Doubly Efficient Fuzzy Private Set Intersection for High-dimensional Data with Cosine Similarity

Hyunjung Son[1], Seunghun Paik[1], Yunki Kim[1], Sunpill Kim[1], Heewon Chung[2], and Jae Hong Seo[1]⋆

[1] Hanyang University, Seoul, Republic of Korea
{dk9050rx,whitesoonguh,dbsrl7665,ksp0352,jaehongseo}@hanyang.ac.kr
[2] Independent Researcher, Seoul, Republic of Korea
{tyler.heewonchung}@gmail.com

Abstract. Fuzzy private set intersection (Fuzzy PSI) is a cryptographic protocol for privacy-preserving similarity matching, which is one of the essential operations in various real-world applications such as facial authentication, information retrieval, or recommendation systems. Despite recent advancements in fuzzy PSI protocols, still a huge barrier remains in deploying them for these applications. The main obstacle is the high-dimensionality, *e.g.*, from 128 to 512, of data; lots of existing methods, Garimella et al. (CRYPTO'23, CRYPTO'24) or van Baarsen et al. (EUROCRYPT'24), suffer from exponential overhead on communication and/or computation cost. In addition, the dominant similarity metric in these applications is cosine similarity, which disables several optimization tricks based on assumptions for the distribution of data, *e.g.*, techniques by Gao et al. (ASIACRYPT'24). In this paper, we propose a novel fuzzy PSI protocol for cosine similarity, called FPHE, that overcomes these limitations at the same time. FPHE features linear complexity on both computation and communication with respect to the dimension of set elements, only requiring much weaker assumption than prior works. The basic strategy of ours is to homomorphically compute cosine similarity and run an approximated comparison function, with a clever packing method for efficiency. In addition, we introduce a novel proof technique to harmonize the approximation error from the sign function with the noise flooding, proving the security of FPHE under the semi-honest model. Moreover, we show that our construction can be extended to support various functionalities, such as labeled or circuit fuzzy PSI. Through experiments, we show that FPHE can perform fuzzy PSI over 512-dimensional data in a few minutes, which was computationally infeasible for all previous proposals under the same assumption as ours.

Keywords: Fuzzy Private Set Intersection · Fuzzy Matching · Homomorphic Encryption

⋆ Corresponding Author.

1 Introduction

In recent years, data-driven technologies such as artificial intelligence (AI) and data collaboration have rapidly advanced, offering innovative and useful services such as biometric identification systems [21,34], image or text retrieval [31,41], recommendation systems [32], or collaborative data analysis [47]. During operating these services, one of the essential operations is *similarity matching*, *i.e.*, measuring the similarity between the data held by the client and stored in the service provider. However, the exchanged data would contain sensitive information of each party, such as biometrics or location information, which could result in privacy risks such as unauthorized surveillance and potential misuse. Therefore, in compliance with regulations on data privacy such as GDPR [55] or CCPA [10], it is necessary to devise a safeguard for deploying these services.

Private set intersection (PSI) [20,28,50,48,7] is one of the notable candidates for addressing the above-mentioned problem, which is a cryptographic protocol that allows parties to compute the intersection between their datasets without revealing any additional information. For decades, several studies have been made to improve the efficiency and functionality of PSI protocols, and these achievements enabled practical deployment of PSI protocols in various application scenarios. For example, private matching for compute [9,44,43] from Meta, PSI [6] from Apple, and private join and compute [33,37] from Google enable computing intersections and additional functionality between two parties while protecting the underlying data privacy. However, traditional PSI protocols have been focused on the *exact* matching only; their methods are not suitable for similarity matching.

Recently, *fuzzy* private set intersection (fuzzy PSI), which is a PSI protocol supporting both exact and similarity matching, was recently highlighted and actively studied [54,29,30,13,3,27,26,49]. Several fuzzy PSI protocols have been proposed for various similarity metrics, including Hamming distance between binary vectors, or Euclidean distance (ℓ_p for $p \in \mathbb{N} \cup \{\infty\}$) between integer-valued vectors. Moreover, recent proposals have achieved various functionalities, such as labeled PSI or circuit PSI [3,27], or stronger security notions, including sender privacy [3] or malicious security [30]. However, despite their theoretical achievements, we found that existing fuzzy PSI protocols would not be applicable in our target scenarios in practice. In our target scenarios, each set element is often represented by high-dimensional real-valued vectors, *e.g.*, from 128 to 512 in face recognition or recommendation system, and the dominating similarity metric is cosine similarity. On the other hand, to our knowledge, existing proposals (1) suffer from exponential communication and/or communication overhead with respect to the dimension of set elements [29,30,27,3], or (2) require strong assumptions on the distribution of data, *e.g.*, each set data should be far from each other [29,30,3,49] or all the neighborhoods of data should not overlap in at least one [3,26] or all axes [29,30], to greatly reduce computation or communication costs. However, we found that their assumptions are *all* disabled when dealing with cosine similarity. Therefore, all these proposals are insufficient for deploying to our target applications.

Metric	Protocol	Assumption	Communication	Computation	
				Sender	Receiver
L_∞	[3]	R. $> 2T$	$O(TdN + 2^d M)$	$O(2^d dM)$	$O(TdN + 2^d M)$
		R. $> 4T$	$O(T2^d dN + M)$	$O(dM)$	$O(T2^d dN + M)$
		R. disj. proj.	$O((Td)^2 N + M)$	$O(d^2 M)$	$O((Td)^2 N + M)$
	[26]	R. $\wedge$ S. disj. proj.	$O(TdM + TdN)$	$O(TdM + N)$	$O(M + TdN)$
	[27]	-	$O(Nd \log T + NM(\log T)^d)$	$O(M(\log T)^d + d \log T)$	$O(N(\log T)^d + Md \log T)$
$L_{p\in[1,\infty)}$	[3]	R. $> 2T(d^{1/p} + 1)$	$O(T^p M + T2^d dN)$	$O((d + T^p)M)$	$O(M + T2^d dN)$
	[26]	R $\wedge$ S. disj. proj.	$O((Td + p \log T)M + TdN)$	$O((Td + p \log T)M + N)$	$O(p \log TM + TdN)$
Cos. Sim.	**Ours**	S.$> 2T$	$d + 1$ Ctxt	$\lceil \frac{2NM}{n} \rceil d$ PtCtMul, $\lceil \frac{2NM}{n} \rceil$ Sign	d Enc

Table 1. Comparison of fuzzy PSI protocols across various metrics. N, M: number of set elements held by receiver and sender, respectively. d: dimension of each set element. T: threshold. n: ring dimension of FHE. Ctxt: FHE ciphertext. Enc: FHE encryption. PtCtMul: plaintext-ciphertext multiplication in FHE. Sign: evaluation of *sgn* in FHE. Note that assumptions in **red** turn out not to be applicable in cosine similarity.

- R. $> v$: the distance between the elements in the receiver's set is greater than v.
- S. $> v$: the distance between the elements in the sender's set is greater than v.
- R. disj. proj.: for the receiver's set, there exists at least one dimension where the distance between the projections of elements exceeds $2T$.
- R. $\wedge$ S.: the above assumption holds for both the receiver's and sender's sets.

Contribution. In this paper, we propose a novel fuzzy PSI protocol called FPHE, which is tailored for dealing with high-dimensional data using cosine similarity as a similarity metric. Compared to previous fuzzy PSI proposals, FPHE is doubly efficient under the assumption that the neighborhoods of each set element do not overlap, as it achieves both linear communication and computation cost with respect to the dimension of each set element. This enables us to handle high-dimensional data such as 512, which was computationally infeasible in all the existing fuzzy PSI constructions under the same assumption. We compare our PFHE with other previous fuzzy PSI protocols in Tab. 1. We identify that the assumptions of the works have limitations in the context of our scenarios (see Section 3.1).

The main ingredient of FPHE is to homomorphically perform the similarity matching procedure using CKKS [17], which is a fully homomorphic encryption (FHE) scheme supporting real-valued arithmetic, and an (approximated) sign function, with a clever encoding method to reduce the computation. We also prove the security of FPHE under the semi-honest model by introducing a new proof technique to incorporate the approximation error of a circuit into the noise flooding technique. Moreover, we show that FPHE can be extend to its functionalities without significant overhead, such as *labeled* fuzzy PSI or *circuit* fuzzy PSI, to cover broader application scenarios.

As a proof-of-concept, we instantiate FPHE using the sign function approximation by Cheon et al. [18] and OpenFHE [1] library. FPHE successfully performs the fuzzy PSI on 512-dimensional data in a few minutes, which was computationally infeasible, not even considered, in all existing proposals. To facilitate further study, we released our source code on github[3].

[3] https://github.com/Cryptology-Algorithm-Lab/FPHE

1.1 Technical Overview

Fuzzy PSI protocol using FHE. A typical approach to constructing an exact PSI through FHE is to subtract elements from each party and homomorphically multiply all of them; the decrypted result would be zero if there is an overlapping element. However, for the case of fuzzy PSI, this approach is no longer applicable because we now need to determine whether these two elements are sufficiently close or not, *i.e.*, within the predetermined threshold. We detour this issue by an alternative approach: computing similarity scores between elements in an encrypted domain and homomorphically evaluating a sign function. The reason to evaluate the sign function is to distinguish whether the similarity score exceeds the threshold or not. Intuitively, our protocol goes as follows: first, the receiver encrypts his/her input and sends it to the sender. The sender homomorphically computes the matching scores and then evaluates the sign function. Finally, the receiver retrieves the final result from the sender and determines whether there is an overlapping element or not by decrypting it.

Achieving Semi-honest Security. Although the above high-level construction seems to satisfy the desirable functionality, it fails to satisfy the semi-honest security without appropriate treatments. Note that CKKS itself does not satisfy *circuit privacy* [38]; the ciphertext after evaluation could leak information about the private input of the circuit, which is the set held by the sender in our context. To mitigate this, a typical solution is for the sender to employ the noise flooding technique [39] on the final ciphertext being sent to the sender. However, we found that applying it *as is* would not ensure the semi-honest security because our circuit is an approximation of the sign function. Note that the approximation error also leaks information about the sender's data, independent of the noise flooding technique. We resolved this issue by proposing a novel technique to take into account the approximation error from the circuit in the noise flooding technique. Thanks to this, by selecting the parameter of FHE accordingly, we finally prove the security of the proposed protocol.

Extending Functionality. In FPHE, the receiver receives the intersection result only. However, some real-world applications require more functionalities. For example, in image retrieval, the receiver needs to learn the corresponding label of the intersecting set elements. For this reason, we also propose variants for richer functionalities, including *labeled* fuzzy PSI and *circuit* fuzzy PSI. Note that since we utilize the sign function to determine whether two set elements are close enough or not, the resulting ciphertext contains only one of 0 or 1. Hence, by multiplying a label on the ciphertext, we can easily embed the label. In addition, by cleverly packing the plaintext during the protocol, we show that the arithmetic circuit, such as summation, can be evaluated by applying homomorphic operations on the ciphertext from the sign function evaluation. With these ideas, we successfully extend the functionality of FPHE without significant overhead.

1.2 Related Works

Traditional PSI. Efficient traditional PSI protocols have been proposed in many studies through various cryptographic primitives: oblivious transfer (OT) [46,45,5], oblivious pseudorandom function [53,50,8,14], vector oblivious linear evaluation (VOLE) [50,48], oblivious key-value store (OKVS) [7,28,48], and fully homomorphic encryption [16,15,20,52,56]. Additionally, there are PSI protocols that output related information for the intersection, such as the cardinality [24,51,57,56,58], the associated labels for each set element [15,20,8], or the evaluation of arbitrary circuit [50,52]. In particular, for FHE-based constructions. Chen et al. [16] proposed the first efficient PSI protocol, along with several optimization tricks. Based on their framework, there has been a series of improvements, including achieving malicious security [15], improving concrete efficiency [20], extending functionalities [15,52,56]. However, all these protocols are designed for exact matching, so they are not applicable in our target scenarios.

Fuzzy PSI. Fuzzy PSI was first introduced by Freedman et al.[25] and recently several fuzzy PSI protocols have been proposed for various distance metrics. For Hamming distance, Uzun et al. [54] proposed a HE-based construction, and Chakraborti et al. [13] improved it. However, their approaches suffer from non-negligible false-positive rates. For Euclidean distances, which is our main interest, Garimella et al. [29,30] proposed structure-aware PSI (sa-PSI) for L_∞ norm, where the receiver only holds a set A of disjoint L_∞ balls. They used weak boolean function secret sharing (bFSS) and spatial hashing technique, which is later improved at [27] to reduce the costs of prior works [29,30]. On the other hand, van Baarsen et al. [3] improved the idea of prior works [29,30] by employing the idea of Apple's PSI [6], relying on a weaker assumption. Additionally, they extended their idea to handle $L_{p\in[1,\infty)}$ distance. Gao et al. [26] proposed a fuzzy PSI for L_p distances based on a novel primitive called fuzzy mapping, which enables them to improve all prior works under a similar assumption. Although fuzzy PSI protocols for L_2 metric can be used for dealing with cosine similarity, as we mentioned, all of them are not applicable in our setting.

2 Preliminaries

Notations. For an integer d, $[d]$ denotes a set of integers $\{1, 2, \ldots, d\}$. We use bold font with uppercase letters to represent matrices, bold font with lowercase letters to represent vectors, and italic font to represent sets. For example, matrix $\mathbf{P}$, vector $\mathbf{c}$, and set I. For a vector $\mathbf{c}$, c_k denotes the k-th component of $\mathbf{c}$. For a matrix $\mathbf{P}$, we use $\mathbf{p}^{(i,\cdot)}$ as the i-th row and $\mathbf{p}^{(\cdot,j)}$ as the j-th column. $p^{(i,j)}$ denotes a component of the matrix $\mathbf{P}$ at position (i, j). For $\mathbf{P} \in \mathbb{R}^{d\times N}$ and $\mathbf{Q} \in \mathbb{R}^{d\times M}$, $\mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)}$ denotes the inner product operation between the i-th column of $\mathbf{P}$ and the j-th column of $\mathbf{Q}$, i.e., $\mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)} = \sum_{k=1}^{d} p^{(k,i)} q^{(k,j)}$. $S^{d-1} = \{\mathbf{c} \in \mathbb{R}^d \mid ||\mathbf{c}||_2 = 1\}$ denotes a unit $(d-1)$-sphere.

For any distribution D, $x \leftarrow D$ denotes sampling x from the distribution D, and it denotes the sampling from the uniform distribution over D when D is a finite set. For a real $\sigma > 0$, $N_{\mathbb{Z}}(0, \sigma^2)$ denotes a discrete Gaussian distribution with a mean of 0 and a variance of σ^2. $N_{\mathbb{Z}^n}(0, \sigma^2 \mathbf{I}_n)$ denotes a (discrete) multivariate normal distribution with zero mean vector and covariance matrix $\sigma^2 \mathbf{I}_n$, where $\mathbf{I}_n$ is an $n \times n$ identity matrix. For an integer q, we identify $\mathbb{Z} \cap (-q/2, q/2]$ as a representative of $\mathbb{Z}_q$. For an integer m, $\mathbb{Z}_m^* = \{x \in \mathbb{Z}_m \,|\, \gcd(x, m) = 1\}$. $\zeta = \exp(-2\pi i/m)$ denots a primitive m-th root of unity.

2.1 Fuzzy Private Set Intersection for Cosine Similarity

We now clarify our goal, fuzzy PSI protocol for cosine similarity, which is a two-party protocol between the receiver $\mathcal{R}$ and the sender $\mathcal{S}$. Suppose $\mathcal{R}$ and $\mathcal{S}$ have datasets $\mathbf{P} = \{\mathbf{p}^{(\cdot,1)}, \ldots, \mathbf{p}^{(\cdot,N)}\} \subseteq S^{d-1}$ and $\mathbf{Q} = \{\mathbf{q}^{(\cdot,1)}, \ldots, \mathbf{q}^{(\cdot,M)}\} \subseteq S^{d-1}$, respectively. For a threshold $T \in [-1, 1]$, the goal of the protocol is to let $\mathcal{R}$ learn indices $\{\mathrm{idx} \in [N] : \mathbf{p}^{(\cdot,\mathrm{idx})} \cdot \mathbf{q}^{(\cdot,j)} > T \text{ for some } j \in [M]\}$, without leaking any information to $\mathcal{S}$. We denote the corresponding ideal functionality as $\mathcal{F}_{\texttt{FPSI}}$, which is described in Fig. 1.

Ideal Functionality $\mathcal{F}_{\texttt{FPSI}}$
Parameters: threshold $T \in [-1, 1]$.
Inputs: $\mathcal{R}$ has input $\mathbf{P} = \{\mathbf{p}^{(\cdot,1)}, \ldots, \mathbf{p}^{(\cdot,N)}\} \subseteq S^{d-1}$. $\mathcal{S}$ has input $\mathbf{Q} = \{\mathbf{q}^{(\cdot,1)}, \ldots, \mathbf{q}^{(\cdot,M)}\} \subseteq S^{d-1}$.
Output: Returns an index set I to $\mathcal{R}$ such that for each $\mathrm{idx} \in [N]$, $\mathrm{idx} \in I$ if there exists $j \in [M]$ such that $\mathbf{p}^{(\cdot,\mathrm{idx})} \cdot \mathbf{q}^{(\cdot,j)} > T$; otherwise, $\mathrm{idx} \notin I$. $\mathcal{S}$ receives nothing.

Fig. 1. Ideal Functionality $\mathcal{F}_{\texttt{FPSI}}$.

Security Model. Throughout this paper, we consider a semi-honest adversary [11], *i.e.*, each party honestly follows the protocol, but one of them can be corrupted. Later, we show that the proposed protocol $\pi_{\texttt{FPHE}}$ securely implements the ideal functionality $\mathcal{F}_{\texttt{FPSI}}$ under the semi-honest model: see Theorem 2 for details. We provide a formal definition in Appendix A.

2.2 CKKS

We use CKKS [17], an approximate FHE to perform operations with encrypted real-valued data. As approximate FHE, each ciphertext contains some noise, which grows with operations. The scheme begins by selecting the following parameters: an initial modulus q_0, a scaling factor $\Delta > 0$, a ring dimension n, and the maximum level $L \in \mathbb{N}$. For each level l, the modulus is defined as $q_l = \Delta^l \cdot q_0$ for $0 \leq l \leq L$. Then, it operates on the following spaces: plaintext space $\mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1\rangle$, and ciphertext space $\mathcal{R}_{q_l} = \mathbb{Z}_{q_l}[X]/\langle X^n + 1\rangle$. Here, n is a power of two, and $X^n + 1$ is a m-th cyclotomic polynomial, where $m = 2n$.

A polynomial $m(X) \in \mathcal{R}$ can be embedded into $\mathbb{C}^{n/2}$ through the canonical embedding $\delta(m)$. It consists of an evaluation of $m(X)$ at the roots of $X^n + 1$, *i.e.*, ζ^k for a primitive m-th root of unity ζ. Since $m(\zeta^k) = m(\overline{\zeta^{-k}})$, these n evaluations can be represented as a vector in $\mathbb{C}^{n/2}$. Then, a message $\mathbf{m}$ is encoded to $\lfloor \Delta\delta^{-1}(\mathbf{m}) \rceil \in \mathcal{R}$, and $m(X)$ is decoded to $\Delta^{-1}\delta(m(X))$.

Below, we provide a brief description of the scheme, with details available in [17]. Note that a polynomial is sampled from a distribution D by independently sampling each of its coefficients from D.

- $\mathsf{KeyGen}(1^\lambda) \to (pk, evk, sk)$: Given the security parameter λ, set parameters involving m and σ, and output a public key pk, an evaluation key evk, and a secret key sk.
- $\mathsf{Enc}_{pk}(m(X)) \to \mathbf{c} \in \mathcal{R}_{q_L}^2$: Output a ciphertext $\mathbf{c}$ of plaintext $m(X) \in \mathcal{R}$.
- $\mathsf{Dec}_{sk}(\mathbf{c}) \to m'(X)$: For a ciphertext $\mathbf{c}$ of a plaintext $m(X)$, output $m'(X) = m(X) + e(X)$ with an error $e(X)$.
- $\mathsf{Add}(\mathbf{c}_1, \mathbf{c}_2) \to \mathbf{c}_{\mathsf{add}}$: For ciphertexts of $m_1(X)$ and $m_2(X)$, output a ciphertext of $m_1(X) + m_2(X)$, where it has an error that is bounded by the sum of two errors for the inputs.
- $\mathsf{Mult}_{evk}(\mathbf{c}_1, \mathbf{c}_2) \to \mathbf{c}_{\mathsf{mult}}$: For ciphertexts of $m_1(X)$ and $m_2(X)$, output a ciphertext $\mathbf{c}_{\mathsf{mult}}$, where its decryption is $(m_1(X)+e_1(X))(m_2(X)+e_2(X)) + e_{\mathsf{mult}}$ for some polynomial $e_{\mathsf{mult}} \in \mathcal{R}$.
- $\mathsf{RS}_{l\to l'}(\mathbf{c}) \to \mathbf{c}' \in \mathcal{R}_{q_{l'}}^2$: For $\mathbf{c} \in \mathcal{R}_{q_l}^2$, output $\mathbf{c}' \leftarrow \lfloor (q_{l'}/q_l) \cdot \mathbf{c} \rceil \pmod{q_{l'}}$.

The CKKS scheme satisfies IND-CPA security under the hardness assumption of the RLWE problem. Decisional RLWE problem is defined as the computational indistinguishability between the tuple $(as + e, a) \in \mathcal{R}_q^2$, where $s \leftarrow \chi_{\mathsf{Ham}}$ and $e \leftarrow N_{\mathbb{Z}}(0, \sigma^2)$, and a uniformly random tuple $(b, a) \leftarrow \mathcal{R}_q^2$.

2.3 Sign function

The sign function is one of the fundamental operations, especially for implementing the comparison operator. Due to its discontinuity, homomorphically computing a sign function has been a non-trivial problem, and several methods [19,18,36,42,35] have been proposed to efficiently compute its approximation. Note that we also need this for determining whether two set elements are sufficiently close or not. First, we define the comparison function and the sign function as follows.

Definition 1 ([18]). *A comparison function comp* : $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *and a sign function sgn* : $\mathbb{R} \to \mathbb{R}$ *are defined as*

$$comp(X,Y) = \begin{cases} 1 & \text{if } X > Y, \\ \frac{1}{2} & \text{if } X = Y, \\ 0 & \text{if } X < Y \end{cases} \quad , \quad sgn(X) = \begin{cases} 1 & \text{if } X > 0, \\ 0 & \text{if } X = 0, \\ -1 & \text{if } X < 0. \end{cases}$$

Note that $comp(X,Y) = \frac{sgn(X-Y)+1}{2}$.

For an approximated polynomial $p(X)$ of $sgn(X)$, we characterize its precision by defining the notion of (α, ϵ)-close. Each parameter describes how $p(x)$ is *close* to sgn and how much $p(x)$ can be faithfully approximated near 0, respectively. The formal definition is given as follows:

Definition 2 ([18]). *For $\alpha > 0$ and $0 < \epsilon < 1$, a polynomial $p(X)$ is said to be (α, ϵ)-close to $sgn(X)$ over $[-1, 1]$ if $p(X)$ satisfies the following:*

$$\|p(X) - sgn(X)\|_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} \leq 2^{-\alpha},$$

where $||\cdot||_{\infty,R}$ denotes the infinity norm over the domain R.

In this paper, we call $|p(x) - sgn(x)|$ an *accuracy error* for $x \in [-1, -\epsilon] \cup [\epsilon, 1]$.

2.4 Noise Flooding

In our construction, the receiver finally obtains a ciphertext corresponding to an output of homomorphic evaluation of some circuit. However, it is known that such a ciphertext in CKKS could leak information about non-public inputs of the circuit [38], which is the dataset of the sender in our case. Hence, we employ noise flooding to prevent this issue.

The effect of noise flooding in CKKS was thoroughly studied by Li et al. [39]. For the sake of a rigorous security proof in our protocol, we introduce some results of theirs. Their analysis was based on the previous result [38] that the leakage of non-public inputs stems from the decryption error, which is defined as follows:

Definition 3 ([39]). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add}, \mathsf{Mult}, \mathsf{RS})$ be a (approximation) FHE scheme with plaintext space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, which is a normed space with norm $\|\cdot\| : \widetilde{\mathcal{M}} \to \mathbb{R}_{\geq 0}$. For any ciphertext $\mathbf{c}$, plaintext m, and secret key sk, the ciphertext error of $(\mathbf{c}, m, sk)$ is defined to be*

$$\mathsf{Error}(\mathbf{c}, m, sk) = \|\mathsf{Dec}_{sk}(\mathbf{c}) - m\|.$$

For CKKS, canonical embedding norm is used, which is defined by for $a(X) \in \mathbb{Q}[X]/\langle X^n + 1\rangle$, $\|a(X)\|_\infty^{\mathsf{can}} = \|(a(\zeta^k))_{k\in\mathbb{Z}_m^*}\|_\infty$.

Suppose $\mathbf{c}_1, \ldots, \mathbf{c}_k$ are encryptions of plaintexts $m_1, \ldots, m_k$, respectively. Given an arithmetic circuit $\mathcal{G}$, let $\mathbf{c}$ denote a ciphertext computed on $\mathbf{c}_1, \ldots, \mathbf{c}_k$ for $\mathcal{G}$, and let m be an output of $\mathcal{G}(m_1, \ldots, m_k)$. In this case, we call the ciphertext error $\mathsf{Error}(\mathbf{c}, m, sk)$ a *circuit error*. Li et al. [36] proved that one can blend the non-trivial information from the circuit error via a similar scale of the noise, hence achieving the desirable goal. More precisely, one can achieve q-IND-CPA^D security with this modification, which means that the adversary cannot obtain any information about the non-public input with up to q queries to a decryption oracle. We provide the full statement as follows:

Definition 4 ([39]). *Let ct be a tuple $(\mathsf{ct}.c, \mathsf{ct}.t)$ for CKKS scheme, where $\mathsf{ct}.c$ is a ciphertext, and $\mathsf{ct}.t$ is an estimation of the worst error bound for $\mathsf{ct}.c$. Then, for $\sigma^* > 0$, S-CKKS$_{\sigma^*}$ is CKKS scheme which modifies decryption to compute S-CKKS$_{\sigma^*}.\mathsf{Dec}_{sk}(\mathsf{ct}) = \mathsf{Dec}_{sk}(\mathsf{ct}.c) + e$ for $e \leftarrow N_{\mathbb{Z}^n}(0, {\sigma^*}^2 \mathsf{ct}.t^2 \mathbf{I}_n)$.*

Theorem 1 (Corollary 2 of [39]). *For any $q, n \in \mathbb{N}$, let q be a maximum of decryption queries an attacker can make, and let n be the ring dimension. Then, if CKKS is $(c + \log_2 24)$-bit IND-CPA-secure, and $\sigma^* = \sqrt{24qn}2^{s/2}$, then S-CKKS$_{\sigma^*}$ is (c, s)-bit q-IND-CPA^D-secure.*

Here, s is about statistical security, *i.e.*, the adversary can gain any non-trivial information from a decrypted result with probability 2^{-s}. We also note that the worst error bound can be estimated according to [17].

3 Fuzzy PSI Protocol for Cosine Similarity

We now present our Fuzzy PSI protocol for cosine similarity, denoted by $\pi_{\texttt{FPHE}}$, which is specialized for dealing with high-dimensional data. $\pi_{\texttt{FPHE}}$ requires weaker assumption than those for prior works for L_2 norm [3,26], which turn out to be impractical for cosine similarity. We also prove its security in a semi-honest model and extend its functionalities, including *labeled* and *circuit* fuzzy PSI.

3.1 Limitation of Assumptions in Previous Proposals

The cosine similarity and L_2 distance are equivalent when the given two vectors are unit. More precisely, if the cosine similarity of two vectors $\mathbf{x}, \mathbf{y}$ is c, then $\|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{2 - 2c}$. Hence, fuzzy PSI protocols [3,26] supporting L_2 distance would be employed in our target applications. However, we point out that their assumptions would not be achieved for unit vectors in practice.

For a precise explanation, we first focus on the assumption used in van Baarsen et al. [3], which assumes that all elements should be far from each other, more than $2T_2(d^{1/2} + 1)$ in L_2 distance, where T_2 is the threshold in L_2 distance. That is, all these elements should not overlap each other when we consider L_∞ balls of radius T centered at these set elements. However, note that the upper bound of L_∞ distance between two unit vectors is 2. That is, this restricts the choice of the possible threshold T_2 by $T_2 < \frac{1}{\sqrt{d}+1} < \frac{1}{\sqrt{d+1}}$, *i.e.*, from the perspective of cosine similarity, $T > \frac{2d+1}{2d+2}$. For example, if $d = 512$, then the lower bound becomes ≈ 0.999, which is extremely tighter than the threshold of usual face authentication models [22,34] that uses $T \approx 0.3$.

On the other hand, Guo et al. [26] used the disjoint projection assumption, which claims that there exists an axis such that all the neighborhoods of set elements are non-overlapping when projected to this axis. We can observe that this condition is implied by that used in van Baarsen et al. Hence, to validate this assumption as well, their choice of T subjects to the lower bound in above. Therefore, they are not applicable in practice, and an alternative approach tailored for cosine similarity is needed.

3.2 Construction of $\pi_{\texttt{FPHE}}$

At a high level, our protocol consists of the following four steps. We will denote the receiver and the sender as $\mathcal{R}$ and $\mathcal{S}$, respectively.

1. $\mathcal{R}$ encrypts its data via CKKS and sends it to $\mathcal{S}$.
2. $\mathcal{S}$ homomorphically computes the cosine similarity between each data.
3. $\mathcal{S}$ homomorphically evaluates a comparison function with the threshold.
4. $\mathcal{S}$ applies a noise flooding technique at the result and sends it to $\mathcal{R}$.

In the remaining of this subsection, we will provide details about each step.

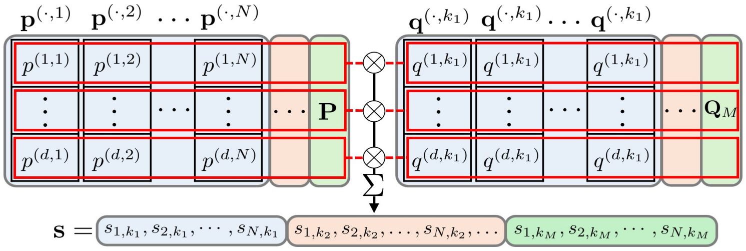

Fig. 2. The description of our approach to compute cosine similarity scores. $\otimes$ and Σ denote component-wise multiplication and summation, respectively.

Settings and Data Format. We denote $\mathbf{P} = \{\mathbf{p}^{(\cdot,1)}, \ldots, \mathbf{p}^{(\cdot,N)}\} \subseteq S^{d-1}$ and $\mathbf{Q} = \{\mathbf{q}^{(\cdot,1)} \ldots, \mathbf{q}^{(\cdot,N)}\} \subseteq S^{d-1}$ as datasets held by $\mathcal{R}$ and $\mathcal{S}$, respectively. For the ease of explanation, we assume that $MN = n/2$, *i.e.*, the number of packed components in one CKKS ciphertext. For the case when $MN > n/2$, we run the protocol several times. We assume that the dataset $\mathbf{Q}$ held by $\mathcal{S}$ is *non-overlapping*, *i.e.*, no pair of elements in $\mathbf{Q}$ are closer to each other than the threshold T. This is reasonable in practice because $\mathcal{S}$ may pre-process to delete overlapping data before running the protocol. Note that in this setting, for each set element in $\mathbf{P}$ there is at most one element in $\mathbf{Q}$ within the threshold. Finally, we assume that $\mathbf{P}$ and $\mathbf{Q}$ are well-quantized so that for sufficiently small $\epsilon > 0$, the inner product between each pair of set elements does not lie within the range $(T - 2\epsilon, T + 2\epsilon)$. Note that this is not a harsh assumption because several quantization methods, *e.g.*, 8-bit or 16-bit, showed that such a quantization would not give a significant effect on the inner product value [4,23]. Throughout this paper, we use $\epsilon = 2^{-16}$.

Computation of Similarity Score. Each dataset can be represented as matrices, *i.e.*, $\mathbf{P} \in \mathbb{R}^{d \times N}$ and $\mathbf{Q} \in \mathbb{R}^{d \times M}$, respectively. In this step, the goal is to homomorphically compute $\mathbf{P}^T \cdot \mathbf{Q} \in \mathbb{R}^{N \times M}$. To this end, our key idea is to cleverly encode $\mathbf{P}$ to take advantage of homomorphic operations between plaintext and ciphertext, which is much cheaper than ciphertexts. This process is described in Figure 2. $\mathcal{R}$ sets $\overline{\mathbf{P}}$ by concatenating M copies of $\mathbf{P}$. For $j \in [M]$, $\mathcal{S}$ creates $\overline{\mathbf{Q}}_j$, each containing N copies of $\mathbf{q}^{(\cdot,j)}$. The final $\overline{\mathbf{Q}}$ is generated by uniformly permuting $\overline{\mathbf{Q}}_j$'s and concatenating them. The plaintext-ciphertext multiplication between the i-th rows of $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ corresponds to element-wise multiplication. Summing the results obtained from only d multiplications generates a single ciphertext that contains all scores $s_{i,k_j} = \mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,k_j)}$ between $\mathbf{P}$ and $\mathbf{Q}$. This single ciphertext allows the desired result to be computed with a single evaluation of the comparison function.

Evaluating Comparison Function. After computing the cosine similarity scores, $\mathcal{S}$ runs a comparison circuit with respect to the threshold T, *i.e.*, $\frac{sgn(\mathbf{c}-T)+1}{2}$. To this end, $\mathcal{S}$ utilizes a polynomial $p(X)$ that is $(\alpha - 1, \epsilon)$-close to the sign function sgn over the interval $[-1, 1]$. Here, one caveat is that the value of $\mathbf{c} - T$ lies within $[-2, 2]$, so we need to (homomorphically) multiply $\frac{1}{2}$ to fit the desired range of $[-1, 1]$. After evaluation, we can ensure that the resulting ciphertext $\mathbf{s}$ holds a message that lies within $[1 - 2^{-\alpha}, 1 + 2^{-\alpha}]$ if there is an overlap between set elements, otherwise lies within $[-2^{-\alpha}, 2^{-\alpha}]$.

Noise Flooding Technique. To ensure the privacy of $\mathcal{S}$'s dataset, $\mathcal{S}$ finally applies the noise flooding technique on $\mathbf{s}$. Recall that the Theorem 1 tells us the precise amount of the noise for whitening the information leakage from the circuit error. However, in our setting, this theorem *as is* would not ensure the security of our protocol because of the approximation error on evaluating *sgn*. Hence, instead of adding the noise according to the worst-case circuit error $\mathsf{ct}.t$, we need to use an adjusted value $\overline{\mathsf{ct}.t}$ that also considers the approximation error.

We also note that the decrypted values are also affected by the noise flooding. For this reason, $\mathcal{R}$ employs a parameter r to determine the overlapping indices from the received ciphertext $\mathbf{s}$. More precisely, for the message $\mathbf{z}$ corresponding to $\mathbf{s}$, $\mathcal{R}$ checks whether each component of $\mathbf{z}$ lies within $[1 - 2^r, 1 + 2^r]$ or $[-2^r, 2^r]$. The precise analysis on the desired $\overline{\mathsf{ct}.t}$ and r will be provided in the next subsection, along with the security proof of our protocol.

The Full Protocol. By merging these building blocks all together, we finally construct our Fuzzy PSI protocol $\pi_{\mathtt{FPHE}}$. For completeness, we describe the overall protocol with details in Fig. 3.

Computation/Communication Costs. We now analyze the computation and communication cost of $\pi_{\mathtt{FPHE}}$ for each party. We first consider the case when $MN = n/2$, and extend it to the general case. During the protocol, the communication occurs when sending and retrieving ciphertexts, *i.e.*, d ciphertexts for encrypting $\overline{\mathbf{P}}$ and a ciphertext corresponding to $\mathbf{s}$. That is, the communication cost is $(d + 1)$ ciphertexts. For computation, we first observe that $\mathcal{R}$ proceeds with d encryption and 1 decryption during the protocol. On the other hand, $\mathcal{S}$ takes d plaintext-ciphertext multiplications and 1 evaluation of the sign function.

We now consider the case $MN > n/2$. If we assume $N < n/2$, then we can chunk the dataset $\mathbf{Q}$ of $\mathcal{S}$ into $k = \lceil \frac{2MN}{n} \rceil$ pieces, say $\mathbf{Q}_1, \ldots, \mathbf{Q}_k$, where each piece has elements at most $\frac{n}{2N}$. Hence, it suffices to run k runs of $\pi_{\mathtt{FPHE}}$ with inputs $(\mathbf{P}, \mathbf{Q}_1), \ldots, (\mathbf{P}, \mathbf{Q}_k)$ for $\mathcal{R}$ and $\mathcal{S}$, respectively. Here, we can observe that (1) for each protocol, $\mathcal{R}$ sends the same ciphertexts to $\mathcal{S}$, and (2) the evaluation result of the sign function $\mathbf{s}_i$ for each run of the protocol on inputs $(\mathbf{P}, \mathbf{Q}_i)$ can be aggregated by summation. Hence, the computation cost of $\mathcal{R}$ and the communication cost remain the same as $\pi_{\mathtt{FPHE}}$, whereas the computation of $\mathcal{S}$ is scaled by k.

3.3 Security Analysis

In this subsection, we show that the proposed $\pi_{\mathtt{FPHE}}$ indeed achieves the semi-honest security, *i.e.*, securely implements the ideal functionality $\mathcal{F}_{\mathtt{FPSI}}$ in Fig. 1. The core part of our security proof is to analyze the scale $\overline{\mathsf{ct}.t}$ of noise flooding to ensure that the retrieved ciphertext $\mathbf{s}$ did not contain any useful information. For this reason, we will focus on providing our analysis about $\overline{\mathsf{ct}.t}$.

Parameters: threshold $T \in [-1, 1]$, and security parameters λ and r.
Inputs: $\mathcal{R}$ has input $\mathbf{P} = \{\mathbf{p}^{(\cdot,1)}, \dots, \mathbf{p}^{(\cdot,N)}\} \subseteq S^{d-1}$. $\mathcal{S}$ has input $\mathbf{Q} = \{\mathbf{q}^{(\cdot,1)}, \dots, \mathbf{q}^{(\cdot,M)}\} \subseteq S^{d-1}$. i.e., $\mathbf{P} \in \mathbb{R}^{d\times N}$ and $\mathbf{Q} \in \mathbb{R}^{d\times M}$.
Procedure:

1. $\mathcal{R}$ runs $(pk, evk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $\mathcal{R}$ sets a matrix $\overline{\mathbf{P}} = \mathbf{P}\|\cdots\|\mathbf{P} \in \mathbb{R}^{d\times n/2}$ by concatenating M copies of $\mathbf{P}$.
3. For each $i \in [d]$, $\mathcal{R}$ encodes $\overline{\mathbf{p}}^{(i,\cdot)}$ to a polynomial $\overline{p}^{(i,\cdot)}(X) \in \mathcal{R}$ and runs $\mathbf{p}_i \leftarrow \mathsf{Enc}_{pk}(\overline{p}^{(i,\cdot)}(X))$.
4. $\mathcal{R}$ sends pk, evk, and $\{\mathbf{p}_i\}_{i\in[d]}$ to $\mathcal{S}$.
5. For $j \in [M]$, $\mathcal{S}$ creates $\overline{\mathbf{Q}}_j \in \mathbb{R}^{d\times N}$ by concatenating N copies of $\mathbf{q}^{(\cdot,j)}$, then sets $\overline{\mathbf{Q}} = \overline{\mathbf{Q}}_{k_1}\|\cdots\|\overline{\mathbf{Q}}_{k_M} \in \mathbb{R}^{d\times n/2}$ through random permutation.
6. For each $i \in [d]$, $\mathcal{S}$ encodes $\overline{\mathbf{q}}^{(i,\cdot)}$ to a polynomial $\overline{q}^{(i,\cdot)}(X) \in \mathcal{R}$.
7. For each $i \in [d]$, $\mathcal{S}$ computes $\mathbf{v}_i = \overline{q}^{(i,\cdot)}(X) \cdot \mathbf{p}_i$ and $\overline{\mathbf{v}}_i \leftarrow \mathsf{RS}(\mathbf{v}_i)$.
8. $\mathcal{S}$ computes $\mathbf{c}$ by adding $\overline{\mathbf{v}}_i$ for all $i \in [d]$ using the Add algorithm.
9. $\mathcal{S}$ computes $\overline{T} \leftarrow \lfloor \Delta T \rceil \in \mathcal{R}$ and $\mathbf{w} = \mathbf{c} - (\overline{T}, 0)$.
10. $\mathcal{S}$ computes $\mathbf{s}$ for the sign function sgn with $\mathbf{w}$ by using the Add, Mult and RS algorithms.
11. $\mathcal{S}$ samples $e \leftarrow N_{\mathbb{Z}^n}(0, \sigma^{*2}\overline{\mathsf{ct}.t}^2\mathbf{I_n})$ and computes $\overline{\mathbf{z}} = \mathbf{s} + (e, 0)$.
12. $\mathcal{S}$ sends $\overline{\mathbf{z}}$ to $\mathcal{R}$.
13. $\mathcal{R}$ runs $z(X) \leftarrow \mathsf{Dec}_{sk}(\overline{\mathbf{z}})$ and decodes $z(X)$ to obtain a message $\mathbf{z}$.
14. For $\mathbf{z} = (z_1, \dots, z_{n/2})$, $\mathcal{R}$ sets an index set $I \subset [N]$ satisfying $\mathrm{idx} \in I$ if and only if $\exists j \in [M]$ such that $z_{(j-1)N+\mathrm{idx}} \in [1 - 2^r, 1 + 2^r]$

Fig. 3. The description of the proposed Fuzzy PSI Protocol $\pi_{\mathtt{FPHE}}$.

Deriving $\overline{\mathsf{ct}.t}$ for Noise Flooding. We first recall that the rationale of employing the noise flooding is to cover the circuit error via a large noise. For a precise description, let us denote $x_0 = \mathsf{Dec}_{sk}(\mathsf{ct}.c) = m(X) + e_c(X)$ as a decryption result with a circuit error $e_c(X)$ and $x_1 = m(X)$. Then the effect of the noise flooding can be viewed as adding $e^*(X)$ on both x_0 and x_1, ensuring that $e^*(X)$ and $e_c(X) + e^*(X)$ are statistically indistinguishable. The Theorem 1 tells us the scale of noise when $\|e_c(X)\|_\infty^{\mathsf{can}} \le \mathsf{ct}.t$.

On the other hand, in our setting, one more thing needs to be considered: if we decrypt $\mathbf{s}$, say $s(X) = \mathsf{Dec}_{sk}(\mathbf{s})$, then $s(X)$ contains not only the circuit error but also the accuracy error. More precisely, if we denote acc_i as the accuracy error of the i'th component, *i.e.*, the difference between the evaluation result of $p(X)$ and the actual comparison function, then we have that $\mathsf{acc}_i \in [-2^{-\alpha}, 2^{-\alpha}]$. Note that $\mathsf{acc}_i \in \mathbb{R}$, so in the context of plaintexts, we need to consider an encoded polynomial $e_{\mathsf{acc}}(X) = \lfloor \Delta\delta^{-1}((\mathsf{acc}_i)_{i\in[N]}) \rceil$. This can be interpreted as an additional error added to the circuit error $e_c(X)$. That is, by using the same argument as above and Theorem 1, it suffices to compute the upper bound of $\|e_c(X) + e_{\mathsf{acc}}(X)\|_\infty^{\mathsf{can}}$ for applying the noise flooding. Note that by the property of canonical norm and the triangular inequality, we obtain that

$$||e_c(X) + e_{\mathsf{acc}}(X)||_\infty^{\mathsf{can}} \le ||e_c(X)||_\infty^{\mathsf{can}} + ||e_{\mathsf{acc}}(X)||_\infty^{\mathsf{can}} \le \mathsf{ct}.t + \Delta 2^{-\alpha}.$$

Therefore, for the scale term $\overline{\mathsf{ct}.t} = \mathsf{ct}.t + \varDelta 2^{-\alpha}$, we finally can exploit the result of Theorem 1 by replacing $\mathsf{ct}.t$ with $\overline{\mathsf{ct}.t}$.

Parameter Selection. To ensure the correctness of our protocol $\pi_{\mathtt{FPHE}}$, $\mathcal{R}$ should be able to distinguish whether the retrieved ciphertext from $\mathcal{S}$ with noise flooding contains 0 or 1. That is, two intervals $[1-2^r, 1+2^r]$ and $[-2^r, 2^r]$ in the previous section should be non-overlapping. In this paragraph, we show that this is possible by a careful parameter selection of CKKS. Therefore, combining with Theorem 1, we can conclude that $\pi_{\mathtt{FPHE}}$ securely implements the ideal functionality $\mathcal{F}_{\mathtt{FPSI}}$.

To address this, let us assume that the noise $e^*(X)$ from $N(0, {\sigma^*}^2\overline{\mathsf{ct}.t}^2\mathbf{I}_n)$ is added to $\mathbf{s}$, in accordance with Theorem 1. Note that $\sigma^* = \sqrt{24qn}2^{s/2}$ for parameters $q, s \in \mathbb{N}$, which will be chosen later. Then we can observe that the total error compared to the result from the actual comparison function becomes

$$\|e_c(X) + e_{acc}(X) + e^*(X)\|_\infty^{\mathsf{can}} \leq \overline{\mathsf{ct}.t} + \|e^*(X)\|_\infty^{\mathsf{can}}.$$

Hence, the error corresponding to each message is at most $\frac{\overline{\mathsf{ct}.t} + \|e^*(X)\|_\infty^{\mathsf{can}}}{\varDelta}$.

To analyze this, we attempt to put $\beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}$ in place of $\|e^*(X)\|_\infty^{\mathsf{can}}$, where β is a parameter determined later. Then we obtain

$$\frac{\overline{\mathsf{ct}.t} + \beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}}{\varDelta} = (\beta\sqrt{n}\sigma^* + 1)\left(\frac{\mathsf{ct}.t}{\varDelta} + 2^{-\alpha}\right) = (1 + \beta n\sqrt{24q}2^{s/2})\left(\frac{\mathsf{ct}.t}{\varDelta} + 2^{-\alpha}\right).$$

If we choose $q = 2^5$, $s = 40$, and $n = 2^{17}$, then the R.H.S. of the above equality becomes $\approx \beta \cdot 2^{41.79}(\frac{\mathsf{ct}.t}{\varDelta} + 2^{-\alpha})$.

Here, note that the random variable $e^*(\zeta^k)$ for $k \in \mathbb{Z}_m^*$ follows $N_{\mathbb{Z}}(0, n{\sigma^*}^2\overline{\mathsf{ct}.t}^2)$, since it is a linear combination of n independent samples of $N_{\mathbb{Z}}(0, {\sigma^*}^2\overline{\mathsf{ct}.t}^2)$ with weights $\{\zeta^{k(i-1)}\}_{i=1}^n$. Since all random variables $\{e^*(\zeta^k)\}_{k\in\mathbb{Z}_{m^*}}$ are mutually independent because of its structure, we obtain

$$\Pr[\|e^*(X)\|_\infty^{\mathsf{can}} > \beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}] = 1 - \left(1 - \Pr[|e^*(\zeta^k)| > \beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}]\right)^n < np$$

for any $k \in \mathbb{Z}_m^*$, where $p = \Pr[|e^*(\zeta^k)| > \beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}]$. The last inequality holds because $(1-p)^n \geq 1 - np$. Finally, by using the fact that $\Pr_{Z\sim N_{\mathbb{Z}(0,1)}}[|Z| > \lambda] \leq e^{-\frac{\lambda^2}{2}}$ [12], we obtain $\Pr[\|e^*(X)\|_\infty^{\mathsf{can}} > \beta\sqrt{n}\sigma^*\overline{\mathsf{ct}.t}] < n \cdot e^{-\frac{\beta^2}{2}} < n \cdot 2^{-\frac{\beta^2}{2}}$.

If we select $\beta = 16 = 2^4$, then the above bound becomes 2^{-111}. Therefore, by setting $\varDelta$ and α to satisfy $\varDelta \geq 2^4 \cdot 2^{41.79}\mathsf{ct}.t > \alpha$, we can expect that $\mathcal{R}$ successfully achieves its goal with an error of probability 2^{-111}. Note that q means the number of possible queries for q-IND-CPA^D, which becomes the number of possible runs of our protocol without regenerating keys. That is, after 2^5 runs of $\pi_{\mathtt{FPHE}}$ without changing the key, both $\mathcal{R}$ and $\mathcal{S}$ regenerate the key pair.

Under the above parameter setting, we finally show the semi-honest security of $\pi_{\mathtt{FPHE}}$ protocol, as stated in Theorem 2. The full proof is given in Appendix B.

Theorem 2. *The protocol $\pi_{\mathtt{FPHE}}$ described in Figure 3 securely implements the ideal functionality $\mathcal{F}_{\mathtt{FPSI}}$ under the static semi-honest model.*

3.4 Extending Functionality of π_{PFHE}

As we can figure out in Fig. 1, the π_{FPHE} is only designed for the scenario where the receiver is sufficient to learn whether the input elements are overlapping with some other elements of the sender or not. However, this functionality is insufficient for covering other practical applications, for example, where the receiver needs to know the label of the intersecting set elements. For this reason, we now focus on extending the functionality of π_{FPHE}.

Labeled and Circuit Fuzzy PSI. For the candidate of extended functionalities, we consider the labeled fuzzy PSI and circuit fuzzy PSI, which are analogues of labeled PSI [15,20,8] and circuit PSI [50,52] for traditional PSIs. We briefly introduce their functionalities. For the labeled fuzzy PSI, we assume that there is a label $l_i \in \mathcal{L} \subset \mathbb{N}$ assigned to each set element $\mathbf{q}^{(\cdot,i)}$ for $i \in [M]$ held by the sender $\mathcal{S}$. We assume that these labels are represented by w-bit (unsigned) integers, so for all $l \in \mathcal{L}$, $l < 2^w$ holds. From this setting, the receiver $\mathcal{R}$ wishes to learn a pair of an index $i \in [N]$ and a label l_j such that $\mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)} > T$. For indices of non-overlapping set elements, $\mathcal{R}$ gets $0 \in \mathbb{Z}$. On the other hand, in circuit fuzzy PSI, the $\mathcal{R}$ wishes to learn the evaluation result of an arbitrary function f that takes N integers $v_1, \ldots, v_N$ such that $v_i = |\{j \in [M] : \mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)} > T\}|$ as inputs. Note that under the non-overlapping assumption, $v_i \in \{0, 1\}$.

We note that achieving labeled fuzzy PSI is straightforward in our construction. This is because we can easily embed labels in the output of the comparison function $\mathbf{s}$ by a single plaintext-ciphertext multiplication, with a careful ordering about the index permutation for computing cosine similarity. The security proof also remains the same, except for a simple treatment on analyzing an additional noise accompanied by the label.

Leveraging the Packing Structure for Circuit Fuzzy PSI. Achieving circuit fuzzy PSI seems to be rather tricky compared to labeled fuzzy PSI. Nevertheless, we show that our π_{FPHE} can be easily extended to achieve this functionality by utilizing the structure of packed messages. For the ease of explanation, we assume that the indices of data held by $\mathcal{S}$ are not permuted. We first note that in our protocol, we set $MN = n/2$. If we denote $c_{i,j} = comp(\mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)}, T)$, then with ignoring errors, the final ciphertext $\mathbf{s}$ before noise flooding contains a decoded message $\mathbf{z}$ denoted by

$$\mathbf{z} = (c_{1,1}, \ldots, c_{N,1}, c_{1,2}\ldots, c_{M,2}, \ldots, c_{1,M}, \ldots, c_{N,M}) \in \mathbb{R}^{MN}.$$

Here, we can observe that the subvector $\mathbf{z}_j := (\mathbf{z}[i+jN])_{j=0}^{M-1}$ of stride N, where $\mathbf{z}[k]$ denotes the k'th component of $\mathbf{z}$, corresponds to $(c_{i,1}, \ldots, c_{i,N})$, *i.e.*, values with the same first index i.

We now explain why this packing structure enables us to achieve the circuit Fuzzy PSI. Because of the structure, we can observe that when rotating $\mathbf{z}$ by N on the left, then the corresponding $\mathbf{z}_j$ becomes $(c_{i,2}, \ldots, c_{i,M}, c_{i,1})$, *i.e.*, rotation on the original subvector $(c_{i,1}, \ldots, c_{i,N})$ by 1 on the left. Therefore, by a folklore

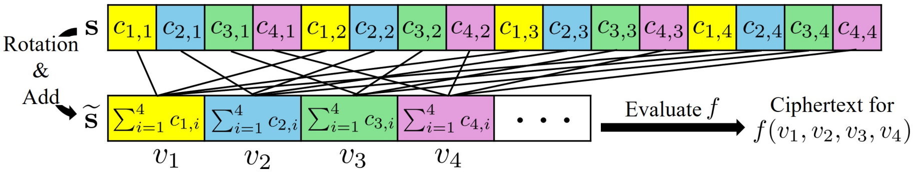
Rotation
&
Add
$\mathbf{s}$
$c_{1,1}$ $c_{2,1}$ $c_{3,1}$ $c_{4,1}$ $c_{1,2}$ $c_{2,2}$ $c_{3,2}$ $c_{4,2}$ $c_{1,3}$ $c_{2,3}$ $c_{3,3}$ $c_{4,3}$ $c_{1,4}$ $c_{2,4}$ $c_{3,4}$ $c_{4,4}$
$\widetilde{\mathbf{s}}$
$\sum_{i=1}^{4} c_{1,i}$ $\sum_{i=1}^{4} c_{2,i}$ $\sum_{i=1}^{4} c_{3,i}$ $\sum_{i=1}^{4} c_{4,i}$ • • •
v_1 v_2 v_3 v_4
Evaluate f
Ciphertext for
$f(v_1, v_2, v_3, v_4)$

Fig. 4. The description of our approach when $N = M = 4$. Best viewed in color.

rotation-and-add technique, we can compute the ciphertext $\widetilde{\mathbf{s}}$ containing M concatenations of $(\sum_{j=1}^{M} c_{1,j}, \sum_{j=1}^{M} c_{2,j}, \ldots, \sum_{i=1}^{M} c_{N,j})$. Here, note that this result is regardless of the permutation on the indices of $\{\mathbf{q}^{(\cdot,j)}\}_{j\in[M]}$ because the summation is done over $[M]$. Since $\sum_{j=1}^{M} c_{i,j} = |\{j \in [M] : \mathbf{p}^{(\cdot,i)} \cdot \mathbf{q}^{(\cdot,j)} > T\}| = v_i$ holds by definition, we finally implement the desired functionality by homomorphically evaluating the circuit f on $\widetilde{\mathbf{s}}$. To help understand, we depict an overview of our idea with the structure of $\mathbf{z}$ in Fig. 4.

The security proof goes similarly to Theorem 2. The only difference comes from calculating the scale of noise flooding. To this end, we need to consider the error appeared by the rotation-and-add technique and evaluating f, tracking how much the accuracy error acc is magnified during these operations. Since the remaining part of the security proof is straightforward, we omit the proof.

4 Implementation Results

We now provide the implementation result of $\pi_{\mathtt{FPHE}}$. We used the official Python wrapper of OpenFHE [1] for CKKS. All experiments were done on a machine with i7-11700K CPU (8 cores with 16 threads; 3.6 GHz) and 64 GB of RAM.

Detials on Instantitation. We use an approximation sign function in Cheon et al. [18] to implement (α, ϵ)-close to $sgn(X)$. We select $\alpha = 66$ to make $\overline{\mathsf{ct}.t}$ reasonably small and select $\epsilon = 2^{-16}$. We evaluate a polynomial composed of g_4 (7 times) and f_4 (3 times) from Algorithm 3 in [18]. To estimate $\mathsf{ct}.t$, we use the static noise estimation in OpenFHE. With $MN = n/2$, we measure $\mathsf{ct}.t \approx 2^{8.1}$; hence we select $\mathsf{ct}.t = 2^9$ for noise flooding.

From the above settings, we set the following parameters of CKKS: the ring dimenstion $n = 2^{17}$, the scaling factor $\Delta = 2^{59}$, maximum level $L = 42$. The remaining parameters are chosen to ensure 128-bit security according to to homomorphic encryption standard [2]. In addition, we apply the noise flooding to ensure 40-bit statistical security and $q = 2^5$ by following our analysis.

Evaluation Results. For the performance evaluation, we measured the elapsed time and communication cost for various values of M and N, increasing M from $2^{11} \approx 2,000$ to $2^{11} \times 6 \approx 12,000$, with fixed d and N. To show the effect of set element dimensions, we also experimented with $d = 2^4$ to $d = 2^9$. Note that previous studies [29,30,3,26] only considered cases where d is at most 10. The results in Tab. 2 show that both communication and computation costs grow linearly

(N,M)	$d=2^7$		
	Comm. (GB)	**Comp. (s)**	
		$\mathcal{R}$	$\mathcal{S}$
$(2^5,2^{11})$	10.15	65.71	53.63
$(2^5,2^{11}\cdot 2)$			106.71
$(2^5,2^{11}\cdot 3)$			160.15
$(2^5,2^{11}\cdot 4)$			216.19
$(2^5,2^{11}\cdot 5)$			273.19
$(2^5,2^{11}\cdot 6)$			334.10

d	$(N,M)=(2^5,2^{11})$		
	Comm. (GB)	**Comp. (s)**	
		$\mathcal{R}$	$\mathcal{S}$
2^4	1.41	11.20	30.06
2^5	2.65	19.00	32.80
2^6	5.15	34.67	39.42
2^7	10.15	65.97	51.60
2^8	20.14	128.07	76.29
2^9	40.13	243.46	128.59

Table 2. Implementation results of $\pi_{\texttt{FPHE}}$.

with the dimension, and as M increases linearly, the sender's computational cost grows accordingly while the receiver's costs remain unchanged. We also highlight that $\pi_{\texttt{PFHE}}$ covers the case $d = 2^9$, which was computationally infeasible in all existing works but is necessary to handle real-world applications [31,21,34,32].

5 Conclusion

In this work, we propose FPHE, a novel doubly efficient fuzzy PSI protocol designed for handling high-dimensional data with cosine similarity score as a similarity metric. FPHE achieves linear communication and computation costs with respect to the dimension, requiring much weaker assumptions compared to several previous Fuzzy PSI proposals [29,30,3,27]. This overcomes the limitations of previous works that either rely on strong assumptions or incur exponential costs. We also show that our FPHE is secure under semi-honest security model, by carefully addressing the information leakage from approximation operations with noise flooding technique. Additionally, we extend FPHE to support labeled and circuit fuzzy PSI, enriching its applicability to various scenarios.

We leave several interesting future directions. First, we note that the major bottleneck of our protocol is computing the approximation of the sign function. We expect that our protocol can be improved by employing more efficient methods [36,42,35]. Of independent interest, it would also be interesting to devise an alternative approach for designing fuzzy PSI protocols without sign function. In addition, our protocol is based on the semi-honest security model. Extending our protocol to defend against stronger adversaries, such as malicious security, would be another interesting research direction. Finally, with a slight modification, we expect that our framework would support the fuzzy PSI with L_p distance over the Euclidean space, hence increasing its flexibility for a wider range of applications. We leave investigations for these aspects as future work.

References

1. Official python wrapper for openfhe (2022), `https://github.com/openfheorg/openfhe-python`
2. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., et al.: Homomorphic encryption standard. Protecting privacy through homomorphic encryption pp. 31–62 (2021)

3. van Baarsen, A., Pu, S.: Fuzzy private set intersection with large hyperballs. In: EUROCRYPT. pp. 340–369. Springer (2024)
4. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. Adv. Neural Inform. Process. Syst. **32** (2019)
5. Ben-Efraim, A., Nissenbaum, O., Omri, E., Paskin-Cherniavsky, A.: Psimple: Practical multiparty maliciously-secure private set intersection. In: ACM Asia Conf. Comp. and Comm. Sec. pp. 1098–1112 (2022)
6. Bhowmick, A., Boneh, D., Myers, S., Talwar, K., Tarbe, K.: The apple psi system. Apple, Inc., Tech. Rep (2021)
7. Bienstock, A., Patel, S., Seo, J.Y., Yeo, K.: Near-optimal oblivious key-value stores for efficient psi, psu, and volume-hiding multi-maps. In: USENIX Secur. Symp. pp. 301–318 (2023)
8. Bienstock, A., Patel, S., Seo, J.Y., Yeo, K.: Batch pir and labeled psi with oblivious ciphertext compression. IACR Cryptol. ePrint Arch. **2024**, 215 (2024)
9. Buddhavarapu, P., Knox, A., Mohassel, P., Sengupta, S., Taubeneck, E., Vlaskin, V.: Private matching for compute. Cryptology ePrint Archive (2020)
10. BUKATY, P.: The California Consumer Privacy Act (CCPA): An implementation guide. IT Governance Publishing (2019), `http://www.jstor.org/stable/j.ctvjghvnn`
11. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: IEEE Symp. Found. Comput. Sci. (FOCS). pp. 136–145. IEEE (2001)
12. Canonne, C.L., Kamath, G., Steinke, T.: The discrete gaussian for differential privacy. Adv. Neural Inform. Process. Syst. **33**, 15676–15688 (2020)
13. Chakraborti, A., Fanti, G., Reiter, M.K.: Distance-aware private set intersection. In: USENIX Secur. Symp. pp. 319–336 (2023)
14. Chase, M., Miao, P.: Private set intersection in the internet setting from lightweight oblivious prf. In: CRYPTO. pp. 34–63. Springer (2020)
15. Chen, H., Huang, Z., Laine, K., Rindal, P.: Labeled psi from fully homomorphic encryption with malicious security. In: ACM Conf. Comp. and Comm. Secur. pp. 1223–1237 (2018)
16. Chen, H., Laine, K., Rindal, P.: Fast private set intersection from homomorphic encryption. In: ACM Conf. Comp. and Comm. Secur. pp. 1243–1255 (2017)
17. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: ASIACRYPT. pp. 409–437. Springer (2017)
18. Cheon, J.H., Kim, D., Kim, D.: Efficient homomorphic comparison methods with optimal complexity. In: ASIACRYPT. pp. 221–256. Springer (2020)
19. Cheon, J.H., Kim, D., Kim, D., Lee, H.H., Lee, K.: Numerical method for comparison on homomorphically encrypted numbers. In: ASIACRYPT. pp. 415–445. Springer (2019)
20. Cong, K., Moreno, R.C., da Gama, M.B., Dai, W., Iliashenko, I., Laine, K., Rosenberg, M.: Labeled psi from homomorphic encryption with reduced computation and communication. In: ACM Conf. Comp. and Comm. Secur. pp. 1135–1150 (2021)
21. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4690–4699 (2019)
22. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4690–4699 (2019)

23. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L.: Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. Adv. Neural Inform. Process. Syst. **35**, 30318–30332 (2022)
24. Duong, T., Phan, D.H., Trieu, N.: Catalic: Delegated psi cardinality with applications to contact tracing. In: ASIACRYPT. pp. 870–899. Springer (2020)
25. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: ASIACRYPT. pp. 1–19. Springer (2004)
26. Gao, Y., Qi, L., Liu, X., Luo, Y., Wang, L.: Efficient fuzzy private set intersection from fuzzy mapping. In: ASIACRYPT. pp. 36–68. Springer (2025)
27. Garimella, G., Goff, B., Miao, P.: Computation efficient structure-aware psi from incremental function secret sharing. In: CRYPTO. pp. 309–345. Springer (2024)
28. Garimella, G., Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: Oblivious key-value stores and amplification for private set intersection. In: CRYPTO. pp. 395–425. Springer (2021)
29. Garimella, G., Rosulek, M., Singh, J.: Structure-aware private set intersection, with applications to fuzzy matching. In: CRYPTO. pp. 323–352. Springer (2022)
30. Garimella, G., Rosulek, M., Singh, J.: Malicious secure, structure-aware private set intersection. In: CRYPTO. pp. 577–610. Springer (2023)
31. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: Eur. Conf. Comput. Vis. pp. 241–257. Springer (2016)
32. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: ACM Conf. Inform. and Knowl. Manag. pp. 2333–2338 (2013)
33. Ion, M., Kreuter, B., Nergiz, A.E., Patel, S., Saxena, S., Seth, K., Raykova, M., Shanahan, D., Yung, M.: On deploying secure computing: Private intersection-sum-with-cardinality. In: IEEE Eur. Symp. Secur. Priv. (EuroS&P). pp. 370–389. IEEE (2020)
34. Kim, M., Jain, A.K., Liu, X.: Adaface: Quality adaptive margin for face recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18750–18759 (2022)
35. Kim, S., Cho, W.: Accelerating homomorphic comparison operations for thresholding using an asymmetric input range and input scaling. In: Proceedings of the Great Lakes Symposium on VLSI 2024. pp. 427–432 (2024)
36. Lee, E., Lee, J.W., No, J.S., Kim, Y.S.: Minimax approximation of sign function by composite polynomial for homomorphic comparison. IEEE Trans. Dependable Secure Comput. **19**(6), 3711–3727 (2021)
37. Lepoint, T., Patel, S., Raykova, M., Seth, K., Trieu, N.: Private join and compute from pir with default. In: ASIACRYPT. pp. 605–634. Springer (2021)
38. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: EUROCRYPT. pp. 648–677. Springer (2021)
39. Li, B., Micciancio, D., Schultz-Wu, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: CRYPTO. pp. 560–589. Springer (2022)
40. Lindell, Y.: How to simulate it–a tutorial on the simulation proof technique. Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich pp. 277–346 (2017)
41. Malali, N., Keller, Y.: Learning to embed semantic similarity for joint image-text retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 10252–10260 (2021)
42. Moon, J., Omarov, Z., Yoo, D., An, Y., Chung, H.: Adaptive successive over-relaxation method for a faster iterative approximation of homomorphic operations. Cryptology ePrint Archive (2024)

43. Mouris, D., Masny, D., Trieu, N., Sengupta, S., Buddhavarapu, P., Case, B.: Delegated private matching for compute. Proc. Priv. Enhancing Technol. (2024)
44. Movahedi, M., Case, B.M., Honaker, J., Knox, A., Li, L., Li, Y.P., Saravanan, S., Sengupta, S., Taubeneck, E.: Privacy-preserving randomized controlled trials: A protocol for industry scale deployment. In: Proceedings of the 2021 on Cloud Computing Security Workshop. pp. 59–69 (2021)
45. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: Psi from paxos: fast, malicious private set intersection. In: EUROCRYPT. pp. 739–767. Springer (2020)
46. Pinkas, B., Schneider, T., Zohner, M.: Scalable private set intersection based on ot extension. ACM Trans. on Priv. and Secur. (TOPS) **21**(2), 1–35 (2018)
47. Poddar, R., Kalra, S., Yanai, A., Deng, R., Popa, R.A., Hellerstein, J.M.: Senate: a maliciously-secure mpc platform for collaborative analytics. In: USENIX Secur. Symp. pp. 2129–2146 (2021)
48. Raghuraman, S., Rindal, P.: Blazing fast psi from improved okvs and subfield vole. In: ACM Conf. Comp. and Comm. Secur. pp. 2505–2517 (2022)
49. Richardson, D., Rosulek, M., Xu, J.: Fuzzy psi via oblivious protocol routing. Cryptology ePrint Archive (2024)
50. Rindal, P., Schoppmann, P.: Vole-psi: fast oprf and circuit-psi from vector-ole. In: EUROCRYPT. pp. 901–930. Springer (2021)
51. Shi, R.H., Li, Y.F.: Quantum private set intersection cardinality protocol with application to privacy-preserving condition query. IEEE Trans. Circuits Syst. I: Regul. Pap. **69**(6), 2399–2411 (2022)
52. Son, Y., Jeong, J.: Psi with computation or circuit-psi for unbalanced sets from homomorphic encryption. In: ACM Asia Conf. Comp. and Comm. Sec. pp. 342–356 (2023)
53. Sun, Y., Katz, J., Raykova, M., Schoppmann, P., Wang, X.: Actively secure private set intersection in the client-server setting. In: ACM Conf. Comp. and Comm. Secur. pp. 1478–1492 (2024)
54. Uzun, E., Chung, S.P., Kolesnikov, V., Boldyreva, A., Lee, W.: Fuzzy labeled private set intersection with applications to private real-time biometric search. In: USENIX Secur. Symp. pp. 911–928 (2021)
55. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
56. Wu, M., Yuen, T.H.: Efficient unbalanced private set intersection cardinality and user-friendly privacy-preserving contact tracing. In: USENIX Secur. Symp. pp. 283–300 (2023)
57. Yang, Y., Dong, X., Shen, J., Cao, Z., Yang, Y., Zhou, J., Fang, L., Liu, Z., Ge, C., Su, C., et al.: Mdppc: Efficient scalable multiparty delegated psi and psi cardinality. In: Annu. Int. Conf. Priv. Secur. Trust (PST). pp. 1–7. IEEE (2023)
58. Zhao, H., Zhang, W., Li, X., Shang, S., Huang, K., Zhang, X.: Towards efficient delegated private set intersection cardinality protocol. In: Int. Conf. Comput. Support. Coop. Work Des. (CSCWD). pp. 729–734. IEEE (2024)

A Definition of Semi-honest Security

In this section, we provide the formal definition for semi-honest security. We borrow the notations in [40]. Let us consider a two-party protocol π for computing $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$, where one party computes $\mathcal{F}_1(x, y)$ on input x and the other computes $\mathcal{F}_2(x, y)$ on input y. Then, π securely computes $\mathcal{F}$ in the presence of static semi-honest adversaries if it satisfies the following definition.

Definition 5 ([40]). *Let $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ be a functionality. We say that π securely computes a functionality $\mathcal{F}$ in the presence of static semi-honest adversaries if there exists probabilistic polynomial-time algorithm S_1 and S_2 such that*

$$\{\mathsf{S}_1(1^n, x, \mathcal{F}_1(x, y)), \mathcal{F}(x, y)\}_{x,y,n} \overset{c}{\equiv} \{\mathsf{view}_1^\pi(x, y, n), \mathsf{output}^\pi(x, y, n)\}_{x,y,n}, \textit{ and}$$

$$\{\mathsf{S}_2(1^n, y, \mathcal{F}_2(x, y)), \mathcal{F}(x, y)\}_{x,y,n} \overset{c}{\equiv} \{\mathsf{view}_2^\pi(x, y, n), \mathsf{output}^\pi(x, y, n)\}_{x,y,n},$$

where x and y are inputs of protocol, and $n \in \mathbb{N}$ is a security parameter.

Here, $\mathsf{view}_i^\pi(x, y, n)$ denotes the view of party P_i during an execution of π, including the messages it received. $\mathsf{output}^\pi(x, y, n)$ denotes the outputs of the parties. In addition, the notation $\overset{c}{\equiv}$ means computational indistinguishability between two probability ensembles.

B Proof for Theorem 2

Proof. Due to the parameter settings described earlier, the index set and the output of the functionality are identical with a negligible failure probability. Hence, it suffices to show the existence of probabilistic polynomial-time algorithms S_1 and S_2, which simulate the views of the client and the server, respectively.

First, we show how S_1 simulates the client's view in the real protocol. Specifically, the client's view consists of $\overline{\mathbf{z}} \in \mathcal{R}^2$. The simulator S_1 receives the index set I as the output of the functionality. From the set, it generates a vector $\mathbf{z}' = (z'_1, \ldots, z'_{n/2})$ as follows.; For each $i \in [N]$, let $P_i = \{(j-1)N + i \mid j \in [M-1]\}$. If $\mathrm{idx} \in I$, the simulator randomly selects $k \in P_{\mathrm{idx}}$, sets $z'_k = 1$, and sets $z'_j = 0$ for all $j \in P_{\mathrm{idx}} \setminus \{k\}$. If $\mathrm{idx} \notin I$, it sets $z'_j = 0$ for all $j \in P_{\mathrm{idx}}$. Then, S_1 runs $\overline{\mathbf{z}}' \leftarrow \mathsf{Enc}_{pk}(\mathbf{z}')$ and add $(e, 0)$ to it for $e \leftarrow N_{\mathbb{Z}^n}(0, \sigma^{*2}\overline{\mathsf{ct}.t}^2)$.

The remaining part is to show the indistinguishability between $\overline{\mathbf{z}}$ and $\overline{\mathbf{z}}'$ with the knowledge of secret key. Note that this corresponds to the scenario of q-IND-CPA^D game, where the knowledge of the secret key corresponds to the decryption query. First of all, since the indices of the server's data, $\mathbf{q}_i$, is randomly permuted during running the protocol, we can ensure that the the distribution of nonzero entries in $\mathbf{z}$ and $\mathbf{z}'$ are identical. Moreover, in Section 3.3, we already show that the proposed amount of noise is sufficient for satisfying q-IND-CPA^D security according to Theorem 1.

Finally, we show how S_2 simulates the server's view in the real protocol. Specifically, the server's view consists of $pk, evk,$ and $\{\mathbf{p}_i\}_{i \in [d]}$. Similarly, the simulator S_2 can simulate the server's view by uniformly sampling $2d + 4$ components from $\mathcal{R}$. Since CKKS is IND-CPA secure under the hardness assumption of the Ring-LWE problem, the server's view and simulated view are computationally indistinguishable.