Approximate PSI with Near-Linear Communication

Wutichai Chongchitmate*[1], Steve Lu†[2], and Rafail Ostrovsky‡ §[3]

[1]Chulalongkorn University
[2]Stealth Software Technologies, Inc.
[3]UCLA

Abstract

Private Set Intersection (PSI) is a protocol where two parties with individually held confidential sets want to jointly learn (or secret-share) the intersection of these two sets and reveal nothing else to each other. In this paper, we introduce a natural extension of this notion to approximate matching. Specifically, given a distance metric between elements, an *approximate* PSI (Approx-PSI) allows to run PSI where "close" elements match. Assuming that elements are either "close" or sufficiently "far apart", we present an Approx-PSI protocol for Hamming distance that dramatically improves the overall efficiency compared to all existing approximate-PSI solutions. In particular, we achieve a near-linear $\tilde{\mathcal{O}}(n)$ communication complexity, an improvement over the previously best-known $\tilde{\mathcal{O}}(n^2)$. We also show Approx-PSI protocols for Edit distance (also known as Levenstein distance), Euclidean distance and angular distance by deploying results on low distortion embeddings to Hamming distance. The latter two results further imply secure Approx-PSI for other metrics such as *cosine similarity* metric. Our Approx-PSI for Hamming distance is up to 20x faster and communicating 30% less than best known protocols when (1) matching small binary vectors; or (2) matching large threshold; or (3) matching large input sets. We demonstrate that the protocol can be used to match similar images through spread spectrum of the images.

[*] `wutichai.ch@chula.ac.th`

[†] `steve@stealthsoftwareinc.com`

[‡] `rafail@cs.ucla.edu`

[§] This research was supported in part by DARPA under Cooperative Agreement HR0011-20-2-0025, the Algorand Centers of Excellence programme managed by Algorand Foundation, NSF grants CNS-2246355, CCF-2220450, CNS-2001096, US-Israel BSF grant 2022370, Amazon Faculty Award and Sunday Group. Any views, opinions, findings, conclusions, or recommendations contained herein are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, the Department of Defense, the Algorand Foundation, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein.

1 Introduction

Secure computation protocols enable two or more parties to engage in distributed computation while preserving the confidentiality of their inputs. Among these, private set intersection (PSI) has recently garnered significant research attention as a specialized secure computation protocol. PSI allows parties to compute the intersection, the common elements between their input sets without exposing other unrelated data. Consequently, at the end of the protocol, the parties are only aware of the shared elements, ensuring confidentiality. This characteristic has made PSI indispensable in various applications ranging from private contact discovery and business data matching to efficient data management and contact tracing. We refer the reader to recent literature [IKN^{+}20, PRTY19, DPT20, CM20, MPR^{+}20, RT21, RS21, GPR^{+}21] and references therein.

In numerous applications, identifying an exact overlap between both parties' datasets might be improbable or overly restrictive. Here, discovering approximate matches – elements that share a "distance" under a specified threshold – becomes increasingly relevant. In the rapidly evolving landscape of privacy-preserving data analysis, these secure protocols adept at identifying such approximate matches are gaining traction, signifying their potential in recognizing analogous elements spanning datasets. This can be useful in various applications, such as:

- Biometric data: If two parties have databases of biometric data (like fingerprints or facial features represented as vectors), they may want to find matches, or near matches due to variations in sampling biometric data, without revealing the entirety of their databases [Dau09, UCK^{+}21].

- Genomic data: Parties might be interested in finding genomic sequences that are close matches without revealing sensitive genomic data [MKHSO17, WHZ^{+}15] as similarities of such data are already useful in medical diagnoses or resulting features in biology.

- Security: Traditional methods of identifying malicious network traffic often rely on exact signature matches or IP addresses [MPDC19] to pinpoint known security threats, potentially missing out on novel or slightly altered malicious payloads. With near match intersection, network security tools can detect and flag traffic patterns that closely resemble known attack signatures, even if they are not an exact match, or cover ranges of potentially malicious IP addresses [CSF^{+}07, WACL10].

- Image data: In fields such as computer vision and image processing, parties may seek to perform similar image matching without disclosing the entirety of their image datasets. This is particularly relevant in applications where identifying similar images can aid in tasks such as object recognition, content-based image retrieval, and image classification, contributing to advancements in fields like autonomous vehicles, surveillance systems, and medical imaging analysis.

Distance-Aware PSI. Recently, Chakraborti *et al.* [CFR23] introduced a variant of PSI, called *distance-aware PSI (DA-PSI)*. In this setting, two parties jointly compute a set of pairs of elements, one from each of their individual datasets, that are within a specified threshold based on a particular distance metric. More precisely, given input sets $A, B \subseteq \mathcal{U}$ for the two parties, and a distance metric δ defined on $\mathcal{U}$, the objective of DA-PSI is to securely compute a set $S = \{(a,b) \in A \times B : \delta(a,b) \leq d\}$, with d being a pre-defined threshold.

Nevertheless, a significant challenge associated with existing DA-PSI protocols is their extensive communication complexity. This complexity limits their practicality, especially in contexts demanding fast or nearly instantaneous feedback. In particular, the communication and computation complexity of the DA-PSI protocol for the Hamming distance in [CFR23] is $\mathcal{O}(n^2)$, where n represents the size of sets A and B. Such scalability issues render these protocols impractical for analyzing extensive data sets.

Structure-Aware PSI. Garimella *et al.* [GRS22] introduced another related PSI variant called *structure-aware PSI (sa-PSI).* In this setting, the receiver's set adheres to a specific structure, for instance, a union of fixed-radius balls based on a particular distance metric. The output is the same as the standard PSI for the receiver, but the efficiency (computation and communication) is only influenced by the structure of the receiver's set. In the case of the union of balls, the efficiency would depends on the number of balls present in the receiver's set, rather than the total number of individual elements.

The sa-PSI concept is broad since the sender's set structure can vary widely. Nonetheless, a primary area of interest within sa-PSI centers around the previously mentioned case of a union of disjoint balls with a fixed radius. Considering a distance metric δ and a ball radius d, sa-PSI is similar to DA-PSI with distance threshold d. However, the distinctions lies in their outputs: DA-PSI yields a set of pairs to both parties, when viewing in terminology of sa-PSI, include both the sender's elements and the centers of the receiver's balls, while sa-PSI only outputs to the party with structured set. When this result made known to the other party, the centers are still concealed. Notably, in a semi-honest model, parties involved in an sa-PSI protocol can subsequently exchange data to discern these pairs, suggesting that sa-PSI implies DA-PSI under these conditions, but not the other way around. While the sa-PSI protocol in [GRS23] is linear in the number of balls, their construction is specifically for the ℓ_∞ norm for integral vectors.

1.1 Approximate PSI

Here, we consider another setting of PSI where elements are from a metric space, i.e., a set $\mathcal{U}$, equipped with a distance metric δ. Instead of computing the intersection or precise matches of elements from each set, we consider approximate matches (with respect to δ), which are pairs of elements that have distance at most d. When $d = 0$, this setting is equivalent to the standard PSI. When $d > 0$ and the protocol output is the set of pairs of matches, the variant is called DA-PSI by [CFR23]. Given that both input sets have size n, the upper bound for matched pairs is n^2. This creates challenges to avoid the quadratic communication as in [CFR23].

To reduce the excessive communication costs, we introduce an additional constraint: for any $a \in A$ and $b \in B$, either $\delta(a,b) \leq d$ or $\delta(a,b) \geq td$ for some $t \geq 2$. This allows for clustering elements from A and B, that are within distance d of each other. Each cluster in each input set is represented by only one element from that cluster. By only considering the representations of the clusters, we further assume that elements $a, a' \in A$ satisfy $\delta(a,a') \geq td$, and each element in A can match with at most one element in B and vice versa. Finally, one party (or both) outputs which elements in their set near-match with elements in the other party's set. Our proposed PSI variant offers flexibility: it can output either one party's elements (as in sa-PSI), both parties' own elements, or element pairs (as in DA-PSI). We call this problem an *approximate PSI (Approx-PSI)*, and t the *gap* distinguishing matches from non-matches. The setting in the first variant is similar to sa-PSI for the structure of the union of disjoint fixed-radius balls with center $(t-1)d$ apart, with additional assumption on non-structure side that elements must also be far apart.

Nevertheless, imposing such a restriction is difficult when honest parties are unaware of the counterpart's elements. Our approach assumes both input sets lie within a subset $\mathcal{S} \subseteq \mathcal{U}$, where every pair of elements in this subset is either near or far apart. This condition allows functionality to simply check if input sets are subsets of $\mathcal{S}$, and encapsulates the preceding one for individual sets.

Such condition has various applications when the distance is generally of concern. For instance, this set may contain a compilation of texts and their small-error-induced variants. A single base text could have close relatives with just a handful of typographical mistakes, while remaining entirely distinct from other base texts within the same set. Similarly, it could be a set of ID numbers engineered with error-correcting properties or checksums. In these sets, tiny changes can make elements look very similar to the original, even though they're different. This makes it important to have good ways to tell these small differences apart from the big ones.

We note that when such condition does not hold across the input sets, our protocol remains correct when elements within each set are clustered and only represented by elements that are far apart. The false negatives only occurs for the omitted elements clustered around representatives as the transitive property of being near no longer holds.

Our goal is to find an approximate PSI protocol with linear communication complexity in n, the size of both input sets, improving the result directly implies by the DA-PSI of [CFR23].

Euclidean and angular distance. The Euclidean distance metric measures the straight-line distance between two points in a multi-dimensional space and is especially useful for capturing the geometric relationships among sets of continuous variables. Unlike other metrics that focus on discrete modifications or element-wise comparisons, the Euclidean distance offers a holistic view of the positional relationship between entire vectors, making it suitable for a wide range of applications that require assessing the similarity or dissimilarity between multi-attribute entities.

Angular distance measures the angle between two vectors in a multi-dimensional space, focusing on their directions rather than absolute position. This metric is useful for assessing the orientation or alignment between vectors, making it ideal for applications like text similarity in natural language processing or preference analysis in recommendation systems. Unlike Euclidean distance, which measures linear spacing, angular distance evaluates how parallel or divergent vectors are, highlighting relationships based on direction rather than distance.

The integration of Euclidean and angular distance metrics into Approx-PSI protocols has vast potential, particularly in fields requiring spatial or multi-dimensional analysis. In machine learning and data science, for example, such protocols could be invaluable for securely conducting k-means clustering or nearest neighbor searches across distributed datasets. In financial analytics, Approx-PSI with Euclidean or angular distance can enhance fraud detection algorithms by identifying not just the occurrence of similar transactions across multiple datasets but also the closeness of those transactions in a multi-dimensional feature space, such as amount, location, and time. Likewise, in medical research, Approx-PSI could enable disparate healthcare organizations to securely compare patient data, finding similarities in symptoms, treatment responses, or other multi-dimensional health metrics, without compromising individual privacy. The ability to calculate the Euclidean distance between intersecting elements securely expands the utility of Approx-PSI beyond mere set intersection, allowing for more nuanced, privacy-preserving analytics in multi-dimensional data environments.

Edit distance. Edit distance (specifically,*Levenshtein* distance) serves as a crucial tool for assessing the similarity between two sequences, such as strings of text, genetic data, or time-series numerical data. By accounting for the minimum number of single-character edits—insertions, deletions, or substitutions—required to transform one sequence into the other, edit distance metrics offer a fine-grained understanding of sequence similarity or disparity. In disciplines ranging from computational biology and linguistics to data mining and cybersecurity, the capability to efficiently measure or estimate edit distance forms the backbone of numerous fundamental operations such as sequence alignment, clustering, and anomaly detection [WHZ$^+$15]. Hence, edit distance plays an indispensable role in quantitative analysis across multiple fields.

When coupled with edit distance metrics, Approx-PSI opens up new avenues for secure, privacy-preserving computations that require nuanced understandings of data similarity. For example, in genomic research, Approx-PSI could enable two entities to securely identify shared genetic markers and evaluate the minutiae of those markers' sequences. Similarly, in the area of natural language processing, Approx-PSI can facilitate secure collaborative filtering or content recommendation by accounting for the edit distance between text strings. By enabling the secure comparison of sequences without compromising the confidentiality of the data, Approx-PSI protocols incorporating edit distance metrics stand to significantly advance the state of secure, multi-party computations where sequence similarity are of concern.

1.2 Related Work

PSI for approximate or near-matches for Hamming distance has been studied in the context of securely comparing biometric or fuzzy data [OPJM10, HEKM11, UCK$^+$21]. Secure Hamming distance comparison can be turned into DA-PSI or Approx-PSI protocols by comparing all n^2 pairs of elements [OPJM10, HEKM11]. Uzun *et al.* [UCK$^+$21] allows comparing multiple elements at once using fully homomorphic encryption (FHE).

Chakraborti *et al.* [CFR23] formally defined and constructed the first DA-PSI for Hamming distance that the communication and computation complexity do not depend on the element size (ℓ). Thus, the resulting protocol is more efficient when $d \ll \ell$. However, their protocol is quadratic in the number of elements. Additionally, they also constructed DA-PSI for integers with their difference as distance with linear communication complexity.

Table 1: Asymptotic communication of our protocols in comparison to existing works. The protocol for the Euclidean distance assumes that all input vectors are within a ball of constant radius. We assume $\log n < \lambda < \ell$ to simplify some notations.

Metric	Protocol	Gap	Communication
ℓ_∞	[GRS23]	$\mathcal{O}(1)$	$\mathcal{O}(n\lambda^2\ell + \lambda d^\ell)$
Hamming	[CFR23]	1	$\mathcal{O}(n^2 d^2 \lambda)$
	Ours	$\mathcal{O}(\log n)$ $\mathcal{O}(\log n / \log\log n)$ $\mathcal{O}(\log\log n)$ $t = \mathcal{O}(1)$	$\mathcal{O}(n\lambda\ell)$ $\mathcal{O}(n(\text{polylog}\, n)\lambda\ell)$ $n^{1+o(1)}\lambda\ell$ $\mathcal{O}(n^{1+\frac{1}{t-1}}\lambda\ell)$
Euclidean (ℓ_2)	Ours	$\mathcal{O}(\log n)$ $\mathcal{O}(1)$	$\mathcal{O}(n(\text{polylog}\, n)\lambda^2)$ $\mathcal{O}(n^{1+\epsilon}\lambda^2)$
Angular	Ours	$\mathcal{O}(\log n)$ $\mathcal{O}(1)$	$\mathcal{O}(n \log^2 n\lambda^2)$ $\mathcal{O}(n^{1+\epsilon}\lambda^2)$
Edit	Ours	$2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})} \log n$ $2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}$	$\mathcal{O}(n\lambda^2\ell^2 \log \ell)$ $\mathcal{O}(n^{1+\epsilon}\lambda^2\ell^2 \log \ell)$

Garimella *et al.* [GRS22] defined and constructed the sa-PSI protocol for the case of disjoint balls of u-bit integer vectors with ℓ_∞ norm, and a more efficient one where centers of the balls are far apart. The original protocols is secure against semi-honest adversaries, and later improved in [GRS23] using derandomizable function secret sharing to be secure against malicious adversaries.

1.3 Our Results

In this work, we present Approx-PSI protocols for three distance metrics: Hamming distance, Euclidean distance (and the related cosine similarity) and edit distance. We summarize our results in Table 1.

Hamming distance. Our main result is an Approx-PSI protocol for Hamming distance for gap $t \geq 2$ with $\tilde{\mathcal{O}}(n^{1+\frac{1}{t-1}})$ communication. For $t = \mathcal{O}(\log n)$, the protocol has near linear $\tilde{\mathcal{O}}(n)$ communication, and only gains poly-logarithmic or sub-linear multiplicative factor for $t = \mathcal{O}(\frac{\log n}{\log\log n})$ or $\mathcal{O}(\log\log n)$, respectively.

Our protocol draws inspiration from [CFR23], where they constructed a DA-PSI for the Hamming distance. In their work, they introduced an efficient subprotocol that securely compares the Hamming distance between two elements. Notably, the communication complexity of this subprotocol depends solely on the threshold d and the security parameter, and remains independent of the element size ℓ. Nevertheless, their DA-PSI protocol executes this subprotocol across all n^2 pairs to find all potential matches. Such an approach inevitably results in quadratic communication and computation in the size of input sets. Circumventing this quadratic efficiency is inherently challenging in standard DA-PSI, given that match count could go up to n^2.

However, our Approx-PSI can be reduced to the case where elements of the same set

Table 2: Number of calls to subprotocols in our Approx-PSI for Hamming distance.

Gap	Number of Calls	
	secret-shared PSI	secret-shared Hamming distance comparison and vector multiplication
$\mathcal{O}(\log n)$	$\mathcal{O}(\log n + \lambda)$	$\mathcal{O}(n(\log n + \lambda))$
$\mathcal{O}(\log\log n)$	$\mathcal{O}(n^{o(1)}(\log n + \lambda))$	$n^{1+o(1)}(\log n + \lambda)$
$t = \mathcal{O}(1)$	$\mathcal{O}(n^{\frac{1}{t-1}}(\log n + \lambda))$	$\mathcal{O}(n^{1+\frac{1}{t-1}}(\log n + \lambda))$

are far apart, capping the match count at a maximum of n. This offers a way to bypass the exhaustive comparison of all n^2 pairs. Our strategy incorporates an additional phase to eliminate non-matching pairs. To achieve this, we adopt the random projection idea from [KOR98]. First, both parties jointly sample a random subset of positions. Then, every party calculates a set of element projections based on these agreed positions. Elements in a matched pair are more likely to have identical projections, as opposed to elements that are far apart. These projections then undergo an exact match evaluation, leveraging the traditional PSI for security. Finally, the probability can be amplified by repeatedly and independently sampling positions, and computing intersections of projections.

However, the outputs of intermediate steps of the aforementioned method disclose more about the distance between elements in the input sets than what the intended Approx-PSI outcome should reveal. Consequently, both the PSI subprotocol and the secure Hamming distance comparison subprotocol must output secret shares of their respective results. While there exist known PSI protocols that output secret shares, prominent among them being circuit-based PSI like the protocol in [RS21], the secure Hamming distance comparison subprotocol deployed in [CFR23] is not suitable to be compiled to output secret shares. To addressing this challenge, we either rely on generic garbled circuit technique, or substantially modify the subprotocol using the techniques from [GS19, KMWF07]. Combining these subprotocols gives a Approx-PSI for Hamming distance. Our construction provides a reduction from Approx-PSI to the standard PSI (with secret-shared output) using secret-shared Hamming distance comparison test and other secret-shared operations with numbers shown in Table 2.

Euclidean distance. As the Euclidean distance is one of the most used distance metrics, there is a long line of work on embedding Euclidean distance or its related metrics such as cosine distance and angular distance into the Hamming distance. The ideas follow from the Johnson-Lindenstrauss lemma [JL84]. The recent line of work [PV14, OR15, HS20, DS20, DM21] gives low distortion of balls or the unit sphere centered at the origin in $\mathbb{R}^N$ with Euclidean metric or angular metric into binary string with Hamming distance. We construct the Approx-PSI using the similar method as the one with the edit distance.

Cosine similarity and Angular distance. Since cosine similarity, cosine distance and angular distance can be computed from the Euclidean distance, the Approx-PSI for Euclidean distance naturally gives the Approx-PSI for these metrics as well. Many Johnson-Lindenstrauss-styled embeddings are done directly for the angular distance [PV14, OR15]. We briefly discuss the Approx-PSI for the angular distance from these direct embeddings as well. This implicitly give us a second way to reach the cosine similarity as it has tighter connection to the angular distance.

Edit distance. Ostrovsky and Rabani [OR07] showed how to embed edit distance metric in Hamming distance metric with bounded distortion. Such embedding could not be used to construct standard DA-PSI as the distortion could turn a match into a non-match and vice versa. However, the Approx-PSI tolerates some degree of distortion. Thus, we can embed elements in Hamming distance metric, securely compute matches and look up the original elements in the result.

Implementation and Application in Image Matching We optimize and implement our Approx-PSI protocol in various parameter settings for Hamming distance and angular distance. We also present Approx-PSI for similar image matching using a variant of discrete cosine transform [CKLS97] that can match images that are resized, blurred or brightness/contrast adjusted through the angular distance metric embedding. This application demonstrates the adaptability of our Approx-PSI that can further be used to securely match any objects with similar embedding.

2 High-Level Overview of our Approach

In this section, we provide an informal overview of our approximate PSI protocol, starting from the protocol for Hamming distance. The basic idea of construction is inspired by the DA-PSI for the same distance metric in [CFR23]. Their strategy is to securely comparing the Hamming distance between two binary strings and applying this comparison for each element pair across two input sets. However, this approach incurs quadratic communication and computation complexity when both parties hold identically sized input sets.

In order to overcome this limitation, we consider three primary aspects:

Input Restriction: We limit the potential inputs to those that result in at most linear number of matches. This effectively translates to scenarios where each element in one input set corresponds to just one element in its counterpart. For meaningful enforcement of this condition, we impose a structure for all elements in both input sets: every pair of elements should be either near or far apart. Such a structure mirrors real-world scenarios where legitimate texts or numbers differ substantially, while their errors deviate only in few character or digit counts. The parties then consolidate their elements, ensuring each cluster is represented just once within their input set. This streamlined input set, now with at most linear number of matches, aligns with our goal for linear communication.

Efficient Matching: Despite the linear match limit, the necessity of comparing every possible pair still results in quadratic communication. Utilizing the technique in [KOR98], we minimize the comparisons to a near-linear count. Here, each binary vector is projected to a shorter vector based on randomly and independently selected positions. This reduces the problem of near-matching to exact matching. If two vectors differ by a few position, the probability that none of those position are chosen is high, leading to matching short vectors. This exact matching is securely and efficiently computed using standard PSI. Repeating this process logarithmic number of times ensures that our protocol finds all approximate matches except with negligible probability. On the other hand, if the number of selected positions is too small, an excessive number of vectors, even the non-matching ones, project, or "collide", into identical vectors. This scenario inadvertently raises the potential match count, leading back to near-quadratic communication. We meticulously adjust the position selection probability, ensuring minimal collisions while preserving actual matches. This results in a reduction from Approx-PSI to logarithmic number of standard PSI.

Information Leakage: The process of projecting and comparing the projections potentially leaks information, even with PSI. For instance, vectors with identical projection failing the Hamming distance check might inadvertently disclose some bits of a party's vector to the other party. To hide these intermediate results, we use secret-share version of both PSI and the Hamming distance comparison test. While ready-to-use PSI protocols that output secret shares of results exist, such as the circuit version in [RS21], efficient Hamming distance checks outputting secret shares remain unknown. A standard technique transforming one to output secret shares would result in a less efficient subprotocol with communication depending on the length of the binary inputs. Our goal is to maintain the efficiency in [CFR23] – one that is independent of input length. The subprotocol in [CFR23] also reveals both parties' inputs when matched, which forces their DA-PSI to output the result in pairs rather than only to the owner of each matched element. As our Approx-PSI may be customized to give the result to one party, we cannot follow their approach directly.

The secret-share Hamming distance comparison test can be constructed simply from garbled circuit. The resulting subprotocol is efficient for small and median size elements. For large elements (8000 bits or more), the length-independent comparison test can be constructed by combining the ideas from [CFR23, GS19, KMWF07]. We use [CFR23] as a starting point, representing binary vectors as subsets of finite field elements, whose Hamming distance corresponds to the size of set difference. These subsets can be further encoded as matrices whose subtraction corresponds to the set difference, using the idea in [GS19]. Moreover, the dimension of the matrices corresponds to the threshold value and the size of the set difference can be tested if above or below the threshold from the determinant of the matrix difference. The parties can jointly and securely compute the determinant using additive homomorphic encryption in [KMWF07]. Further modification of the homomorphically encrypted output gives the secret shares of the result. Finally, we utilize generic secret-sharing scheme operations for addition and multiplication to manipulate the secret shares between steps of our protocol.

We then combine the Approx-PSI protocol for Hamming distance with low distortion embedding from edit distance by [OR07], Euclidean distance by [DM21] and angular distance by [DS18] to construct Approx-PSI protocols for these distance metrics. We take advantage of the gap to guarantee that the pairs of elements that are near or far apart remain so after the embedding. Using the relationship between Euclidean distance, angular distance and cosine similarity, we also obtain the Approx-PSI protocol for cosine similarity.

Similar Image Matching Application: Approx-PSI enables various applications matching "similar" objects as long as they can be transformed into metric space with known embedding into binary vectors with Hamming distance metric. In particular, digital images can be transformed into real vectors through discrete cosine transform (DCT) [CKLS97] that are preserved under several image manipulations such as resizing and blurring. Applying such transformation followed by binary embedding [DS18], we obtain an Approx-PSI that can match similar images from two sets of images without revealing other non-matched images.

3 Preliminaries

We denote the set $\{1, 2, \ldots, n\}$ by $[n]$. Let $x \in \{0,1\}^*$. We denote the length of x by $|x|$. For $i \in [|x|]$, we denote by x_i the ith character in x. We use λ to denote the security parameter. We use the standard definition of negligible functions and computational indistinguishability [GM84]. We denote by $\Pr_r[A]$ the probability of an event A over coins r, and $\Pr[A]$ when r is not specified. We denote by $\mathbb{E}[X]$ the expectation of a random variable X. For a finite set S, we denote $a \leftarrow S$ a uniformly random choice of a from S. For a randomized algorithm A, let $A(x; r)$ denote running A on an input x with random coins r. If r is chosen uniformly at random with an output y, we denote $y \leftarrow A(x)$.

3.1 Approximate PSI

We consider the setting of two parties with input sets A, B whose elements are drawn from a subset $\mathcal{S}$ of the universe $\mathcal{U}$, equipped with a distance metric $\delta : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$. The subset $\mathcal{S}$ has the property that any pair of elements must be either near or far from each other. More specifically, for any elements $a, b \in \mathcal{S}$, either $\delta(a,b) \leq d$ (called *matched*, close, or near) or $\delta(a,b) \geq td$ (called *non-matched*, or far) for some $d > 0$ and $t > 1$. We call d the *threshold* and t the *gap*.

The approximate PSI (Approx-PSI) functionality is defined in Figure 1. The goal of the Approx-PSI is to find pairs of elements, one from each input set, that are near, i.e., approximate matches. We allow three possibilities for the output: only one party receives their matched elements;each party receives matched elements in their respective set; or both parties receive a set of matches pairs.

We note that any matched elements can be grouped by the following lemma.

Lemma 3.1. *Let δ be a distance metric on $\mathcal{U}$. Let $\mathcal{S} \subseteq \mathcal{U}$ such that for any $a, b \in \mathcal{S}$, either $\delta(a,b) \leq d$ or $\delta(a,b) \geq td$ for some $t > 1$. Let $a, a', b \in \mathcal{S}$ such that $\delta(a,b) \leq d$. Then*

1. *$\delta(a',b) \geq td$ if and only if $\delta(a,a') \geq (t-1)d$*
2. *$\delta(a',b) \leq d$ if and only if $\delta(a,a') \leq 2d$*

$\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$

Parameters. set size n, threshold d

Functionality.

1. Upon receiving a message (inputS, A) from the sender with $A \subseteq \mathcal{S}$ and $|A| = n$, store A; otherwise, ignore the message.
2. Upon receiving a message (inputR, B) from the receiver with $B \subseteq \mathcal{S}$ and $|B| = n$, store B; otherwise, ignore the message.
3. If both A and B are stored, compute $M = \{(a, b) \in A \times B : \delta(a, b) \leq d\}$; otherwise, abort. Let $M_A = \{a : (a, b) \in M\}$ and $M_B = \{b : (a, b) \in M\}$.
4. Send M_B to the receiver. Optionally, send M_A to the sender, or send M to both parties.

Figure 1: Ideal functionality for approximate private set intersection

Proof.

1. Suppose $\delta(a', b) \geq td$. Then by the triangle inequality,
$$td \leq \delta(a', b) \leq \delta(a, a') + \delta(a, b) \leq \delta(a, a') + d.$$
Thus, $\delta(a, a') \geq (t-1)d$. Now suppose $\delta(a, a') < (t-1)d$. By the triangle inequality,
$$\delta(a', b) \leq \delta(a, b) + \delta(a, a') < d + (t-1)d = td.$$

2. Suppose $\delta(a', b) \leq d$. Then by the triangle inequality,
$$\delta(a, a') \leq \delta(a, b) + \delta(a', b) \leq d + d = 2d.$$
Now suppose $\delta(a, a') > 2d$. By the triangle inequality,
$$2d < \delta(a, a') \leq \delta(a, b) + \delta(a', b) \leq d + \delta(a', b).$$
Thus, $\delta(a', b) > d$.

□

When $t > 2$, if $a, a' \in A$ satisfy $\delta(a, a') \leq d$, then for any $b \in \mathcal{S}$, either they both match with b or they are both far from b. Thus, we may group all $a' \in A$ within distance d from a into one class represented by a. Whenever, a and b are matched (as output by $\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$), then every a' in the same class are matched to b as well. We call the process of removing all $a' \in A$ within distance d from a representative $a \in A$ *clustering*, and adding the a' back if a is matched with some $b \in B$ *declustering*. By performing clustering and declustering in the beginning and at the end of an Approx-PSI protocol with semi-honest parties, we may further assume that elements of A are far apart, and so are elements of B.

3.2 Distance Metrics

In this work, we consider two distance metrics for binary strings: Hamming distance and edit distance. We let ℓ denote the length of the string, i.e., the universe $\mathcal{U} = \{0,1\}^\ell$.

For $x, y \in \{0,1\}^\ell$, the *Hamming distance* between x and y, denoted $\mathcal{H}(x, y)$, is the number of positions $i \in [\ell]$ such that $x_i \neq y_i$. We also denote $\mathcal{H}(x) = \mathcal{H}(x, 0)$, the *Hamming weight* of x. The *edit distance* (also known as Levenstein distance) between x and y, denoted $\mathrm{ed}(x, y)$, is the minimum number of insert, delete and substitute operations (one character at a time) needed to convert x to y.

In many practical contexts, the distance metric most frequently employed to measure the separation between two points or vectors in space is the Euclidean distance. When the vectors are normalized, we can consider them on a unit sphere and measure the shortest path on the sphere connecting two vectors. This distance is called *angular distance*. The Euclidean distance between two vectors can be computed from their dot product or angular distance, and vice versa. We refer to Appendix A for their formulas and relationship.

4 Building Blocks: Secret-Shared Operations

Our construction requires several operations whose outputs are secret shared between two parties to hide the intermediate results. In particular, the building blocks are secret-shared PSI, secret-shared Hamming distance comparison test, and operations on secret-shared data including scalar-vector multiplication.

4.1 Secret Sharing

In this work, we consider only a two-party secret sharing for binary strings. For a secret $s \in \mathcal{S}$, secret sharing of s are denoted $\mathsf{Share}(s) \to ([s]_0, [s]_1)$ (or $[s]_S, [s]_R$ when the shares belong to the sender and the receiver in PSI, respectively) where for any $s, s' \in \mathcal{S}$ and $i \in \{0,1\}$

$$\{[s]_i : \mathsf{Share}(s) \to ([s]_0, [s]_1)\} = \{[s']_i : \mathsf{Share}(s') \to ([s']_0, [s']_1)\}.$$

The secret can then be reconstructed by $\mathsf{Recon}([s]_0, [s]_1) = s$. When it is clear from context, we may omit the subscript and only denote the shares by $[s]$ when each party operates on their own share. We also denote the process when a party sends (and authenticates, in the malicious setting) their share to the other party to allow the later party to reconstruct the secret as *opening*.

In the semi-honest setting and $\mathcal{S} = \{0,1\}^\ell$, $\mathsf{Share}(s)$ simply uniformly samples $[s]_0, [s]_1 \in \{0,1\}^\ell$ conditioned on $[s]_0 \oplus [s]_1 = s$. The maliciously secure variant can be done using more complicated authenticated secret sharing [NNOB12, FKOS15].

4.2 Secret-shared PSI

Two-party PSI protocols can be constructed from various techniques resulting in different performance and properties [PRTY19, DPT20, CM20, MPR$^+$20, RT21, RS21, GPR$^+$21, CILO22, RR22, BPSY23]. In this work, we focus on PSI Payload variant, where each party's

$\mathcal{F}_{\mathsf{ssPSI}}$

Parameters. element set $\mathcal{U}$, payload set $\{0,1\}^\sigma$, set size m, output size $m' > m$

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, \tilde{A})$ from the sender where $\tilde{A} = \{(a_i, \tilde{a}_i)) : a_i \in \mathcal{U}, \tilde{a}_i \in \{0,1\}^\sigma\}_{i\in[m]}$, store $\tilde{A}$.

2. Upon receiving a message $(\mathsf{inputR}, \tilde{B})$ from the receiver where $\tilde{B} = \{(b_i, \tilde{b}_i)) : b_i \in \mathcal{U}, \tilde{b}_i \in \{0,1\}^\sigma\}_{i\in[m]}$, store $\tilde{B}$.

3. If both $\tilde{A}$ and $\tilde{B}$ are stored, compute $\pi = \mathsf{Reorder}(B)$ and

$$z_j = \begin{cases} (\tilde{a}_i \| \tilde{b}_{j'}) & \text{if } \exists a_i \in A, \text{ s.t. } a_i = b_j \\ 0^{2\sigma} & \text{otherwise} \end{cases}$$

   for $j' = \pi(j)$. Compute $\mathsf{Share}(z) \rightarrow ([z]_S, [z]_R)$. Send $[z]_S$ to the sender and $[z]_R$ to the receiver.

Figure 2: Ideal functionality for secret-shared PSI

input consists of two sets, an elements set for intersection and a set of values associated to the elements. The output of the protocol also contains the associated values of the elements in the intersection. These associated values are called "payloads." Most PSI protocols can be configured to transfer the payloads with differing efficiency [IKN+20, CILO22].

In our Approx-PSI construction, the output of PSI Payload should be secret shared between parties. Such protocols are often constructed using circuit-based PSI techniques such as the circuit-based variant of the PSI protocol in [RS21]. They call the variant circuit PSI. We simplify the variant to better serve our purpose in Figure 2. See Appendix B for the ideal functionality in [RS21].

Here, the intersection of m-element sets is mapped to a slightly larger set $m' > m$ (concretely $m' \approx 1.27m$ in [RS21]). In the original version each party also learns a secret shared bit indicating if each element is in the intersection, thus hiding even the intersection size. In our work, we only need the protocol to output the shares of the payloads, and not the actual PSI elements, in any order. Since the construction in [RS21] executes a garbled circuit in the last step, we simply modify the circuit to only output the payload parts.

The protocol in [RS21] is already quite efficient, and can be further improved using more recent oblivious key-value stores (OKVS) in [RR22, BPSY23] and VOLE setup [BCG+22] resulting in a high-performance protocol. The communication and computation cost of the secret-shared PSI when instantiated with the protocol above is linear in the number of elements $\mathcal{O}(\lambda n)$ [RS21].

$\mathcal{F}_{\mathsf{ssHamCom}}$

Parameters. element size ℓ, threshold Hamming distance d

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, [a]_S, [b]_S)$ from the sender where $[a]_S, [b]_S \in \{0,1\}^\ell$, store $([a]_S, [b]_S)$.
2. Upon receiving a message $(\mathsf{inputR}, [a]_R, [b]_R)$ from the receiver where $[a]_R, [b]_R \in \{0,1\}^\ell$, store $([a]_R, [b]_R)$.
3. If both $([a]_S, [b]_S)$ and $([a]_R, [b]_R)$ are stored, compute $a = \mathsf{Recon}([a]_S, [a]_R)$ and $b = \mathsf{Recon}([b]_S, [b]_R)$. Let $out = 1$ if $\mathcal{H}(a, b) \leq d$ and $out = 0$ otherwise. Send $[out]_S$ and $[out]_R$, secret shares of out to each party.

Figure 3: Ideal functionality for secret-shared Hamming distance comparison test

4.3 Secret-shared Hamming distance comparison test

Similar to the DA-PSI for Hamming distance in [CFR23], our Approx-PSI protocol utilizes a subprotocol for computing the Hamming distance. Unlike to one in [CFR23] that outputs both input binary strings to both parties when matched, our protocol takes secret shares of the strings as input, and outputs secret share of a single bit indicating whether the Hamming distance between the two inputs is within a certain threshold or not. We define the functionality $\mathcal{F}_{\mathsf{ssHamCom}}$ in Figure 3.

We note that the secret-share inputs of $\mathcal{F}_{\mathsf{ssHamCom}}$ can be added locally to obtain different inputs with the same Hamming distance. More specifically, $\mathcal{H}(a, b) = \mathcal{H}(a \oplus b)$ where $[a \oplus b]$ can be locally computed from $[a]$ and $[b]$ for additive secret sharing. Thus, we only need to construct one with secret shares output. However, we cannot simply convert the protocol in [CFR23] in the final step to output secret share as their immediate results reveal other party's input if they are matched.

The simplest way to realize this functionality is to through garbled circuit. However, the communication complexity of the resulting protocol will depend on the length ℓ. To obtain length-independent communication as in [CFR23], we consider a more complicated technique as follows.

As in the beginning of the protocol in [CFR23], we consider a binary vector of length ℓ as an ℓ-subset whose elements indicated by each bit of the vector. The Hamming distance can then be computed from the set difference. We use the idea from [GS19] that transforms a subset into a sparse polynomial and then evaluates the polynomial at various points to form a $d \times d$ matrix. The matrices has the property that the size of the corresponding set difference is below d if and only if the subtraction of the matrices is singular. This property can be checked securely using a secure determinant computation from [KMWF07]. As the protocol in [KMWF07] uses an additive homomorphic encryption. We can further modify it to output a secret share of the indicator result. The resulting protocol has communication complexity

$\tilde{\mathcal{O}}(d^2)$ and computation complexity of $\tilde{\mathcal{O}}(\ell + d^2)$ similar to the protocol in [CFR23]. We refer to Appendix C for the detailed construction. We note that for most concrete parameters, the simpler garbled circuit method yields better result as the constant in the big O for d^2 is quite large, and ℓ is generally not too large relative to d^2.

4.4 Secret-shared vector multiplication

The final functionality used in our work is the secret-shared vector multiplication described in Figure 4. Both parties hold secret shares of a bit c and a binary vector $\vec{v}$, and would like to compute the product $c\vec{v}$, and output as secret shares between two parties.

$\mathcal{F}_{\mathsf{ssVMult}}$

Parameters. element size ℓ

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, [c]_S, [\vec{v}]_S)$ from the sender, store $([c]_S, [\vec{v}]_S)$.
2. Upon receiving a message $(\mathsf{inputR}, [c]_R, [\vec{v}]_R)$ from the receiver, store $([c]_R, [\vec{v}]_R)$.
3. If both $([c]_S, [\vec{v}]_S)$ and $([c]_R, [\vec{v}]_R)$ are stored, compute $c = \mathsf{Recon}([c]_S, [c]_R)$ and $\vec{v} = \mathsf{Recon}([\vec{v}]_S, [\vec{v}]_R)$. If $c \in \{0,1\}$ and $\vec{v} \in \{0,1\}^\ell$, compute $\vec{out} = c\vec{v}$, and $\mathsf{Share}(\vec{out}) \to ([\vec{out}]_S, [\vec{out}]_R)$. Send $[\vec{out}]_S$ and $[\vec{out}]_R$ to the sender and the receiver, respectively.

Figure 4: Ideal functionality for Secret-Share Vector Multiplication

The functionality can be implemented using standard techniques for multiplication on secret shares such as using setup triples using OT or HE preprocessing in the semi-honest model, or using the standard frameworks for maliciously secure secret-shared operations such as TinyOT [NNOB12], Tinier [FKOS15] and MD-SPDZ [Kel20] in the malicious model. See Appendix D for a concrete protocol in the semi-honest model using OT. Using OT extension techniques, the (amortized) communication and computation is $\mathcal{O}(\ell)$ and $o(1)$, respectively, per multiplication.

5 Gaining intuition about the problem: false starts

The starting point of our constructions is the Approx-PSI protocol for Hamming distance. We begin by constructing an insecure version using the idea from [KOR98]. First, we randomly select a subset of position $I \subseteq [\ell]$ such that each $i \in [\ell]$ is chosen independently with probability p. We project every element of $A, B \subseteq \{0,1\}^\ell$ onto the position in I. We denote the projection of individual element $a \in \{0,1\}^\ell$ by $a_I = (a_{i_1}, a_{i_2}, \ldots, a_{i_{|I|}})$ for $I = \{i_1, \ldots, i_{|I|}\}$, and denote the sets of projections $A_I = \{a_I : a \in A\}$, and B_I defined similarly. We also denote the reverse map by $I_A^{-1}(c) = \{a \in A : a_I = c\}$ for $c \in \{0,1\}^{|I|}$. When it is clear from context, we may drop the set to $I^{-1}(a_I)$. For each $c \in A_I \cap B_I$, we can compute the probability that $\mathcal{H}(a,b) \leq d$ when $c = a_I = b_I$. By repeatedly sampling I, independently, and merging all pairs (a,b) from each projection, the probability that we fail to find any matched pairs is negligible.

We note that the goal of [KOR98] is to efficiently approximate the Hamming distance between vectors, while security is not their concern. This version of the protocol is thus not secure even when we use PSI to compute $A_I \cap B_I$. When a projected vector is in the intersection, it reveals that they share the same bits for the positions in I, even when they are not matched. We will fix this leakage in the final version of the protocol. Here we analyze the correctness.

For $i \in [k]$, where k is the number of repeats, the parties choose $I_i \subseteq [\ell]$ by choosing each position independently with probability p. Let $q = (1-p)^d$.

Lemma 5.1. *When $k \geq (\ln 2)\frac{\log n + \lambda}{q}$, the probability that there exists $a \in A, b \in B$ such that $\mathcal{H}(a,b) \leq d$ but $a_{I_i} \neq b_{I_i}$ for all $i \in [k]$, is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a,b) \leq d$. We have

$$\Pr[a_I \neq b_I] = 1 - \Pr[a_I = b_I] \leq 1 - (1-p)^d = 1 - q \leq e^{-q}.$$

Then

$$\Pr[a_{I_i} \neq b_{I_i} \forall i \in [k]] \leq e^{-kq},$$

and

$$\Pr[a_{I_i} \neq b_{I_i} \forall i \in [k], \exists (a,b) \in A \times B, \mathcal{H}(a,b) \leq d] \leq ne^{-kq}$$

by the union bound on A as each $a \in A$ has at most one b that is matched. When $kq \geq (\ln 2)(\log n + \lambda)$, we have negligible probability. □

This lemma guarantees that any matched pair will be found from at least one of the projections as long as the protocol repeats sufficiently many times, which is only $\mathcal{O}(\log n + \lambda)$ when q is constant.

Now we analyze the probability for non-matched pairs. Our goal is rule out as many non-matched pairs as possible to ensure that the number of remaining pairs to be checked is near-linear, and thus near-linear communication. To increase the probability that the pair a, b such that $\mathcal{H}(a,b) \geq td$ are not projected to the same vector, we first consider the case when $t = \log n$. In other words, for any $a \in A$ and $b \in B$, either $\mathcal{H}(a,b) \leq d$ or $\mathcal{H}(a,b) \geq td$ where $t = \log n$. We will ease this assumption in the later sections.

5.1 First attempt (that does not work for most parameters)

Here we show that under the condition $\lambda = \mathcal{O}(\log n)$, any pair (a,b) with $\mathcal{H}(a,b) \geq td$ will not be projected to the same element with high probability.

Lemma 5.2. *Assuming $\lambda = \mathcal{O}(\log n)$ and $\frac{1}{q} = 2^{\lambda/\log n + 2}$, the probability that there exists a, b such that $\mathcal{H}(a,b) \geq td$ and $a_{I_i} = b_{I_i}$ for some $i \in [k]$ is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a,b) \geq td$. We have

$$\Pr[a_I = b_I] \leq (1-p)^{td} = q^t.$$

When $t = \log n$, we have $q^t = q^{\log n} = 2^{\log q \log n} = n^{\log q}$. Then

$$\Pr[a_{I_i} = b_{I_i} \exists i \in [k]] \leq kn^{\log q},$$

and

$$\begin{aligned}\Pr[a_{I_i} = b_{I_i} \exists i \in [k], \exists (a,b) \in A \times B, \mathcal{H}(a,b) \geq td] &\leq n^2 kn^{\log q} \\ &= \frac{k}{n^{\log(1/q)-2}}.\end{aligned}$$

Assuming $\lambda = \mathcal{O}(\log n)$, we may choose $\frac{1}{q} = 2^{\frac{\lambda}{\log n}+2} = \mathcal{O}(1)$. Then $k = \mathcal{O}(\log n) = \mathcal{O}(\lambda)$, and the above probability is $\frac{k}{2^\lambda}$, which is negligible. □

In this case, by projecting A and B onto the coordinates in I_i for $i \in [k]$ with probability $p = 1 - q^{\frac{1}{d}} = 1 - 2^{-\frac{\frac{\lambda}{\log n}+2}{d}}$, and computing the intersection $A_I \cap B_I$, each party learns the elements that matched with another party's elements with overwhelming probability without additional direct comparison. We could construct a secure Approx-PSI protocol by merging the result of the intersection from each round.

The assumption $\lambda = \mathcal{O}(\log n)$ is probable in some cases as λ is a statistical security parameter. For example, we may choose $\lambda = 40$ and $n = 2^{20}$, which gives $q = \frac{1}{16}$. When $d = 4$, each position is chosen with probability 0.5. However, in the general case when λ is much larger than $\log n$, the number of rounds k will be exponential in $\frac{\lambda}{\log n}$, so is the communication from computing the intersections. Thus, we need a different method to separate the non-matched pairs.

In term of security, we note that when I is jointly chosen uniformly, and the intersection is computed using a PSI protocol, the resulting protocol is secure as the intermediate result C_j can be computed from the projection of each party's output.

5.2 Second attempt (that is too complicated to obtain security)

Now we consider a more complicated solution when $\lambda \gg \mathcal{O}(\log n)$. In this case, the projection alone cannot completely rule out the false positive, i.e., the case when $\mathcal{H}(a,b) \geq td$ but project to the same vector, while keeping the number of rounds (k) in poly-logarithmic. Each party needs to run a 2PC protocol to compare every pair of $a \in A$ and $B \in B$ (such as the one in [CFR23]) that project to the same c. Repeat this k (to be determined later) times with independently sampled I where λ is the security parameter.

Unfortunately, the above method may not give a linear number of comparisons as the number of possible pairs for each projection can be super-linear. To resolve this problem, the parties must select a "good" projection that only results in a linear number of comparisons. However, revealing the "good" projection also reveal the structure of the set. So, we construct a special-purpose PSI that outputs $\perp$ (privately as secret shares) when the projection produces too many collisions.

We define a "good" projection $I \subseteq [\ell]$ by I such that, for all $a \in A$, $|I^{-1}(a_I)| \leq \tau$ for a fixed constant $\tau \geq 2$. The PSI is modified to a special-purpose private *multiset* intersection that only outputs when I is good. Instead of having I as an additional input to PSI, both parties' input sets must be a multiset. That is a set where each element is associated with an integer representing the number of repeating of that element. In this case, the number associating to a_I is $|I^{-1}(a_I)|$. The protocol may only output $\perp$ when there exists a_I in the intersection whose $|I^{-1}(a_I)| \geq \tau$ for some constant τ.

Both parties will perform the special-purpose PSI on the projected sets for $k' \geq k$ rounds. We will show that for some choice of k', the number of "good" rounds is at least k.

Lemma 5.3. *For $k' = 8k$, $\tau = 2$ and $q = \frac{1}{4}$, the probability that there are less than k good projections is negligible.*

Proof. From the proof of the above lemma,

$$\Pr[I \text{ is not good}] = \Pr[|I^{-1}(a_I)| \geq \tau, \exists a \in A] \leq \frac{1}{\tau n^{\log(1/q)-2}}.$$

For $i \in [k']$, let X_i be an indicator that I_i chosen in round i is not good. Let $X = \sum_{i=1}^{k'} X_i$ Then

$$\mathbb{E}[X] = \sum_{i=1}^{k'} \mathbb{E}[X_i] \leq \frac{k'}{cn^{\log(1/q)-2}}.$$

Choose $\tau = 2$, $q = \frac{1}{4}$ and $k' = 8k$. Since X_i's are independent, by Chernoff bound, we have

$$\Pr[X \geq k' - k] \leq \Pr[X \geq \frac{7}{4} \cdot 4k] \leq e^{-\frac{(\frac{3}{4})^2 4k}{\frac{11}{4}}} \leq 2^{-k}$$

which is negligible. □

We could construct a secure Approx-PSI protocol by privately compare all pairs that projected to the same elements in the output of the special-purpose PSI in each round, and merging the results from all rounds as before.

For this solution, we need the special-purpose variant of PSI for multisets. While this idea is potentially doable theoretically, it may be difficult to construct efficiently. Moreover, the intermediate result, in particular, that of the bad projections, reveals some information that cannot be infer from the final result. Thus, the output may need to be secret shared, which further complicating the possible construction.

6 Approx-PSI Protocol

In this section, we show how to build on the previous two ideas to achieve an efficient and secure approximate PSI protocol. The independently repeating projection from the first idea already gives us the matches as long as we could keep the number of rounds small. The "good" or "bad" projection from the second idea, on the other hand, could be improved to make sure that the projection is not dropped entirely. So, we redefine the definition of the "bad" projection to be the union of the condition in the first idea and the condition for the second idea for both sets. In particular, for a pair (a, b) that is close, we call I bad for (a, b) if one of the following holds: (1) $a_I \neq b_I$ (2) $|I^{-1}(a_I)| \geq 2$ (3) $|I^{-1}(b_I)| \geq 2$. When a projection is bad for (a, b), a_I is dropped from A_I if (2) holds, or b_I is dropped from B_I if (3) holds. In the event of (1), they do not show up in the intersection anyway (unless as a different pair (a, b') or (a', b)). This allows other pairs to proceed and get discovered without dropping the whole projection as in the second idea.

For $i \in [k]$, where k will be determined later, we choose $I_i \subseteq [\ell]$ by choosing each position independently with probability p. Both parties project their sets to coordinates in I_i, denoted A_{I_i} and B_{I_i}, respectively. Let $q = (1-p)^d$. For $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$, we define

$$\mathsf{BAD}(a, b) = \{I \subseteq [\ell] : a_I \neq b_I \text{ or } |I_A^{-1}(a_I)| \geq 2 \text{ or } |I_B^{-1}(b_I)| \geq 2\}$$

We assume that for any $a \in A$ and $b \in B$, either $\mathcal{H}(a, b) \leq d$ or $\mathcal{H}(a, b) \geq td$.

Lemma 6.1. *When $k \approx \frac{(nt)^{\frac{1}{t-1}}(\lambda+\log n)}{1-\frac{1}{t}}$, the probability that there exists $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$ but $I_i \in \mathsf{BAD}(a, b)$ for all $i \in [k]$ is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$. We have

$$\Pr[a_I \neq b_I] = 1 - \Pr[a_I = b_I] \leq 1 - (1-p)^d = 1 - q.$$

Fix an element $a \in A$ that is projected to a_I. Let $X_{a'}$ be an indicator that $a'_I = a_I$. Then

$$\mathbb{E}[|I^{-1}(a_I)|] = \sum_{a' \in A} \mathbb{E}[X_{a'}] = n(1-p)^{td} = nq^t.$$

By the Markov's inequality,

$$\Pr[|I^{-1}(a_I)| \geq 2] \leq \frac{nq^t}{2}.$$

Similarly,

$$\Pr[|I^{-1}(b_I)| \geq 2] \leq \frac{nq^t}{2}.$$

By the Union bound, we have the probability that I is bad for (a, b) is at most

$$1 - q + nq^t.$$

Now we consider this probability as a function of q, $f(q) = 1 - q + nq^t$. When $t > 1$, the function takes the minimum value when $f'(q) = -1 + ntq^{t-1} = 0$. Solving the equation above gives $q = \frac{1}{(nt)^{\frac{1}{t-1}}}$. In this case, the above probability becomes

$$\alpha(n, t) = 1 - \frac{1}{(nt)^{\frac{1}{t-1}}} + \frac{n}{(nt)^{\frac{t}{t-1}}} = 1 - \frac{\beta(t)}{n^{\frac{1}{t-1}}}$$

where $\beta(t) = \frac{1}{t^{\frac{1}{t-1}}}\left(1 - \frac{1}{t}\right)$. Then the probability that I_i are bad for all $i \in [k]$ is at most $\alpha(n, t)^k$. Thus, the probability that for some close pair (a, b), all I_i's are bad is at most

$$n\alpha(n, t)^k = \frac{1}{2^\lambda}$$

when $k = \frac{\lambda + \log n}{\log(1/\alpha(n,t))}$. Using an approximation $\log(1-x) \approx -x$, we have

$$k \approx \frac{n^{\frac{1}{t-1}}(\lambda + \log n)}{\beta(t)} = \frac{(nt)^{\frac{1}{t-1}}(\lambda + \log n)}{1 - \frac{1}{t}}$$

□

We note that the second and the third condition of bad interval imply that the projections are set, not multiset, and the number of pairs in the intersection is at most n, which means the number of comparison is at most nk.

Now we analyze this result for different asymptotic cases of $t > 1$.

- When $t = \mathcal{O}(\log n)$:
  $\log(nt)^{\frac{1}{t-1}} = \frac{\log n + \log\log n}{\log n - 1} = \mathcal{O}(1)$. Thus, $k = \mathcal{O}(\lambda + \log n)$.
- When $t = \mathcal{O}(\log n / \log\log n)$:
  $\log(nt)^{\frac{1}{t-1}} = \frac{\log n + \log\log n - \log\log\log n}{\log n / \log\log n - 1} = \mathcal{O}(\log\log n)$.
  Thus, $k = \mathcal{O}((\log n)(\lambda + \log n))$.
- When $t = \mathcal{O}(\log\log n)$:
  $\log(nt)^{\frac{1}{t-1}} = \frac{\log n + \log\log\log n}{\log\log n - 1} = \mathcal{O}(\log n / \log\log n)$.
  Thus, $k = \mathcal{O}(n^{\frac{1}{\log\log n}}(\lambda + \log n)) = n^{o(1)}(\lambda + \log n)$.
- When $t = \mathcal{O}(1)$:
  $k = \mathcal{O}(n^{\frac{1}{t-1}}(\lambda + \log n))$.

Now, the remaining of this section, we construct a Approx-PSI protocol from a number of functionalities: the secret-shared PSI, the secret-shared Hamming distance comparison, and the secret-shared vector multiplication.

We obtain the following protocol assuming the above functionalities. The correctness of the protocol is from Lemma 6.1.

Now we show that this protocol is secure by constructing a simulator. By symmetry, suppose an adversary corrupting the receiver. For each $j \in [k]$, the simulator jointly samples I_j and compute the projections honestly. It simulates $\mathcal{F}_{\mathsf{ssPSI}}$ receiving $\tilde{B}_{I_j}$ from the adversary and returning shares of 0. The simulator uses $\tilde{B}_{I_j}$ to reconstruct B' and sends to $\mathcal{F}_{\mathsf{Approx-PSI}}$ and obtain the output $P \subseteq A \times B$. We may assume that the comparison is done after finishing all intersection first. The simulator simulates $\mathcal{F}_{\mathsf{ssHamCom}}$ by using P to compute *out* for each $b \in B'$ and simulates secure multiplication to create shares of correct output. We refer to Appendix E for more details.

6.1 Communication and Computation of the Approx-PSI Protocol

In this section, we analyze the performance of our Approx-PSI protocol. The protocol consists of one instance of $\mathcal{F}_{\mathsf{ssPSI}}$ in each of the k rounds. n' instance of $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$

```
Algorithm 1: Approx-PSI
Input  : Sets $A, B \subseteq \{0,1\}^\ell$, $|A| = |B| = n$
Output: $\{a \in A : \exists b \in B, \mathcal{H}(a,b) \le d\}$ and $\{b \in B : \exists a \in A, \mathcal{H}(a,b) \le d\}$
Each party replaces their input set by a representation of each cluster. We still
  denote their clustered inputs by $A$ and $B$ ;
for $j = 1$ to $k$ do
    The parties jointly sample $I_j \subseteq [\ell]$ such that each $i \in [\ell]$ has probability
      $p = 1 - \frac{1}{(nt)^{\frac{1}{d(t-1)}}}$ to be in $I_j$;
    The parties project every element in their sets into coordinates in $I_j$. If two
      elements have the same projection, remove both of them. The original element
      are attached to its projection as payload. The projection sets are denoted as
      $\tilde{A}_{I_j} = \{(a_{I_j}, a) : a \in A\}$ and $\tilde{B}_{I_j} = \{(b_{I_j}, b) : b \in B\}$;
    Each party sends $\tilde{A}_{I_j}$ and $\tilde{B}_{I_j}$ to $\mathcal{F}_{\mathsf{ssPSI}}$ and receives shares of the intersection
      $[z] \in (\{0,1\}^{2|I|})^{n'}$ ;
    foreach $i \in [n']$ do
        Each party sends shares $[z_i]$ to $\mathcal{F}_{\mathsf{ssHamCom}}$ and receives shares of $[out_i]$. ;
        Both parties send the shares of $[out_i]$ and $[z_i]$ to $\mathcal{F}_{\mathsf{ssVMult}}$, and obtains shares
          $[\tilde{z}_i]$;
    end
    Each party stores all shares of $[\tilde{z}_i]$ in $\tilde{Z}_j$ (separately as $\tilde{Z}_j^A$ and $\tilde{Z}_j^B$)
end
For each $j \in [k]$ and for each $[\tilde{z}] = ([a],[b]) \in \tilde{Z}_j$, open $[a]$ to the sender and $[b]$ to
  the receiver; let $A'_j$ and $B'_j$ denoted the opened values ;
The party computes $\{a : a \neq 0^\ell \in A'_j, \exists j \in [k]\}$ and $\{b : b \neq 0^\ell \in B'_j, \exists j \in [k]\}$, and
  outputs the elements in the set and their cluster in the original input set ;
```

in each of the k rounds. We instantiate the functionalities used to construct the Approx-PSI in Algorithm 1 as we discussed in Section 4, and compute theoretical communication complexity and computation complexity of the protocol.

The communication and computation of $\mathcal{F}_{\mathsf{ssPSI}}$ when instantiated with circuit PSI of [RS21], with or without later improvement, is $\mathcal{O}(n(\ell+\lambda))$. Here $n' = \mathcal{O}(n)$. The communication and computation of $\mathcal{F}_{\mathsf{ssHamCom}}$ instantiated using garbled circuit is $\mathcal{O}(\ell)$. The communication and computation of $\mathcal{F}_{\mathsf{ssVMult}}$ instantiated using OT as described in Appendix D are $\mathcal{O}(\ell)$ when amortized. Other subprotocols are simply sending data or local computations as shown in Table 3. We remark that replacing the secret-shared Hamming distance comparison test by ones with communication independent of ℓ does not improve the asymptotic complexity of the overall protocol.

Here, the number of rounds k depends on the gap t as proved in Lemma 6.1. We conclude the following corollary.

Corollary 6.2. *The protocol in Algorithm 1 when $\mathcal{F}_{SS-PSI}$, $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$ are instantiated as described above has the communication and com-*

Table 3: Communication and computation complexity of each subprotocol in Approx-PSI for Hamming distance.

Step	Subprotocol	Comm.	Comp.
1.	Clustering data	-	$\mathcal{O}(n\ell)$
2.	Repeat k times		
2.1	Sampling projections	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.2	Projecting vectors	-	$\mathcal{O}(n\ell)$
2.3	SS-PSI	$\mathcal{O}(n(\ell+\lambda))$	$\mathcal{O}(n(\ell+\lambda))$
2.4	Repeat $n' = \mathcal{O}(n)$ times		
2.4.1	SS Ham. comp. test	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.4.2	SS Vector Mult.	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.5	Opening share	$\mathcal{O}(n\ell)$	-
3.	Combining result	-	$\mathcal{O}(nk\ell)$
	Total	$\mathcal{O}(nk(\ell+\lambda))$	$\mathcal{O}(nk(\ell+\lambda))$

putation complexity $\mathcal{O}(\gamma(t)n^{1+\frac{1}{t-1}}(\lambda+\log n)(\ell+\lambda))$ *where* $\gamma(t) = \frac{t^{\frac{1}{t-1}}}{1-\frac{1}{t}}$.

When $t = \log n$, $\frac{\log n}{\log\log n}$ or $\log\log n$, the above communication is $\mathcal{O}(n(\lambda+\log n)(\ell+\lambda))$, $\mathcal{O}(n\,\mathrm{polylog}(n)(\lambda+\log n)(\ell+\lambda))$ or $n^{1+o(1)}(\lambda+\log n)(\ell+\lambda)$, respectively.

When $\log n = \mathcal{O}(\lambda)$ and $\ell = \mathcal{O}(\lambda)$, the above communication can be further simplified to $\mathcal{O}(n^{1+\frac{1}{t-1}}\lambda^2)$, $\mathcal{O}(n\lambda^2)$, $\mathcal{O}(n\,\mathrm{polylog}(n)\lambda^2)$ or $n^{1+o(1)}\lambda^2$, respectively. These assumptions are reasonable for most concrete parameters.

7 Other Distance Metrics

We can construct Approx-PSI for different distance metrics by embedding the set $(\mathcal{U}, \delta)$ into the set of binary strings equipped with the Hamming distance $(\{0,1\}^{\ell'}, \mathcal{H})$. We take advantage of the gap between the matched and non-matched pairs to remain so under the embedding. This method does not work for standard DA-PSI (gap $t = 1$) as the distance distortion caused by embedding could cause matched pairs to become non-matched pairs, or vice versa.

We consider three main distance metrics, namely, edit distance, Euclidean distance and angular distance. As we discussed in Chapter 2, the Euclidean distance implies cosine similarity, cosine distance and angular distance. However, embedding directly into the angular metric gives a better result, which we will use in our main application. We refer to Appendix F for more details.

8 Implementation and Optimization

In this section, we discuss implementations of our Approx-PSI protocol for Hamming distance and Approx-PSI protocol for image matching through angular distance.

8.1 Practical Parameters for Approx-PSI for Hamming Distance

In Section 6, we analyze the number of rounds as a function of security parameter, gap and the number of elements. The exact numbers are shown in Appendix G.

When the gap is $t = \log n$, the protocol needs to run for about 80 rounds to ensure that the probability that protocol would fail is at most 2^{-40}. This is the number of times the underlying PSI protocol is executed. For the smallest possible gap $t = 3$, the number of rounds approximately double whenever the input sizes quadruple. Assuming the underlying PSI is linear time, then for this fixed $t = 3$, the Approx-PSI protocol is $\mathcal{O}(n^{1.5})$.

8.2 Practical Parameters for Approx-PSI for Angular Distance

As our main application is matching images using Approx-PSI for angular distance, we describe here the practical parameters for the setting. We use the embedding in Appendix F.3 to embed a real (unit) vector with angular distance metric as a binary vector with Hamming distance metric.

Since embedding real vectors with angular distance into binary vectors with Hamming distance incurs some error δ, and the size of the binary vectors is proportional to δ^{-2}, we would like to set the error to be just small enough that the protocol is correct, but not too small as that will increase the running time of the Approx-PSI protocol.

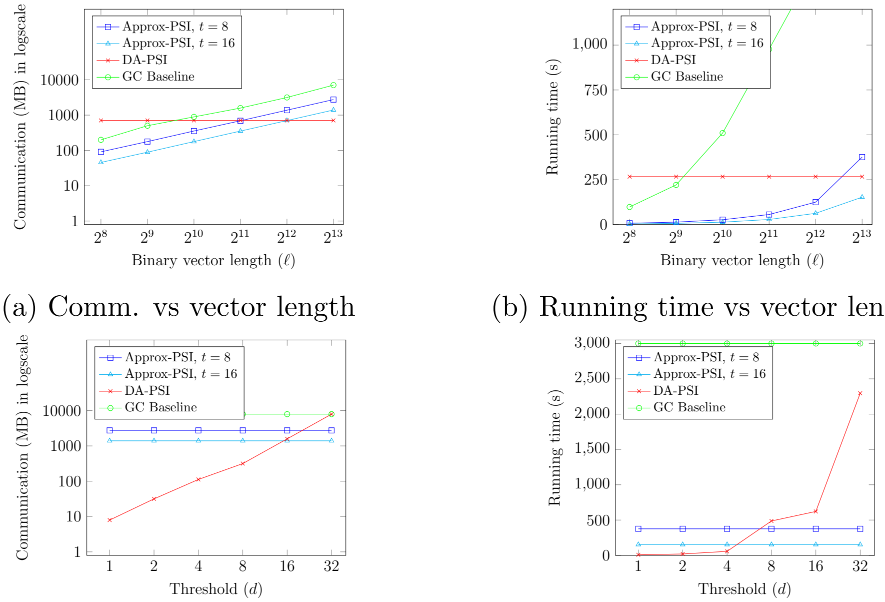
We denote $0 < t_1 < t_2$ be the threshold for match and non-match for the angular distance, respectively. We note that the angular distance of two uniformly sampled real vectors is closer to 0.5 as their size get larger. For example, two random 4096-dimensional vectors have angular distance between 0.45-0.55 with overwhelming probability. Thus, in no circumstance we should set t_2 larger than 0.45.

Using the analysis in Appendix F.3, one example of the parameters is $t_1 = 0.0225$, $t_2 = 0.4$, $\delta = 0.05$ for the angular distance. This results in binary vector length of $\ell = 2^{13} \approx 8000$, match threshold in Hamming distance of $d = 512$ and gap $t = 6$.

8.3 Approx-PSI for Similar Image Matching

In this section, we describe how to use our Approx-PSI to match similar images. We consider an image as a matrix of numbers, one for each pixel. A natural way to compare the images is comparing the distance of their matrices either by Euclidean or angular metrics. However, some change in the pixels that may be not alter the image, i.e., almost indistinguishable to human eyes, may cause the distance to be sufficiently far apart. Thus, we consider image transformation methods that map images to vectors in a way that standard image alterations, such as resizing, blurring or adjusting brightness, barely change these vectors.

We first consider the spread spectrum in [CKLS97]. Here the matrix of image pixels is transformed into another matrix via discrete cosine transform (DCT). The resulting matrix has its significant values near the upper left corner. These values represent the structure of the image in that they resist many image alterations including the ones mentioned above. As in [CKLS97], our protocol computes the DCT of an image, and only extract a submatrix from the upper left corner as the image representation. In [CKLS97], this representation is used to add watermark back to the image.



(a) Comm. vs vector length

(b) Running time vs vector length

(c) Comm. vs threshold

(d) Running time vs threshold

Figure 5: Running Time in seconds and Communication in logscale of MB of Approx-PSI, DA-PSI and garbled circuit baseline for set size $n = 100$, and (a) - (b) fixed threshold $d = 6$; (c) - (d) fixed element size $\ell = 8192$. Our Approx-PSI uses gap $t = 8, 16$ and security parameter $\lambda = 5$ to match the error rate of 0.05.

However, using the DCT submatrix (as a vector) do not solve our problem due to its structure mentioned above. For any image, its DCT tends to has larger numbers (in magnitude, could be positive or negative) in the upper left positions. This results in the vectors representing two different images do not have enough angular distance required for our Approx-PSI. We solve this problem by re-scaling the values with respect to their positions. In particular, for $m \times m$ submatrix, we divide the (i, j) position of the matrix by $1 + i + j$ for $0 \leq i, j \leq m - 1$. Experimentally, this modification makes most pairs of the vectors representing two different images having the angular distance at least 0.4. When an image is altered by resizing, blurring or adjusting brightness and contrast, the modified DCT resulting in angular distance of within 0.02 for most images, depending on the magnitude of alterations.

8.4 Implementation and Performance

We implement our Approx-PSI in C++ using EMP-Toolkit[1] for communications, OTs (for secret sharing operations) and garbled circuits. We use volepsi[2] for the underlying OPPRF protocol in modified circuit PSI in [RS21]. We use opencv[3] for image processing, image alterations and DCT transformation. All of our implementations are singled-threaded.

[1] https://github.com/emp-toolkit

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

[3] https://github.com/opencv/opencv

First, we compare our Approx-PSI with the DA-PSI for the Hamming distance by matching their error rate of 0.05, which translated to our security parameter of $\lambda = 5$, and a garbled circuit baseline. Since we do not have the code for the DA-PSI, we try to match their resources as much as possible, and take their numbers directly from [CFR23]. Thus, the comparison is only roughly estimated. From Figure 5 (a) and (b), our protocol outperforms DA-PSI in both communication and running time for short binary vectors, up to $\ell = 2048$ for communication and up to $\ell = 4096$ for running time, for the gap $t = 8$ without parallel computation. Even when $\ell = 1024$, our protocol is 10-20 times faster depending on the gap. From Figure 5 (c) and (d), even for large element size $\ell = 8192$, our protocol also outperforms DA-PSI when matching threshold are above 4 bits for running time and 16 bits for communication, which are minuscule relative to the total length of the vectors. As an example, in our image matching application, the threshold $d = 512$ is used for this element size.

Table 4: Communication and Running time of Approx-PSI for Hamming distance with element size $\ell = 128$, threshold $d = 4$, gap $t = \log n$ and security parameter $\lambda = 40$ for various set size $n = 256, 1024, 4096$, resulting in number of rounds $k = 96, 89, 86$, respectively.

Step	communication (MB)			running time (s)		
n	256	1024	4096	256	1024	4096
Projection	0.01	0.01	0.01	0.004	1.45	4.859
SS-PSI	214.57	848.65	3279.3	15.13	59.28	226.78
SS Ham. comp.	243.58	902.85	3483.4	22.54	83.58	324.9
SS Vector Mult.	4.52	16.71	64.41	0.659	2.285	8.652
Open & output	3	11.12	42.92	0.365	1.256	4.71
Total	465.68	1779.3	6870	38.7	147.85	569.9

Second, we demonstrate the performance of our Approx-PSI for the Hamming distance in various parameters. Table 4 shows the communication and running time of our Approx-PSI protocol when input size increase, breaking down by main steps, for much larger security parameter of $\lambda = 40$, and the gap $t = \log n$. Both communication and running time of our protocol are near linear in the input size. Thus, it is scaled better when the input sets are large, compared to the quadratic complexity of the previous works. We note that the secret-shared Hamming distance comparison test step dominating both communication and running time, followed by the secret-shared PSI. Thus, they are the main targets for further improvement. While it is not directly comparable due to different distance metrics, the sa-PSI protocol in [GRS23] communicates around 30-100 GB for $n = 2700$ balls under different conditions. Thus, our

Finally, we demonstrate the performance of our Approx-PSI for matching images using the modified DCT and the angular distance metric in Table 5. The transformation steps from images to modified DCTs and then to binary vectors are much faster than the Approx-PSI itself. The first transformation requires no interaction, while the second only requires the parties to agree on a PRG seed to generate random hyperplanes. Thus, the communication of the transformation steps is negligible.

Table 5: Communication and Running time of Approx-PSI for image matching with modified DCT of size $32 \times 32 = 1024$, $\ell = 8192$, threshold $d = 512$, gap $t = 6$ and security parameter $\lambda = 40$ for various set size $n = 32, 64, 128$, resulting in number of rounds $k = 91, 110, 131$, respectively.

Step	communication (GB)			running time (s)		
n	32	64	128	32	64	128
Images to DCT vectors	-	-	-	0.16	0.32	0.64
DCT vectors to binary vectors	-	-	-	1.95	3.23	5.77
Approx-PSI	1.91	4.64	11.05	322	768	1842
Total	1.91	4.64	11.05	324	772	1849

References

[BCG+19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent ot extension and more. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39*, pages 489–518. Springer, 2019.

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In *Annual International Cryptology Conference*, pages 603–633. Springer, 2022.

[BPSY23] Alexander Bienstock, Sarvar Patel, Joon Young Seo, and Kevin Yeo. Near-Optimal oblivious Key-Value stores for efficient PSI, PSU and Volume-Hiding Multi-Maps. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 301–318, Anaheim, CA, August 2023. USENIX Association.

[CFR23] Anrin Chakraborti, Giulia Fanti, and Michael K Reiter. {Distance-Aware} private set intersection. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 319–336, 2023.

[CILO22] Wutichai Chongchitmate, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Psi from ring-ole. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 531–545, 2022.

[CKLS97] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

[CM20] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious prf. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 34–63, Cham, 2020. Springer International Publishing.

[CSF$^+$07] M Patrick Collins, Timothy J Shimeall, Sidney Faber, Jeff Janies, Rhiannon Weaver, Markus De Shon, and Joseph Kadane. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 93–104, 2007.

[Dau09] John Daugman. How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier, 2009.

[DM21] Sjoerd Dirksen and Shahar Mendelson. Non-gaussian hyperplane tessellations and robust one-bit compressed sensing. *Journal of the European Mathematical Society*, 23(9):2913–2947, 2021.

[DMS22] Sjoerd Dirksen, Shahar Mendelson, and Alexander Stollenwerk. Sharp estimates on random hyperplane tessellations. *SIAM Journal on Mathematics of Data Science*, 4(4):1396–1419, 2022.

[DPT20] Thai Duong, Duong Hieu Phan, and Ni Trieu. Catalic: Delegated psi cardinality with applications to contact tracing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 870–899. Springer, 2020.

[DS18] Sjoerd Dirksen and Alexander Stollenwerk. Fast binary embeddings with gaussian circulant matrices: improved bounds. *Discrete & Computational Geometry*, 60:599–626, 2018.

[DS20] Sjoerd Dirksen and Alexander Stollenwerk. Binarized johnson-lindenstrauss embeddings. *arXiv preprint arXiv:2009.08320*, 2020.

[FKOS15] Tore Kasper Frederiksen, Marcel Keller, Emmanuela Orsini, and Peter Scholl. A unified approach to mpc with preprocessing using ot. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 711–735. Springer, 2015.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GPR$^+$21] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Oblivious key-value stores and amplification for private set intersection. In *Annual International Cryptology Conference*, pages 395–425. Springer, 2021.

[GRS22] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *Annual International Cryptology Conference*, pages 323–352. Springer, 2022.

[GRS23] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In *Annual International Cryptology Conference*, pages 577–610. Springer, 2023.

[GS19] Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In *Annual International Cryptology Conference*, pages 3–29. Springer, 2019.

[HEKM11] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure {Two-Party} computation using garbled circuits. In *20th USENIX Security Symposium (USENIX Security 11)*, 2011.

[HS20] Thang Huynh and Rayan Saab. Fast binary embeddings and quantized compressed sensing with structured matrices. *Communications on Pure and Applied Mathematics*, 73(1):110–149, 2020.

[IKN+20] 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 *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 370–389. IEEE, 2020.

[JL84] William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 01 1984.

[Kel20] Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 1575–1590, 2020.

[KMWF07] Eike Kiltz, Payman Mohassel, Enav Weinreb, and Matthew Franklin. Secure linear algebra using linearly recurrent sequences. In *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings 4*, pages 291–310. Springer, 2007.

[KOR98] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 614–623, 1998.

[MKHSO17] Mina Mohammadi-Kambs, Kathrin Hölz, Mark M Somoza, and Albrecht Ott. Hamming distance as a concept in dna molecular recognition. *ACS omega*, 2(4):1302–1308, 2017.

[MPDC19] Luca Melis, Apostolos Pyrgelis, and Emiliano De Cristofaro. On collaborative predictive blacklisting. *ACM SIGCOMM Computer Communication Review*, 48(5):9–20, 2019.

[MPR+20] Peihan Miao, Sarvar Patel, Mariana Raykova, Karn Seth, and Moti Yung. Two-sided malicious security for private intersection-sum with cardinality. In *Annual International Cryptology Conference*, pages 3–33. Springer, 2020.

[NNOB12] Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In *Annual Cryptology Conference*, pages 681–700. Springer, 2012.

[OPJM10] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. Scifi-a system for secure face identification. In *2010 IEEE Symposium on Security and Privacy*, pages 239–254. IEEE, 2010.

[OR07] Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *Journal of the ACM (JACM)*, 54(5):23–es, 2007.

[OR15] Samet Oymak and Ben Recht. Near-optimal bounds for binary embeddings of arbitrary sets. *arXiv preprint arXiv:1512.04433*, 2015.

[PRTY19] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Spot-light: Lightweight private set intersection from sparse ot extension. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 401–431, Cham, 2019. Springer International Publishing.

[PV14] Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

[RR22] Srinivasan Raghuraman and Peter Rindal. Blazing fast psi from improved okvs and subfield vole. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2505–2517, 2022.

[RS21] Peter Rindal and Phillipp Schoppmann. Vole-psi: Fast oprf and circuit-psi from vector-ole. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 901–930. Springer, 2021.

[RT21] Mike Rosulek and Ni Trieu. Compact and malicious private set intersection for small sets. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1166–1181, 2021.

[Sch18] Peter Scholl. Extending oblivious transfer with low communication via key-homomorphic prfs. In *Public-Key Cryptography–PKC 2018: 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I 21*, pages 554–583. Springer, 2018.

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

[WACL10] Andrew G West, Adam J Aviv, Jian Chang, and Insup Lee. Spam mitigation using spatio-temporal reputations from blacklist history. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 161–170, 2010.

[WHZ+15] Xiao Shaun Wang, Yan Huang, Yongan Zhao, Haixu Tang, XiaoFeng Wang, and Diyue Bu. Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 492–503, 2015.

[YCP15] Xinyang Yi, Constantine Caramanis, and Eric Price. Binary embedding: Fundamental limits and fast algorithm. In *International Conference on Machine Learning*, pages 2162–2170. PMLR, 2015.

A Euclidean distance, Cosine Similarity and Angular distance

Here we give formulas for the distances in Euclidean space, and their relationship. For any $x, y \in \mathbb{R}^N$, with $x = (x_1, \dots, x_N)$ and $y = (y_1, \dots, y_N)$, we have

- **Euclidean distance**:
$$\|x-y\|_2 = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2};$$
- **cosine distance**:
$$\delta_{\cos}(x,y) = 1 - \frac{x \cdot y}{\|x\|_2\|y\|_2};$$
We note that $1 - \delta_{\cos}(x,y) = \dfrac{x \cdot y}{\|x\|_2\|y\|_2}$ is called the *cosine similarity* between x and y. In numerous analytical and computational contexts, cosine similarity serves as a prevalent metric to determine the degree of similarity or alignment between two data sets.
- **angular distance**:
$$\delta_\theta(x,y) = \frac{\arccos\left(\frac{x \cdot y}{\|x\|_2\|y\|_2}\right)}{\pi}.$$
When x, y are unit vectors, i.e., in the unit sphere S^{N-1}, this distance is also called *geodesic* distance as it is the length of the shortest path on the sphere connecting x and y.

The cosine distance has values between 0 and 2 inclusive while the angular distance has values between 0 and 1 inclusive. Clearly,

$$\delta_\theta(x,y) = \frac{\arccos(1 - \delta_{\cos}(x,y))}{\pi},$$

and

$$\begin{aligned}\|x-y\|_2^2 &= \|x\|_2^2 + \|y\|_2^2 - 2(x \cdot y)\\ &= \|x\|_2^2 + \|y\|_2^2 - 2\|x\|_2\|y\|_2(1 - \delta_{\cos}(x,y)).\end{aligned}$$

The cosine distance and the angular distance do not concern the length of x, y when they are nonzero vectors. In this case, we may assume that $x, y \in S^{N-1}$, a unit sphere in $\mathbb{R}^N$. Under this condition,

$$\|x - y\|_2 = \sqrt{2\delta_{\cos}(x, y)}.$$

Thus, secure computation of the Euclidean distance implies secure computation of the cosine distance and cosine similarity as well.

B Circuit PSI

We describe the ideal functionality for circuit PSI from [RS21] in Figure 6.

$\mathcal{F}_{\mathsf{cPSI}}$

Parameters. element set $\mathcal{U}$, payload set $\{0,1\}^\sigma$, set size m, a map
$\mathsf{Reorder} : \mathcal{U}^m \to \{\pi : [m] \to [m'], \text{injective}\}$ with $m' > m$

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, A, \tilde{A})$ from the sender where $A = \{a_1, \ldots, a_m\} \subseteq \mathcal{U}$ and $\tilde{A} = \{\tilde{a}_1, \ldots, \tilde{a}_m\} \in \{0,1\}^\sigma\}$, store $(A, \tilde{A})$.
2. Upon receiving a message $(\mathsf{inputR}, B, \tilde{B})$ from the receiver where $B = \{b_1, \ldots, b_m\} \subseteq \mathcal{U}$ and $\tilde{B} = \{\tilde{b}_1, \ldots, \tilde{b}_m\} \in \{0,1\}^\sigma\}$, store $(B, \tilde{B})$.
3. If both $(A, \tilde{A})$ and $(B, \tilde{B})$ are stored, compute $\pi = \mathsf{Reorder}(B)$, and uniformly samples $c^0, c^1 \leftarrow \{0,1\}^{m'}$ and $z^0, z^1 \leftarrow (\{0,1\}^{2\sigma})^{m'}$ conditioned on

$$\begin{cases} c^0_{j'} \oplus c^1_{j'} = 1, z^0_{j'} \oplus z^1_{j'} = (\tilde{a}_{j'} \| \tilde{b}_{j'}) & \text{if } \exists a_i \in A \text{ s.t. } a_i = b_j \\ c^0_{j'} \oplus c^1_{j'} = 0, z^0_{j'} \oplus z^1_{j'} = 0^{2\sigma} & \text{otherwise} \end{cases}$$

for $j' = \pi(j)$. Send c^0, z^0 to the sender and c^1, z^1, π to the receiver.

Figure 6: Ideal functionality for circuit PSI [RS21]

C Length-independent Secret-Shared Hamming Distance Comparison

We give more details on the protocol realizing $\mathcal{F}_{\mathsf{ssHamCom}}$ which has communication complexity $\mathcal{O}(\lambda d^2)$, independent of the length of an element. The protocol combines the ideas from three different protocols from [CFR23, GS19, KMWF07].

We obtain the protocol in Algorithm 2. Let $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an additive homomorphic encryption. Let p be a prime integer such that $p > 2\ell$ and $p > (4d^2 + 2d)2^\lambda$.

```
Algorithm 2: Secret-Shared Hamming Distance Comparison Test
  Input  : $a, b \subseteq \{0,1\}^\ell$
  Output: $[out]_S, [out]_R \in \{0,1\}$ where $out = [out]_S \oplus [out]_R = 1$ if $\mathcal{H}(a,b) \leq d$ and
          $out = 0$ otherwise
1 Parties compute $P_a = \sum_{i \in \{0,1,\ldots,\ell-1\}} x^{2i+a_i}$ and $P_b$ defined similarly;
2 Parties jointly sample $u \leftarrow \mathbb{F}_p$;
3 Parties compute an $(2d+1) \times (2d+1)$ matrix
      $$H_a = \begin{bmatrix} P_a(u^0) & \cdots & P_a(u^{2d}) \\ \vdots & & \vdots \\ P_a(u^{2d}) & \cdots & P_a(u^{4d}) \end{bmatrix}$$
    and $H_b$ defined similarly;
4 Sender generates $\mathsf{KeyGen} \rightarrow (pk, sk)$ and sends $(pk, \mathbf{H}_a = \mathsf{Enc}_{pk}(H_a))$ to Receiver;
5 Receiver computes $\mathbf{H} = \mathbf{H}_a - \mathsf{Enc}_{pk}(H_b)$ (denote $H = H_a - H_b$), samples
    $\vec{u}, \vec{v} \leftarrow \mathbb{F}_p^{2d+1}$;
6 Parties interactively compute $\mathbf{H}^k \vec{v}$ for $k = 1, \ldots, 4d+1$;
7 Receiver computes $\vec{u}^T \mathbf{H}^k \vec{v}$ for $k = 1, \ldots, d$, and $\mathbf{M}_H$, and encryption of the minimal
    polynomial $m_H$ of $H$.;
8 Parties evaluate garbled circuit that decrypts and secret shares an indicator that the
    constant term of $\mathbf{M}_H$ is zero.
```

The correctness of the protocol follows from the fact analyzed in [GS19] that $\det(H_a - H_b) = 0$ if and only if $\mathcal{H}(a,b) \leq d$. We utilize the homomorphic encryption method in [KMWF07] to compute the determinant from the minimal polynomial of the matrix, which can be computed from $\vec{u}^T H^k \vec{v}$.

From the analysis in [KMWF07], the protocol above has communication and computation complexity of $\mathcal{O}(\lambda d^2 \mathsf{polylog}\, d)$. When adding the local transformation in the first step, the computation complexity is $\mathcal{O}(\lambda(\ell + d^2 \mathsf{polylog}\, d))$.

D Secret-Shared Scalar-Vector Multiplication from OT

Using OT, we can easily realize $\mathcal{F}_{\mathsf{ssVMult}}$ in the semi-honest model. Let $\mathcal{F}_{\mathsf{OT}}$ be the 1-out-of-2 OT functionality.

As $w_S = r_R \oplus ([c]_S \cdot [v]_R)$, we have $[out]_S = ([c]_S \cdot [v]_S) \oplus ([c]_S \cdot [v]_R) = [c]_S \cdot ([v]_S \oplus [v]_R)$. Similarly, $[out]_R = [c]_R \cdot ([v]_S \oplus [v]_R)$. Using OT extension techniques, the (amortized) communication and computation can be reduced to $o(1)$ [Sch18, BCG$^+$19].

Algorithm 3: Secret-Shared Vector Multiplication

```
Input  : $[v]_S, [v]_R \subseteq \{0,1\}^\ell$, $[c]_S, [c]_R \in \{0,1\}$
Output: $[out]_S, [out]_R \in \{0,1\}^\ell$ where
        $out = [out]_S \oplus [out]_R = ([c]_S \oplus [c]_R) \cdot ([v]_S \oplus [v]_R)$
Sender samples $r_S \leftarrow \{0,1\}^\ell$ and sends $(\mathsf{inputS}, r_S, r_S \oplus [v]_S)$ to $\mathcal{F}_{\mathsf{OT}}$. Receiver sends
  $(\mathsf{inputR}, [c]_R)$ to $\mathcal{F}_{\mathsf{OT}}$ and receives $w_R$;
Receiver samples $r_R \leftarrow \{0,1\}^\ell$ and sends $(\mathsf{inputS}, r_R, r_R \oplus [v]_R)$ to $\mathcal{F}_{\mathsf{OT}}$. Sender sends
  $(\mathsf{inputR}, [c]_S)$ to $\mathcal{F}_{\mathsf{OT}}$ and receives $w_S$;
Sender outputs $[out]_S = ([c]_S \cdot [v]_S) \oplus r_S \oplus w_S$, and Receiver outputs
  $[out]_R = ([c]_R \cdot [v]_R) \oplus r_R \oplus w_R$.
```

E Proof of Security for Approx-PSI for Hamming distance

Here we proof the security of our main protocol.

Theorem E.1. *The protocol in Algorithm 1 is secure in the* $\mathcal{F}_{\mathsf{ssPSI}}$*,* $\mathcal{F}_{\mathsf{ssHamCom}}$ *and* $\mathcal{F}_{\mathsf{ssVMult}}$ *hybrid model.*

Proof. By symmetry, it suffices to construct a simulator $\mathcal{S}$ for the case when an adversary corrupting the receiver. For each $j \in [k]$, $\mathcal{S}$ follows the protocol to jointly sample I. It simulates $\mathcal{F}_{\mathsf{ssPSI}}$ to learn $\tilde{B}_I$ and outputs a secret share of $0^{2|I|n'}$, instead of z, to $\mathcal{S}$. $\mathcal{S}$ stores $\tilde{B}_I$. It also simulates $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$, and outputs a random secret share of 0 and $0^{2|I|n'}$, instead of out and $\tilde{z}$, to $\mathcal{S}$, respectively. After k rounds, $\mathcal{S}$ uses the stored $\tilde{B}_I$'s to reconstruct the receiver's set B^*. It sends B^* to $\mathcal{F}_{\mathsf{Approx-PSI}}$ to learn the set of Hamming close pairs. Finally, $\mathcal{S}$ computes openings for each $[\tilde{z}] \in \tilde{Z}_I$'s that gives the Hamming close pairs for each I.

We prove the indistinguishability through the following hybrids:

- H_0: This is the real world interaction.
- H_1: Same as H_0 except $\mathcal{S}$ simulates the functionalities honestly. This hybrid is identical to H_0.
- H_2: Same as H_1 except $\mathcal{S}$ outputs shares of $0^{2|I|n'}$ instead of the correct output of $\mathcal{F}_{\mathsf{ssPSI}}$. It then replaces the adversary's input for $\mathcal{F}_{\mathsf{ssHamCom}}$ with the correct one from the adversary's input to $\mathcal{F}_{\mathsf{ssPSI}}$. This hybrid is identical to H_1 as single shares of $0^{2|I|n'}$ and z are identically distributed.
- H_3: Same as H_2 except $\mathcal{S}$ outputs shares of 0 instead of the correct output of $\mathcal{F}_{SS-Ham-Compare}$. It then replaces the adversary's input for $\mathcal{F}_{\mathsf{ssVMult}}$ with the correct shares of the output of $\mathcal{F}_{\mathsf{ssHamCom}}$. This hybrid is identical to H_2 as a single share of 0 and out_i are identically distributed.
- H_4: Same as H_3 except $\mathcal{S}$ outputs random shares instead of the correct output of $\mathcal{F}_{\mathsf{ssVMult}}$. When $\mathcal{S}$ opens the shares in the final step, it opens to the correct outputs of $\mathcal{F}_{\mathsf{ssVMult}}$. This hybrid is identical to H_3 as each share of $\tilde{c}'$'s is uniformly random.

- H_5: Same as H_4 except $\mathcal{S}$ uses $\tilde{B}_I$ to reconstruct B^* from the payload and fill the rest with the special element $\perp$. It uses B^* to compute outputs in each step instead of B. Note that B^* may be smaller than B when there is an element that always collides with others when projected to coordinates in I in every round. We show that such elements occurs with negligible probability.

Claim. *Except with negligible probability, $B^* = B$.*

Proof. Clearly, $B^* \subseteq B$. We need to show that except with negligible probability, every element of B appears in B^*. Note that $b \in B$ does not appear in B^* only when its projection collides with another element in every round. In each round, the probability of such event is at most $\frac{nq^t}{2}$ by the proof of Lemma 6.1. Thus, the probability that $B^* \neq B$ is at most $\left(\frac{nq^t}{2}\right)^k$ which is negligible for the choice of k in the lemma.

□

- H_6: Same as H_5 except $\mathcal{S}$ sends B^* to $\mathcal{F}_{\mathsf{Approx-PSI}}$ and no longer interact with the sender. It uses B^* to compute openings for each $\tilde{C}_I$'s.

□

F Other Distance Metrics

Here we give more details about the embedding from three other distance metric into Hamming distance metric.

F.1 Edit Distance

Our protocol relies on the low distortion embedding by Ostrovsky and Rabani [OR07].

Theorem F.1 ([OR07])**.** *There exists a polynomial time algorithm ϕ that for every $\delta > 0$, $\phi = \phi(\cdot, \ell, \delta) : \{0,1\}^\ell \to \{0,1\}^{\ell'}$ such that $\ell' = \mathcal{O}(\ell^2 \log(\ell/\delta))$ satisfying for any $x, y \in \{0,1\}^\ell$*

$$2^{-\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\mathrm{ed}(x,y) \leq \mathcal{H}(\phi(x), \phi(y)) \leq 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\mathrm{ed}(x,y)$$

with probability at least $1 - \delta$.

We note that asymptotically $\log^M \ell < 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})} < \ell^\epsilon$ for any large constant M and small constant $\epsilon > 0$. Applying the embedding and Approx-PSI for Hamming distance gives the following corollary.

Corollary F.2. *There exists a Approx-PSI for edit distance with gap $t' = 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t$ with communication and computation $\mathcal{O}(n^{1+\frac{1}{t-1}}\ell^2(\log n + \lambda)^2)$.*

Proof. Let d, t be the threshold and the gap of the underlying Approx-PSI for Hamming distance, respectively. By Theorem F.1, for any $a \in A$ and $b \in B$ such that $\mathrm{ed}(a,b) \leq d'$,

$$\mathcal{H}(\phi(a), \phi(b)) \leq 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\mathrm{ed}(a,b) \leq 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}d'.$$

For any $a \in A$ and $b \in B$ such that $\text{ed}(a,b) \geq t'd'$,

$$\mathcal{H}(\phi(a),\phi(b)) \geq 2^{-\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\text{ed}(a,b) \geq 2^{-\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t'd'.$$

Setting the right hand side of each inequality as d and td, respectively, gives $t' = 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t$. Here we set $\delta = \frac{1}{n^2 2^\lambda}$. Thus, $\ell' = \mathcal{O}(\ell^2(\log n + \lambda))$.

The communication and computation of the resulting protocol is that of Approx-PSI for Hamming distance where element size is $\ell' = \mathcal{O}(\ell^2(\log n + \lambda))$. □

F.2 Euclidean Distance

Similar to the Edit distance, there are embedding from the Euclidean distance to the Hamming distance [PV14, HS20, DS20, DM21, DMS22]. Unlike the embedding for the edit distance, which is complicated, the embeddings for Euclidean following the simple ideas from the Johnson-Lindenstrauss lemma [JL84]. The original lemma concerns the dimension reduction of vectors in $\mathbb{R}^N$. However, the technique can be used to constructed an embedding into binary strings, represented by $\{-1,1\}$ instead of $\{0,1\}$.

A hyperplane in $\mathbb{R}^N$ is chosen randomly to cut $\mathbb{R}^N$ into two halves. Vectors in one half is mapped to -1 while the other half is mapped to 1. This can be computed by the sign of inner product between the vectors and the normal vector of the hyperplane. The process is repeated multiple times with independently chosen hyperplanes to obtain a binary vector. The idea has been improved with better methods of chosing the hyperplanes and the analysis of the resulting distortion. Here we choose the most recent results for our construction.

Theorem F.3 ([DM21])**.** *There exists a polynomial time algorithm ϕ that for every $0 < \rho < R$ and $T \subseteq \mathbf{B}(R)$ with $|T| = n$ where $\mathbf{B}(R)$ is a Euclidean ball of radius R, $\phi : \mathbb{R}^N \to \{0,1\}^\ell$ such that $\ell = \mathcal{O}(\frac{R\log(eR/\rho)\log n}{\rho^3})$, satisfying for any $x, y \in T$ such that $\|x-y\|_2 \geq \rho$*

$$\mathcal{O}(\frac{\ell}{R})\|x-y\|_2 \leq \mathcal{H}(\phi(x),\phi(y)) \leq \mathcal{O}(\frac{\ell\sqrt{\log(eR/\rho)}}{R})\|x-y\|_2$$

with probability at least $1 - e^{-\mathcal{O}(\ell\rho/R)}$.

While this multiplicative bound is easy to use, the condition $\|x-y\|_2 \geq \rho$ can be problematic as the protocol cannot check this condition efficiently. Thus, we consider the following additive bound which is a special case of the result in [DMS22].

Theorem F.4 ([DMS22])**.** *There exists a polynomial time algorithm ϕ that for $R > 0$, $0 < \delta < R/2$, $\rho = \mathcal{O}(R\sqrt{\log(R/\delta)})$ and $T \subseteq \mathbf{B}(R)$ with $|T| = n$ where $\mathbf{B}(R)$ is a Euclidean ball of radius R, $\phi : \mathbb{R}^N \to \{0,1\}^\ell$ such that $\ell = \mathcal{O}\left(\frac{\rho^2(\log n+\lambda)}{\delta^2}\right)$, satisfying for any $x, y \in T$,*

$$\left|\frac{\sqrt{2\pi}\rho}{\ell}\mathcal{H}(\phi(x),\phi(y)) - \|x-y\|_2\right| \leq \delta$$

with probability at least $1 - e^{-\mathcal{O}(\delta^2\ell/\rho^2)}$.

Corollary F.5. *There exists a Approx-PSI for Euclidean distance with gap t with communication and computation complexity*
$\mathcal{O}(n^{1+\frac{1}{t-1}}t^2\log t(\log n+\lambda)^2)$.

Proof. Let d_0, t_0 be the threshold and the gap for the underlying Approx-PSI for Hamming distance, respectively. From Theorem F.6, for a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x,y) \leq d$, we have $\mathcal{H}(\phi(x),\phi(y)) \leq (d+\delta)\ell \leq d_0$. For a pair $x,y \in \mathbb{R}^N$ with $d_\theta(x,y) \geq td$, we have $\mathcal{H}(\phi(x),\phi(y)) \geq (td-\delta)\ell \geq t_0 d_0$. Thus, t_0 must satisfy $t_0(d+\delta) \leq td - \delta$.

Since any two vectors in the Euclidean ball of radius R has distance at most $2R$, we may assume that $\frac{R}{d} = \mathcal{O}(t)$. Let $\delta = \mathcal{O}(d)$, $t_0 \leq \frac{td-\delta}{d+\delta} = \mathcal{O}(t)$. We can choose $\rho = \mathcal{O}(R\sqrt{\log t})$. We have $\ell = \mathcal{O}(t^2 \log t(\log n + \lambda))$. This give the communication and computation complexity of the Approx-PSI for Euclidean distance $\mathcal{O}(n^{1+\frac{1}{t-1}t^2\log t(\log n+\lambda)^2)}$ □

For example, when $t = \sqrt{\log\left(\frac{\lambda}{\log n}\right)}\log n$, the communication is $\mathcal{O}(n(\log n+\lambda)^2 R\sqrt{\log\left(\frac{\lambda}{\log n}\right)})$. When $t = \sqrt{\log\left(\frac{\lambda}{\log n}\right)}$, the communication is $\mathcal{O}(n^{1+\epsilon}(\log n+\lambda)^2 R\sqrt{\log\left(\frac{\lambda}{\log n}\right)})$ where $\epsilon = \frac{1}{t_0-1}$ and t_0 is the constant gap of the underlying Approx-PSI for Hamming distance.

When the vectors are in S^{N-1}, we can obtain the result for cosine similarity and cosine distance by transformation

$$\delta_{\cos}(x,y) = \frac{\|x-y\|_2^2}{2}.$$

In this case, the gap is $t = \log\left(\frac{\lambda}{\log n}\right) t_0^2$.

F.3 Angular Distance

The same hyperplane technique above also gives results for angular distance [PV14, YCP15, OR15, DS18]. We consider the embedding described by Dirksen and Stollenwerk [DS18] as its embedding size is smaller, and more concrete parameters are provided. Unlike the first two distance metrics, the angular distance for any pair of vectors are bounded between 0 and 1.

Theorem F.6 ([DS18])**.** *There exists a polynomial time algorithm ϕ that for every $T \subseteq S^{N-1}$ with $|T| = n$, $\phi : S^{N-1} \to \{0,1\}^\ell$ such that $\ell = \mathcal{O}(\log\left(\frac{n}{\eta}\right)/\delta^2)$, satisfying for any $x, y \in T$*

$$\left|\frac{\mathcal{H}(\phi(x),\phi(y))}{\ell} - \delta_\theta(x,y)\right| \leq \delta$$

with probability at least $1-\eta$.

We combine the embedding and the Approx-PSI for Hamming distance to get the following result.

Corollary F.7. *There exists a Approx-PSI for the angular distance where matching vectors have angular distance at most t_1 and non-matching vectors have angular distance at least t_2 with communication and computation complexity $\mathcal{O}(n^{1+\frac{1}{t-1}}t^2(\log n+\lambda)^2)$ where $t = \mathcal{O}(t_2/t_1)$.*

Table 6: Number of rounds for each value of gap t and number of elements in input sets when the security parameter is $\lambda = 40$

gap t	number of elements						
	128	256	512	1024	4096	16384	65536
3	942	1367	1980	2864	5976	12429	25798
4	331	431	558	722	1203	1994	3293
5	189	232	285	349	521	773	1142
6	131	156	186	221	309	429	593
7	101	118	138	160	215	286	378
8	83	96	110	126	164	212	272
9	71	81	92	104	133	167	210
10	63	71	80	89	112	139	171
11	57	63	71	79	97	119	145
12	52	58	64	71	86	104	126
13	48	53	58	64	78	93	111
14	45	49	54	59	71	85	100
15	42	46	51	55	66	78	91
16	40	44	48	52	61	72	84
17	38	41	45	49	57	67	78
18	36	39	43	46	54	63	73

Proof. Let $\delta > 0$ and ℓ as in Theorem F.6. For a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \leq t_1$, we have $\mathcal{H}(\phi(x), \phi(y)) \leq (t_1 + \delta)\ell \leq d$. For a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \geq t_2$, we have $\mathcal{H}(\phi(x), \phi(y)) \geq (t_2 - \delta)\ell \geq td$. Thus, t must satisfy $t(t_1 + \delta) \leq t_2 - \delta$. That is $(t + 1)\delta \leq t_2 - tt_1$. Thus, we need $t_2 - tt_1 > 0$ and $\delta \leq \frac{t_2 - tt_1}{t+1} < \frac{1}{t+1}$ as $t_2 < 1$.

Setting $\eta = 2^\lambda$ and $1/\delta = \mathcal{O}(t)$ gives $\ell = \mathcal{O}(t^2(\log n + \lambda))$ and $t = \mathcal{O}(t_2/t_1)$. This gives the communication and computation complexity of Approx-PSI for Angular distance $\mathcal{O}(n^{1+\frac{1}{t-1}}t^2(\log n + \lambda)^2)$.

□

Here we consider t_2 as a constant while $t_1 < t_2/t$ becomes smaller as t increases.

G Exact Number of Rounds in Approx-PSI for Hamming Distance

Here we calculate the exact number of rounds shown in Table 6 using the calculation from Section 6.

G.1 When inputs do not conform to the structure

Now we discuss what happens when the input sets are not conform to the structure. This means there exists some $a, b \in A \cup B$ such that $d < \delta(a, b) < td$. We note that as we discuss the above reduction, each party must check and modify their input set such that there is only one element representing a cluster of elements of distance at most d from one another. There are two cases:

1. Both parties are (semi-)honest. The problematic case can only occur for $a \in A$ and $b \in B$ as the party must check within their own set. The protocol after the reduction still work correctly and securely even when the pair are projected to the same vector due to the Hamming distance test. As the elements in each party's set are far apart, they will collide with negligible probability, and its match pair will be discovered in one of the projection as analyzed. The problem could, however, arise in the declustering step. If a and b above are hidden from the clusting process, it is possible that $d < \delta(a, b) \leq 2d$. This problem could be solved by further execution of Hamming distance test on such pairs. Though this may lead to worsen efficiency.

2. One party proceeds with bad input set. In this case, it is possible that the party's elements could collide more often than we analyzed. This could lead to some matched pairs undiscovered, in addition to the problem in the first case.

In conclusion, the problem in the first case will only lead to false positive, only occur to the elements hidden in the clustering step, and can be resolved securely by additional checks. The second case only lead to the false negative, and only occur when one party does not check or prepare his own set properly, whether by negligence or with malicious intent.

H Extension to the Malicious Setting

Throughout this work, we have focused on the approximate PSI in the semi-honest setting for simplicity. In this section, we briefly explain how our Approx-PSI protocol for Hamming distance can be extended to remain secure in the malicious setting as well, and so are the ones for other distances. In the Algorithm 1, a malicious party may deviate from the protocol by (1) clustering the elements incorrectly or the elements in their set do not conform to the structure $\mathcal{S}$; (2) providing incorrect element-projection pairs to $\mathcal{F}_{\mathsf{ssPSI}}$; (3) modify their shares output from $\mathcal{F}_{\mathsf{ssPSI}}$, $\mathcal{F}_{\mathsf{ssHamCom}}$ or $\mathcal{F}_{\mathsf{ssVMult}}$.

The deviation (3), as mentioned in Section 4, can be prevented using various authenticated secret sharing techniques. When $\mathcal{F}_{\mathsf{ssPSI}}$ is instantiated using the protocol in [RS21], we may need to modify the circuit PSI to accommodate the secret sharing scheme we may use.

For (1), as discussed in Appendix G.1, it would result in false positive for elements in the cluster or false negative for the elements not conforming to the structure $\mathcal{S}$ in the adversary's set. Neither of which would leak information on the honest party's elements. In this case, however, we need to modify the functionality to allow such mistakes.

Unlike the other two, the deviation (2) requires further machinery to fix. In particular, the parties need to provide a zero-knowledge proof that their computed projection is correct. Since the projection is publicly known linear map, an efficient ZKP can be incorporated into the OKVS technique in the protocol in [RS21].

Since each of the above solution does not require more asymptotic communication, the resulting maliciously secure protocol has the same asymptotic communication complexity as the original protocol.