = 32$		$\ell = 64$		$\ell = 128$		$\ell = 256$		$\ell = 512$		$\ell = 1024$	
	$n_R = 64$	Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time
t	$n_S =$	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)
2	32	2.5	0.3	3.6	0.5	5.9	1.0	10.4	1.9	19.4	3.6	37.4	7.1	73.4	14.0
	64	3.5	0.6	5.6	1.0	9.8	1.9	18.3	3.6	35.3	7.0	69.3	14.0	137.3	27.8
	128	5.4	1.0	9.5	1.9	17.8	3.6	34.3	7.0	67.3	14.0	133.3	27.7	265.3	55.4
	256	9.2	1.9	17.3	3.6	33.6	7.1	66.1	14.0	131.1	27.8	261.1	55.3	521.1	110.5
4	32	3.5	0.6	5.6	1.0	9.8	1.9	18.3	3.6	35.3	7.0	69.3	14.0	137.3	27.8
	64	5.4	1.0	9.5	1.9	17.8	3.6	34.3	7.0	67.3	14.0	133.3	27.7	265.3	55.4
	128	9.2	1.9	17.3	3.6	33.6	7.1	66.1	14.0	131.1	27.8	261.1	55.3	521.1	110.5
	256	16.9	3.7	33.0	7.1	65.3	14.0	129.8	27.8	258.8	55.4	516.8	110.4	1032.8	220.7
8	32	5.4	1.0	9.5	1.9	17.8	3.6	34.3	7.0	67.3	14.0	133.3	27.7	265.3	55.4
	64	9.2	1.9	17.3	3.6	33.6	7.1	66.1	14.0	131.1	27.8	261.1	55.3	521.1	110.5
	128	16.9	3.7	33.0	7.1	65.3	14.0	129.8	27.8	258.8	55.4	516.8	110.4	1032.8	220.7
	256	32.3	7.3	64.4	14.2	128.7	28.0	257.2	55.5	514.2	110.6	1028.2	220.7	2056.2	441.2
16	32	9.2	1.9	17.3	3.6	33.6	7.1	66.1	14.0	131.1	27.8	261.1	55.3	521.1	110.5
	64	16.9	3.7	33.0	7.1	65.3	14.0	129.8	27.8	258.8	55.4	516.8	110.4	1032.8	220.7
	128	32.3	7.3	64.4	14.2	128.7	28.0	257.2	55.5	514.2	110.6	1028.2	220.7	2056.2	441.2
	256	63.0	14.5	127.2	28.2	255.4	55.8	511.9	110.9	1024.9	221.1	2050.9	441.2	4102.9	882.2

		Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time	Comm	Time
t	$n_S = n_R$	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)	(MByte)	(s)
2	32	2.4	0.2	3.5	0.3	5.6	0.6	9.9	1.0	18.4	1.9	35.4	3.7	69.4	7.3
	64	3.5	0.6	5.6	1.0	9.8	1.9	18.3	3.6	35.3	7.0	69.3	14.0	137.3	27.8
	128	5.5	1.9	9.8	3.6	18.3	6.9	35.3	13.7	69.3	27.3	137.3	54.3	273.3	108.4
	256	9.6	7.1	18.1	13.8	35.1	27.2	69.1	53.9	137.1	107.4	273.1	214.3	545.1	428.4
4	32	3.4	0.3	5.5	0.6	9.6	1.0	17.8	1.9	34.3	3.7	67.3	7.3	133.3	14.5
	64	5.4	1.0	9.5	1.9	17.8	3.6	34.3	7.0	67.3	14.0	133.3	27.7	265.3	55.4
	128	9.4	3.6	17.6	7.0	34.1	13.8	67.1	27.3	133.1	54.3	265.1	108.3	529.1	216.5
	256	17.3	14.1	33.8	27.4	66.8	54.2	132.8	107.6	264.8	214.6	528.8	428.2	1056.8	856.1
8	32	5.3	0.6	9.4	1.0	17.5	1.9	33.7	3.7	66.2	7.3	131.2	14.5	261.2	28.8
	64	9.2	1.9	17.3	3.6	33.6	7.1	66.1	14.0	131.1	27.8	261.1	55.3	521.1	110.5
	128	17.0	7.2	33.3	13.9	65.8	27.4	130.8	54.4	260.8	108.5	520.8	216.4	1040.8	432.5
	256	32.7	28.0	65.2	54.8	130.2	108.2	260.2	215.0	520.2	429.0	1040.2	855.9	2080.2	1711.5
16	32	9.2	1.0	17.2	1.9	33.3	3.7	65.6	7.3	130.1	14.5	259.1	28.8	517.1	57.5
	64	16.9	3.7	33.0	7.1	65.3	14.0	129.8	27.8	258.8	55.4	516.8	110.4	1032.8	220.7
	128	32.4	14.2	64.7	27.7	129.2	54.7	258.2	108.7	516.2	216.7	1032.2	432.4	2064.2	864.6
	256	63.4	55.9	127.9	109.4	256.9	216.4	514.9	429.9	1030.9	857.7	2062.9	1711.5	4126.9	3422.2

In contrast to its previous implementation [40], we have ported Park [48]'s scheme to the popular MCL library [41] which has allowed easy adoption of the original KDer algorithm to our distributed setting (Figure 11). Cryptographic operations are performed over the Type-3 BN-254 curve and use 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 [58]. 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 predicate encryption scheme. We use AES-based hash function Blake2 and AES-based PRG from cryptoTools [51]

Table 2 summarizes our benchmark results. All benchmarks were performed on a single Amazon AWS c7i.metal-48xl instance, i.e., without taking network latency into account. Yet, total runtime ("Time" in Table 2) in most scenarios will be dominated by computation time due to the quadratic complexity of Receiver R trying to decrypt all ciphertexts with all secret keys. Yet, as decryptions can be trivially parallelized, we stress that total runtime will greatly benefit from running on a machine with more cores.

The total runtime in Table 2 includes the time to encrypt all sender inputs, performing the OT-based distributed key derivation, and decryption by R. Communication cost ("Comm") includes all n_S ciphertexts and distributed key derivation (OT plus sending the two encrypted choices).

5 Related Work

Fuzzy PSI: The commercial success of PSI has revived interests in Fuzzy PSI. Recently, several schemes have been developed that aim to achieve linear communication complexity, computation complexity, or both. A common strategy consists of clustering input data to reduce complexity. However, these techniques make strong assumptions about the distribution of the data such as distance between elements and the minimal separation between clusters.

Uzun et al. [56] were among the first to propose a Fuzzy Labeled PSI scheme (FLPSI) that targets linear communication. FLPSI uses Locality Sensitive Hashing (LSH) and noise removal techniques to map samples from an Euclidean space to bitstrings amenable to Hamming distance comparisons. The paper uses a combination of subsampling (using masks) and 2PC computation to derive a small set of inputs on both the sender and receiver sides to be fed into existing exact Labeled PSI schemes. Additional measures are taken to reduce leakage, leveraging threshold secret sharing on the labels, and FHE to hide partial matching. However, FLPSI is defined for a closeness function that provides only probabilistic guarantees, and only for when input elements are close (i.e., matching) or far (i.e., non matching), but is not defined for elements that are neither (near). It does not provide formal guarantees for Hamming distance thresholds.

Garimella et al. [27] introduce the notion of structure-aware PSI (sa-PSI) to achieve communication complexity that is linear in the sender's set description size, not its cardinality. They initiate the work of formally exploiting the structure of parties' elements and introduce a generic paradigm for structure-aware PSI based on a new weak boolean function secret-sharing. The paradigm is applied to Fuzzy PSI as an application, specifically considering the structure when the sender's set is defined by balls of radius t, and the metric is L_∞ in an ℓ-dimensional space. Garimella et al. derive different protocols depending on additional constraints. For disjoint balls, communication complexity is in $O((4\log(t))^\ell$ and reduces to $O(2^\ell)$ for balls separated by $4t$. They also show that, for a globally-axis-disjoint structure (i.e., the projection of the balls onto every axis is disjoint), communication complexity becomes linear in the dimension. In a follow-up work, Garimella et al. [28] introduce the first sa-PSI protocol that is also secure against malicious adversaries, by using a cut-and-choose technique and by applying new derandomizable function secret-sharing.

Chakraborti et al. [12] propose FPSI schemes for Hamming distance and integer distance. 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 additional leakage when elements are within the $(t, 2t)$ interval [30], [12] 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. [55] present FPHE, a Fuzzy PSI scheme for cosine similarity. FPHE reduces computation and communication to be linear with respect to the dimension of each set element. They builds on fully homomorphic encryption, optimizing it for approximate sign function evaluation. Unfortunately, FPHE requires that the sender elements are separated by at least twice the threshold (i.e., $2t$).

Richardson et al. [50] generalize the PSI scheme by Cho et al. [17] to provide Fuzzy PSI 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. [24] present Fuzzy mapping (Fmap), an abstraction of previous approaches using coarse mapping to group (bin or cluster) sender and receiver elements to reduce the number of PSI executions (refined filtering). The underlying assumption of Fmap is that the for receiver 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. [18] propose a Fuzzy PSI scheme 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.

To summarize, recent schemes have demonstrated significant reductions in communication and computational complexity for Fuzzy PSI. These methods are particularly powerful and efficient within contexts where input data aligns with their structural assumptions. Nevertheless, their inherent dependence on data structure limits general applicability, a limitation addressed by our approach that achieves efficient Fuzzy PSI regardless of input data structure.

Predicate Encryption: Functional encryption and predicate encryption are active research areas, yielding both foundational results [7, 47] and practical schemes [1, 16, 20, 31, 34, 36, 43–46, 48, 60]. Our work leverages inner-product predicate encryption to demonstrate that Fuzzy PSI is achievable with minimal security assumptions, i.e., weak selective security (IND-WSS). While we instantiate our construction using Park's scheme due to its efficiency and ease of implementation, our approach is not limited to this specific scheme; any predicate encryption scheme satisfying IND-WSS or selective security can be substituted.

6 Conclusion

We have presented a new, efficient protocol for Fuzzy Private Set Intersection (FPSI) that achieves linear communication complexity while avoiding restrictive assumptions on input distributions. Our approach leverages inner-product predicate encryption, reducing secure Hamming distance computation to a secure inner-product test. By establishing a weak simulation-based security definition for predicate encryption, we have demonstrated that existing selectively secure schemes suffice for our protocol. Furthermore, we have introduced a distributed key derivation mechanism, eliminating the need for a trusted third party setup while maintaining efficiency. As indicated by our implementation, our construction not only achieves optimal linear communication complexity, but is also concretely practical for various real-world parameter settings.

Bibliography

[1] Shweta Agrawal. Stronger Security for Reusable Garbled Circuits, General Definitions and Attacks. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2017.

[2] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From Selective to Adaptive Security in Functional Encryption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 657–677. Springer, 2015.

[3] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*, pages 62–73. ACM, 1993.

[4] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pages 321–334. IEEE Computer Society, 2007.

[5] Dan Boneh and Xavier Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[6] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[7] Dan Boneh, Amit Sahai, and Brent Waters. Functional Encryption: Definitions and Challenges. In Yuval Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.

[8] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.

[9] Prasad Buddhavarapu, Andrew Knox, Payman Mohassel, Shubho Sengupta, Erik Taubeneck, and Vlad Vlaskin. Private matching for compute. Cryptology ePrint Archive, Paper 2020/599, 2020. URL `https://eprint.iacr.org/2020/599`.

[10] Ran Canetti, Shai Halevi, and Jonathan Katz. A Forward-Secure Public-Key Encryption Scheme. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003.

[11] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the Achievability of Simulation-Based Security for Functional Encryption. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 519–535. Springer, 2013.

[12] Anrin Chakraborti, Giulia Fanti, and Michael K. Reiter. Distance-Aware private set intersection. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 319–336, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. URL `https://www.usenix.org/conference/usenixsecurity23/presentation/chakraborti-intersection`.

[13] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivio us PRF. In *40th International Cryptology Conference (CRYPTO)*, 2020.

[14] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: practical issues in cryptography. In *Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology: Second International Conference, Mycrypt 2016, Kuala Lumpur, Malaysia, December 1-2, 2016, Revised Selected Papers 2*, pages 21–55. Springer, 2017.

[15] Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. Labeled PSI from Fully Homomorphic Encryption with Malicious Security. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1223–1237. ACM, 2018.

[16] Jie Chen, Junqing Gong, and Hoeteck Wee. Improved Inner-Product Encryption with Adaptive Security and Full Attribute-Hiding. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 673–702. Springer, 2018.

[17] Chongwon Cho, Dana Dachman-Soled, and Stanislaw Jarecki. Efficient concurrent covert computation of string equality and set intersection. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2016. `https://doi.`

org/10.1007/978-3-319-29485-8_10. URL https://doi.org/10.1007/978-3-319-29485-8_10.

[18] Wutichai Chongchitmate, Steve Lu, and Rafail Ostrovsky. Approximate PSI with near-linear communication. Cryptology ePrint Archive, Paper 2024/682, 2024. URL https://eprint.iacr.org/2024/682.

[19] Ronald Cramer and Victor Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM J. Comput.*, 33(1):167–226, 2003.

[20] Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. Adaptively Simulation-Secure Attribute-Hiding Predicate Encryption. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 640–672. Springer, 2018.

[21] David Evans, Vladimir Kolesnikov, and Mike Rosulek. *A Pragmatic Introduction to Secure Multi-Party Computation.* Now Publishers Inc, 2018.

[22] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 1–19, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24676-3.

[23] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *J. Cryptol.*, 26(1):80–101, 2013.

[24] Ying Gao, Lin Qi, Xiang Liu, Yuanchao Luo, and Longxin Wang. Efficient fuzzy private set intersection from fuzzy mapping. Cryptology ePrint Archive, Paper 2024/1462, 2024. URL https://eprint.iacr.org/2024/1462.

[25] Sanjam Garg and Akshayaram Srinivasan. Single-Key to Multi-Key Functional Encryption with Polynomial Loss. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 419–442, 2016.

[26] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Oblivious key-value stores and amplification for private set intersection. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II 41*, pages 395–425. Springer, 2021.

[27] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 323–352, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-15802-5.

[28] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In Helena Handschuh and Anna

Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 577–610, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-38557-5.

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

[30] Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 3–29, Cham, 2019. Springer International Publishing. ISBN 978-3-030-26951-7.

[31] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate Encryption for Circuits from LWE. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523. Springer, 2015.

[32] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006*, pages 89–98. ACM, 2006.

[33] Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On deploying secure computing: private intersection-sum-with-cardinality. In *5th IEEE European Symposium on Security and Privacy (EuroS&P)*, 2020.

[34] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Adaptively Secure Inner Product Encryption from LWE. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part III*, volume 12493 of *Lecture Notes in Computer Science*, pages 375–404, 2020.

[35] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Third Edition.* Chapman & Hall, 2020.

[36] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.

[37] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set inters ection. In *22nd ACM Conference on Computer and Communications Security (CCS)*, 2016.

[38] Catherine Meadows. A More Efficient Cryptographic Matchmaking Protocol for Use in the Absence of a Continuously Available Third Party. In

Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, California, USA, April 7-9, 1986, pages 134–137. IEEE Computer Society, 1986.

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

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

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

[42] Jesper Buus Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2002.

[43] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical Predicate Encryption for Inner-Products. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2009.

[44] Tatsuaki Okamoto and Katsuyuki Takashima. Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.

[45] Tatsuaki Okamoto and Katsuyuki Takashima. Efficient (Hierarchical) Inner-Product Encryption Tightly Reduced from the Decisional Linear Assumption. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 96-A (1):42–52, 2013.

[46] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 99-A(1):92–117, 2016.

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

[48] Jong Hwan Park. Inner-product encryption under standard assumptions. In *Designs, Codes and Cryptography*, volume 58, pages 235–257, 2011.

[49] Srinivasan Raghuraman and Peter Rindal. Blazing Fast PSI from Improved OKVS and Subfield VOLE. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los*

Angeles, CA, USA, November 7-11, 2022, pages 2505–2517. ACM, 2022. https://doi.org/10.1145/3548606.3560658. URL https://doi.org/10.1145/3548606.3560658.

[50] David Richardson, Mike Rosulek, and Jiayu Xu. Fuzzy PSI via oblivious protocol routing. Cryptology ePrint Archive, Paper 2024/1642, 2024. URL https://eprint.iacr.org/2024/1642.

[51] Peter Rindal. cryptoTools library, 2025. https://github.com/ladnir/cryptoTools.

[52] Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[53] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[54] Emily Shen, Elaine Shi, and Brent Waters. Predicate Privacy in Encryption Systems. In Omer Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2009.

[55] Hyunjung Son, Seunghun Paik, Yunki Kim, Sunpill Kim, Heewon Chung, and Jae Hong 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.

[56] Erkam Uzun, Simon P. Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. Fuzzy labeled private set intersection with applications to private Real-Time biometric search. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 911–928. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/uzun.

[57] Aron van Baarsen and Sihang Pu. Fuzzy private set intersection with large hyperballs. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 340–369, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-58740-5.

[58] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, 2016.

[59] Brent Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*,

volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2011.

[60] Hoeteck Wee. Attribute-Hiding Predicate Encryption in Bilinear Groups, Revisited. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 206–233. Springer, 2017.

A Simplified Predicate Encryption from 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 to 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 [3, 19, 23] 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 $m \overset{\$}{\leftarrow} \mathcal{M}$ 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 [3].