Secure Binary Embeddings for Privacy Preserving Nearest Neighbors

Petros Boufounos and Shantanu Rane

Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA.
{petrosb,rane}@merl.com

***Abstract*—We present a novel method to securely determine whether two signals are similar to each other, and apply it to approximate nearest neighbor clustering. The proposed method relies on a locality sensitive hashing scheme based on a secure binary embedding, computed using quantized random projections. Hashes extracted from the signals preserve information about the distance between the signals, provided this distance is small enough. If the distance between the signals is larger than a threshold, then no information about the distance is revealed. Theoretical and experimental justification is provided for this property. Further, when the randomized embedding parameters are unknown, then the mutual information between the hashes of any two signals decays to zero exponentially fast as a function of the ℓ_2 distance between the signals. Taking advantage of this property, we suggest that these binary hashes can be used to perform privacy-preserving nearest neighbor search with significantly lower complexity compared to protocols which use the actual signals.**

I. Introduction

A large number of signal processing, machine learning and data mining applications require comparing signals to determine how similar they are, according to some similarity—or distance—metric. In many of these applications, such comparisons are used to determine which of the signals in a cluster are the *nearest neighbors* of a query signal, i.e., the most similar signals to the query from the ones in the cluster. It is, therefore, inevitable that reliable, efficient, and secure search of a signal's nearest neighbors has received significant attention in the literature. In this paper, we propose an efficient, yet secure computation framework to execute this search.

There is a vast literature on nearest neighbor algorithms for various distance measures (e.g., see [1] and references within). In some applications, the cluster of points is distributed among multiple parties and, in such cases, it is necessary to design algorithms that have manageable computational complexity as well as low communication overhead. The difficulty of nearest neighbor search is exacerbated when there are privacy constraints, i.e., when one or more of the involved parties cannot share their data points.

In recent years, with the advent of social networking, internet based storage of user data, and cloud computing, privacy-preserving nearest neighbor search has gained significant attention in the research community. To satisfy the privacy constraints while still allowing distance computation, the data vectors possessed by one or more parties are encrypted using additively homomorphic cryptosystems such as the Benaloh [2], Paillier [3] or Damgard-Jurik [4] schemes. Using cryptographic protocols, a nearest neighbor search scheme is presented in [5]. In this work, nearest neighbor search is performed in which the client does not reveal his query to the server, and the server does not reveal points in its database other than those belonging to the k-nearest neighbor set. The computational complexity of this scheme is quadratic in the number of datapoints, which is a significant overhead since these distance computation was performed in the encrypted domain. This scheme is improved in [6], which uses a pruning technique first proposed in [7] to reduce the number of distance computations and obtain linear computational and communication complexity.

Our contribution in this paper is two-fold: (1) We propose a scheme for nearest-neighbor search based on a secure stable embeddings using quantized random projections. Our approach produces a locality-sensitive hashing method with a special property: The Hamming distance between the hashes is proportional to the ℓ_2 distance between the underlying vectors so long as the latter distance is below a threshold. If the underlying vectors are too far away, the hashes provide no information about the true distance between them, provided the projection parameters are not revealed. (2) We show how to utilize this embedding scheme for privacy-preserving nearest neighbor search by presenting protocols for clustering and authentication applications. A salient feature of these protocols is that distance computation can often be performed on the hashes in cleartext without revealing the underlying data vectors. Thus, the computational overhead, in terms of encrypted-domain distance computation is significantly lower than the state of the art. Further, even when encryption is necessary, the inherent "nearest neighbor within a ball" property can obviate complex subprotocols required in the final step to select a specified number of nearest neighbors, such as in [6].

Our approach is based on recent work on rate-efficient universal scalar quantization [8], and has strong connections with stable binary embeddings for quantization [9] and with Locality-Sensitive Hashing (LSH) approaches to nearest neighbor computation. LSH uses very short hashes of signals to efficiently compute their approximate distance [10], [11]. The key difference in our approach is that we guarantee the information-theoretic security of our embeddings.

WIFS'2011, November 29th-December 2nd, 2011, Foz do Iguaçu, Brazil. 978-1-4577-1019-3/11/$26.00 ©2011 IEEE.

II. Background: Universal Quantization

Universal Scalar Quantization, first introduced in [8], fundamentally revisits scalar quantization and redesigns the quantizer to have non-contiguous quantization regions. In this section we provide a very brief overview.

Given a K-dimensional signal $\mathbf{x} \in \mathbb{R}^K$, we consider the quantization process described by

$$y_m = \langle \mathbf{x}, \mathbf{a}_m \rangle + w_m, \tag{1}$$

$$q_m = Q\left(\frac{y_m}{\Delta_m}\right), \tag{2}$$

compactly represented by

$$\mathbf{q} = Q(\mathbf{\Delta}^{-1}(\mathbf{A}\mathbf{x} + \mathbf{w})), \tag{3}$$

where $m = 1, \ldots, M$ is the measurement index, y_m are the unquantized measurements, $\mathbf{a}_m$ are the measurement vectors, w_m denotes the additive dither, Δ_m are precision parameters, and $Q(\cdot)$ the quantizer, with $\mathbf{y} \in \mathbb{R}^M, \mathbf{A} \in \mathbb{R}^{M \times K}, \mathbf{w} \in \mathbb{R}^M$ and $\mathbf{\Delta} \in \mathbb{R}^{M \times M}$ the corresponding matrix representations. Here, $\mathbf{\Delta}$ is a diagonal matrix with entries Δ_m, and $Q(\cdot)$ is a scalar quantizer, i.e., operates element-wise on vector inputs.

In the remainder of our work, we base our analysis on the results in [8] and, therefore we follow the same assumptions. Specifically, $\mathbf{A}$ is a random matrix with i.i.d. zero-mean, normally distributed entries with variance σ^2, $\Delta_m = \Delta$ is the same and predetermined for all measurements, and $\mathbf{w}$ is uniformly distributed in $[0, \Delta]$. Further, to ensure universality, efficiency and security, we use the quantization function, $Q(\cdot)$, shown in Fig. 1. Under these assumptions, the next lemma—on which we rely for this work—follows.

Lemma 2.1: [8, Lemma 3.1] Consider signals $\mathbf{x}$, and $\mathbf{x}'$ with $d = \|\mathbf{x} - \mathbf{x}'\|_2$ and the quantized measurement function

$$q = Q\left(\frac{\langle \mathbf{x}, \mathbf{a} \rangle + w}{\Delta}\right),\ q' = Q\left(\frac{\langle \mathbf{x}', \mathbf{a} \rangle + w}{\Delta}\right),$$

where $Q(v) = \lceil v \rceil \mod 2$, $\mathbf{a} \in \mathbb{R}^K$ contains i.i.d. elements drawn from a normal distribution with mean 0 and variance σ^2, and w is uniformly distributed in $[0, \Delta]$. The probability that the quantized measurements of the two signals produce equal bits, i.e., that $q = q'$, is given by

$$P(\mathbf{x}, \mathbf{x}' \text{ consistent}|d) = \frac{1}{2} + \sum_{i=0}^{+\infty} \frac{e^{-\left(\frac{\pi(2i+1)\sigma d}{\sqrt{2}\Delta}\right)^2}}{(\pi(i+1/2))^2},$$

where the probability is taken over the distribution of $\mathbf{a}$ and w. Furthermore, the above probability can be bound using

$$P_{c|d} \leq \frac{1}{2} + \frac{1}{2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}, \tag{4}$$

$$P_{c|d} \geq \frac{1}{2} + \frac{4}{\pi^2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}, \tag{5}$$

$$P_{c|d} \geq 1 - \sqrt{\frac{2}{\pi}} \frac{\sigma d}{\Delta}, \tag{6}$$

where $P_{c|d}$ is henceforth shorthand for $P(\mathbf{x}, \mathbf{x}' \text{ consistent}|d)$. Finally, it is also straightforward to demonstrate that thanks to the dither, for a particular signal each quantization bit takes the value is 0 or 1 with the same probability, $\frac{1}{2}$.

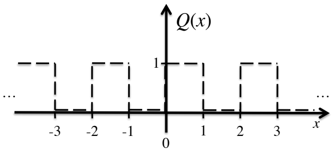

Fig. 1. This non-monotonic quantization function $Q(\cdot)$ allows for universal rate-efficient scalar quantization and provides information-theoretic security.

III. Secure Binary Embeddings

In this section we examine the security and embedding properties of the binary quantization process in (3). The construction of our embedding is very similar, but not identical, to the one constructed in [11]. Since we use the quantization process as an embedding, it has similar properties to Locality Sensitive Hashing (LSH) [10], [11]. Thus we often refer to $\mathbf{q}$, the quantized measurements of $\mathbf{x}$, as the *hash* of $\mathbf{x}$.

Our aim is twofold. First we use an information-theoretic argument to demonstrate that the quantization process provides information about the distance between two signals $\mathbf{x}$ and $\mathbf{x}'$ only if the ℓ_2 distance between them, $\|\mathbf{x} - \mathbf{x}'\|_2$, is sufficiently small. Furthermore, the process does not leak any information about them or their relation if their ℓ_2 distance is sufficiently large. Second, we quantify the information provided by the hashes by demonstrating that they provide a stable embedding of the ℓ_2 distance under the normalized Hamming distance, i.e., we show that the ℓ_2 distance between two signals bounds the normalized Hamming distance between their hashes. A key requirement is that the measurement matrix $\mathbf{A}$ and the dither $\mathbf{w}$ remain secret from the receiver of the hashes. Otherwise, the receiver could, in principle, reconstruct the signals very accurately, according to the guarantees in [8].[1]

A. Information-theoretic Security

To understand the security properties of this embedding we consider the mutual information between the i^{th} bit, q_i and q_i', of the hash of two signals, $\mathbf{x}$ and $\mathbf{x}'$ (measured with the same random $\mathbf{a}_i$ and w_i), conditional on the signal distance d:

$$\begin{aligned}
I(q_i; q_i'|d) &= \sum_{q_i, q_i' \in \{0,1\}} P(q_i, q_i'|d) \log \frac{P(q_i, q_i'|d)}{P(q_i|d)P(q_i'|d)} \\
&= P_{c|d} \log(2P_{c|d}) + (1 - P_{c|d}) \log(2(1 - P_{c|d})) \\
&= \log(2(1 - P_{c|d})) + P_{c|d} \log\left(\frac{P_{c|d}}{1 - P_{c|d}}\right) \\
&\leq \log\left(1 - \frac{4}{\pi^2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}\right) + \\
&\left(\frac{1}{2} + \frac{1}{2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}\right) \log\left(\frac{\frac{1}{2} + \frac{1}{2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}}{\frac{1}{2} - \frac{4}{\pi^2} e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2}}\right) \\
&\leq 10 e^{-\left(\frac{\pi \sigma d}{\sqrt{2}\Delta}\right)^2},
\end{aligned}$$

[1] We should note here that the results in [8] are of theoretical nature, and do not provide any reconstruction algorithms. In fact, reconstruction from such measurements, even if the measurement parameters $\mathbf{A}$ and $\mathbf{w}$ are known, seems to be of combinatorial complexity, and might be computationally prohibitive. Still, we do not rely on this complexity for our development.

where the last step uses $\log x \leq x - 1$ to consolidate the expressions.

For any $\mathbf{x}$, $\mathbf{x}'$ the i^{th} hash bits q_i, q_i' are independent of the j^{th} hash bits q_j, q_j', for $i \neq j$, because the rows of $\mathbf{A}$ and $\mathbf{w}$ are independent random variables. Thus, the mutual information between two length-M hashes, $\mathbf{q}, \mathbf{q}'$ of the two signals is bounded by the following theorem:

Theorem 3.1: Consider two signals, $\mathbf{x}$ and $\mathbf{x}'$, and the quantization method in Lemma 2.1 applied M times to produce the quantized vectors (hashes) $\mathbf{q}$ and $\mathbf{q}'$, respectively. Then,

$$I(\mathbf{q}; \mathbf{q}'|d) \leq 10Me^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2} \tag{7}$$

Theorem 3.1 shows that the mutual information between a pair of hashes decreases exponentially fast with the distance between the signals that generated them. The rate of the exponential decay is controlled by the precision parameter Δ. Thus we cannot recover any information about signals that are far apart, compared to Δ, just by observing their hashes. We remark that this scheme might be susceptible to a chosen-plaintext attack; if a very large number of vectors, carefully selected by an adversary, are all hashed using the same $\mathbf{A}$ and $\mathbf{w}$, then it might be possible to recover $\mathbf{A}$ and $\mathbf{w}$. However, in the applications discussed in Section IV, such an attack is not possible; the protocols ensure that participants who don't already possess $\mathbf{A}$ and $\mathbf{w}$ cannot arbitrarily examine a large number of chosen vector-hash pairs.

B. Stable Embedding

Next, we demonstrate that this approach provides a stable embedding similar in spirit to Johnson-Lindenstrauss embeddings [12]. Such an embedding provides a relationship between distance of signals in the signal space $\mathbb{R}^K$ and the distance of their measurements, i.e., their hashes. Since the hash is in the binary space $\{0, 1\}^M$, the appropriate distance metric is the normalized Hamming distance, denoted $d_H(\mathbf{q}, \mathbf{q}') = \frac{1}{M}\sum_m (q_m \oplus q_m')$.

We first consider the quantization of a pair of vectors $\mathbf{x}, \mathbf{x}'$ with ℓ_2 distance $d = \|\mathbf{x} - \mathbf{x}'\|_2$, as described above. The distance between each pair of individual quantization bits $(q_m \oplus q_m')$ is a random binary value with distribution

$$P(q_m \oplus q_m'|d) = E(q_m \oplus q_m'|d) = 1 - P_{c|d}.$$

Using Hoeffding's inequality [13], it is straightforward to show that the Hamming distance satisfies

$$P\left(|d_H(\mathbf{q}, \mathbf{q}') - (1 - P_{c|d})| \geq t \middle| d\right) \leq 2e^{-2t^2M} \tag{8}$$

Next, consider a cloud of L points to be embedded securely. Using the union bound on at most L^2 possible signal pairs in this cloud, each satisfying (8), the next theorem follows.

Theorem 3.2: Consider a set $\mathcal{S}$ of L signals in $\mathbb{R}^K$ and the quantization method of Lemma 2.1. With probability $1 - 2e^{2\log L - 2t^2M}$ the following holds for all pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{S}$ and their corresponding hashes $\mathbf{q}, \mathbf{q}' \in \{0, 1\}^M$

$$1 - P_{c|d} - t \leq d_H(\mathbf{q}, \mathbf{q}') \leq 1 - P_{c|d} + t, \tag{9}$$

where $P_{c|d}$ is defined in Lemma 2.1, d is the ℓ_2 distance between the signals, and $d_H(\cdot, \cdot)$ is the normalized Hamming distance between their hashes.

This theorem essentially states that with overwhelming probability the normalized Hamming distance between the two hashes will be very close, as controlled by t, to the mapping of the ℓ_2 distance defined by $1 - P_{c|d}$. Furthermore, using the bounds in (4)–(6), we can obtain closed form, albeit looser, embedding bounds for (9):

$$\frac{1}{2} - \frac{1}{2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2} - t \leq d_H(\mathbf{q}, \mathbf{q}') \leq \frac{1}{2} - \frac{4}{\pi^2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2} + t,$$

The mapping $1 - P_{c|d}$, together with its bounds, is shown in Fig 2. The mapping is linear for small d and becomes essentially flat—therefore, not invertible—for large d, with the scaling controlled by the precision parameter Δ. Furthermore, it is very clear in the figure that the upper bounds,

$$1 - P_{c|d} \leq \sqrt{\frac{2}{\pi}}\frac{\sigma d}{\Delta}, \text{ and} \tag{10}$$

$$1 - P_{c|d} \leq \frac{1}{2} - \frac{4}{\pi^2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2}, \tag{11}$$

are very tight for small and large d, respectively, and can be used as approximations of the mapping. Of course, the results of Theorem 3.2, and the bounds on the mapping, can be easily reversed to provide guarantees on the ℓ_2 distance as a function of the Hamming distance.

Figure 3 indicates how the embedding behaves in practice. It shows simulation results on the normalized Hamming distance between pairs of hashes as a function of the ℓ_2 distance between the signals that generated them. The signals are randomly generated in $\mathbb{R}^{1024}$, i.e., $K = 2^{10}$. The top plot uses $M = 2^{12} = 4096$ measurements per hash, i.e., 4 bits per coefficient. The bottom plot uses $M = 2^8 = 256$ measurements per hash, i.e., $1/4$ bit per coefficient. Two different Δ are used in each plot, $\Delta = 2^{-3}, 2^{-1}$. As the Δ increases, the slope of the linear part of the embedding increases, and a larger range of ℓ_2 distances can be identified. This reduces security since information is leaked for signals at longer distances. Furthermore, the width of the linear region increases, which increases the uncertainty in inverting the map in the linear region. On the other hand, as the number of hashing bits M increases, the embedding becomes tighter at the expense of larger bandwidth requirements. This means that the ℓ_2 distance between near neighbors can be more accurately estimated from the hashes. Note that a similar uncertainty on the exact mapping between distance of signals exists even if the signals are quantized, and then compared in the encrypted domain using, for example, a homomorphic cryptosystem.

This behavior is consistent with the information-theoretic security shown earlier for the embedding. For small d, there is information provided in the hashes, which can be used to find the distance between the signals. For large d, information is not leaked so it is not possible to determine the distance between two signals, or any other information, from their hashes.

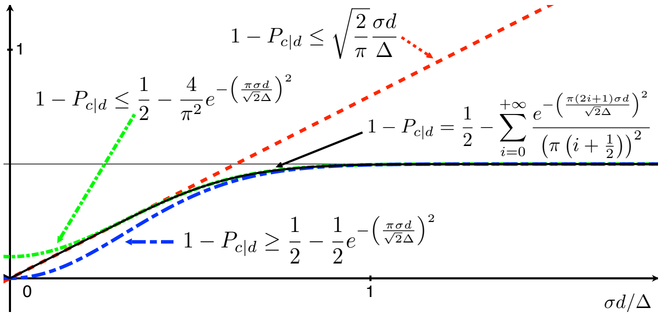

Fig. 2. Embedding map $1 - P_{c|d}$, and its bounds plotted versus the ℓ_2 distance between two signals. The two upper bounds also provide a very good approximation of the embedding, each at a different region of the function.

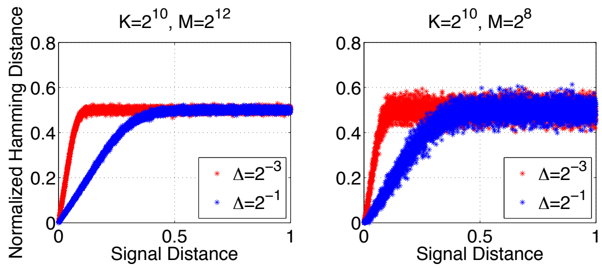

Fig. 3. Simulations demonstrating the embedding behavior. As Δ increases, the slope decreases and the linear region of the map increases, at the expense of revealing information for more distant signals and increasing the uncertainty in the mapping. As the number of bits M increases the mapping becomes tighter, reducing the uncertainty in the mapping, at the expense of larger bandwidth use.

IV. Applications

We now present various application scenarios in which performing a nearest neighbor search based on the hashes is beneficial. We assume that all parties are semi-honest, i.e., they will follow the rules of the protocol but will utilize information available to them at each step of the protocol to discover the data held by other parties. In all of the protocols below, assume that the embedding parameters $\mathbf{A}, \mathbf{w}$ and $\mathbf{\Delta}$ are chosen such that the linear proportionality region in Fig. 2 extends at least up to an ℓ_2 distance of D. Within this proportionality region, denote by D_H the normalized Hamming distance between hashes corresponding to an ℓ_2 distance of D between the underlying signals. Recall from Section III that, outside this region, the embedding is non-invertible and therefore secure. In other words, if the distance between two signals is outside the linear region, then we cannot recover any information about them just by observing their hashes.

A. Privacy Preserving Clustering with a Star Topology

In this application, we take advantage of the property that, when the embedding matrix $\mathbf{A}$ and the dither vector $\mathbf{w}$ are unknown, no information is leaked about the original vector $\mathbf{x}$ by observing the corresponding hash. In the scenario considered here, multiple parties provide data for the purpose of an experiment or a survey performed by a centrally located researcher or auditor. The goal is to allow the researcher to cluster the data and organize the parties into classes without looking at their original data.

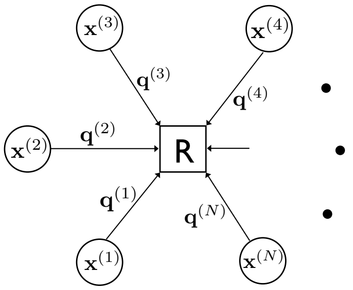

Fig. 4. Researcher can perform approximate nearest neighbor clustering of star-connected parties without discovering their data.

Inputs: There are N parties. Each party U_i possesses data $\mathbf{x}^{(i)}$, $i \in \mathcal{I} = \{1, 2, ..., N\}$.

Output: For each $i \in \mathcal{I}$, the researcher obtains the approximate nearest neighbors of the party U_i within an ℓ_2 distance of D.

Protocol: The protocol transmissions are summarized in Fig. 4 and are explained below.

1) All the parties obtain a common random embedding matrix $\mathbf{A}$, a dither vector $\mathbf{w}$ and the parameter matrix $\mathbf{\Delta}$. One way to accomplish this is for one party to choose $\mathbf{A}, \mathbf{w}$ and $\mathbf{\Delta}$ and transmit them to the other parties using public encryption keys of the intended recipients.
2) Each party U_i computes $\mathbf{q}^{(i)} = Q(\mathbf{\Delta^{-1}}(\mathbf{A}\mathbf{x}^{(i)} + \mathbf{w}))$ and sends $\mathbf{q}^{(i)}$ to the researcher in plaintext form.
3) Corresponding to each party U_i, the researcher constructs sets $\mathcal{G}_i = \{U_j \,|\, d_H(\mathbf{q}^{(i)}, \mathbf{q}^{(j)}) \leq D_H \;\forall\, j \in \mathcal{I}, j \neq i\}$.

From Theorem 3.2, we know that the elements of $\mathcal{G}_i$ are the approximate ℓ_2 nearest neighbors of the party U_i. Note that, owing to the properties of the embedding, the researcher can perform clustering using the binary hashes in cleartext form, without discovering the underlying data $\mathbf{x}^{(i)}$. Thus, apart from the initial overhead incurred in order to communicate the parameters $\mathbf{A}, \mathbf{w}$ and $\mathbf{\Delta}$ to the N parties, encryption is not needed in this protocol. This is in contrast to protocols which need to perform distance calculation based on the original vectors $\mathbf{x}^{(i)}$, which would require the researcher to engage in additional sub-protocols to compute $O(N^2)$ pairwise distances in the encrypted domain using homomorphic encryption.

B. Authentication using Symmetric Keys

Next, we consider authentication using a vector $\mathbf{x}$ derived, for instance, from a biometric or an image. The goal is to authenticate $\mathbf{x}$ with a trusted server without revealing it to an eavesdropper. If the goal is authentication, the user claims an identity and the server should determine whether the submitted vector is within a predefined ℓ_2 distance from that user's enrollment vector stored in its database. If the goal is identification, the server should determine whether or not the submitted vector is within a predefined ℓ_2 distance from at least one enrollment vector stored in its database.

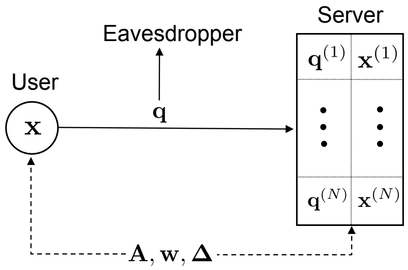

Fig. 5. A user can be identified by an authentication server without revealing its authentication data $\mathbf{x}$ to an eavesdropper. As a variation, to design a protocol for an *untrusted* server, we can stipulate that the server only keeps the $\mathbf{q}^{(i)}$, does not store $\mathbf{x}^{(i)}$ and does not possess the (user-specific) embedding parameters $(\mathbf{A}^{(i)}, \mathbf{w}^{(i)}, \boldsymbol{\Delta}^{(i)})$.

Our proposed scheme accomplishes this by performing the authentication or identification in the subspace of quantized random projections. Here, the triplet $(\mathbf{A}, \mathbf{w}, \boldsymbol{\Delta})$ serves as a symmetric key known only to the user and the authentication server, but not to the eavesdropper. The protocol for the user identification scenario is explained below; the authentication protocol proceeds along very similar lines.

Inputs: The user possesses a vector $\mathbf{x}$ to be used for identification. The server possesses a database of N enrollment vectors $\mathbf{x}^{(i)}, i \in \mathcal{I} = \{1, 2, ..., N\}$. The user and the server (but not the eavesdropper) possess $(\mathbf{A}, \mathbf{w}, \boldsymbol{\Delta})$.

Output: The server determines $\mathcal{G}$ which is the set of approximate nearest neighbors of the probe vector $\mathbf{x}$ within an ℓ_2 distance of D. If $\mathcal{G} = \emptyset$, user identification has failed, otherwise the user has been identified as being close to at least one legitimate enrolled user of the database. The eavesdropper obtains no information about $\mathbf{x}$.

Protocol: The protocol transmissions are summarized in Fig. 5 and are explained in detail below:

1) The user computes $\mathbf{q} = Q(\boldsymbol{\Delta}^{-1}(\mathbf{A}\mathbf{x} + \mathbf{w}))$ and sends $\mathbf{q}$ to the server in plaintext form.
2) The server computes $\mathbf{q}^{(i)} = Q(\boldsymbol{\Delta}^{-1}(\mathbf{A}\mathbf{x}^{(i)} + \mathbf{w}))$ for all $i \in \mathcal{I}$.
3) The server constructs the set $\mathcal{G} = \{i \,|\, d_H(\mathbf{q}, \mathbf{q}^{(i)}) \leq D_H \;\forall\, i \in \mathcal{I}\}$.

Again, from Theorem 3.2, we see that the set $\mathcal{G}$ contains the approximate ℓ_2 nearest neighbors of $\mathbf{x}$. If $\mathcal{G} = \emptyset$, then identification has failed, otherwise the user has been identified as having one of the indices in $\mathcal{G}$. As the eavesdropper does not know $(\mathbf{A}, \mathbf{w}, \boldsymbol{\Delta})$, the quantized projections do not reveal information about the underlying vector. This protocol does not require the user to encrypt the hash before sending it to the authentication server. In terms of the communication overhead, this is an advantage over conventional nearest neighbor search algorithms which require that the client should send the original vector to the server in encrypted form in order to hide it from the eavesdropper.

An interesting variation of the above scheme is as follows: If the authentication server is *untrusted*, users would not wish to enroll using their identifying vectors $\mathbf{x}^{(i)}$. In that case, change the above protocol so that only a user (but not the server) possesses $(\mathbf{A}^{(i)}, \mathbf{w}^{(i)}, \boldsymbol{\Delta}^{(\mathbf{i})})$. The users now enroll into the server's database using the hashes $\mathbf{q}^{(i)}$ instead of the corresponding vectors $\mathbf{x}^{(i)}$. These hashes are the only data stored on the server. In this situation, since the server does not know $(\mathbf{A}^{(i)}, \mathbf{w}^{(i)}, \boldsymbol{\Delta}^{(\mathbf{i})})$, it cannot reconstruct $\mathbf{x}^{(i)}$ from $\mathbf{q}^{(i)}$. Further, if the server's database is compromised, then the $\mathbf{q}^{(i)}$ can be revoked and new hashes can be enrolled using different embedding parameters $(\mathbf{A}^{(i)\prime}, \mathbf{w}^{(i)\prime}, \boldsymbol{\Delta}^{(\mathbf{i})\prime})$.

C. Privacy Preserving Clustering with Two Parties

Next, we consider a two-party protocol in which a client initiates a query on a server's database. The privacy constraint is that the server should not discover the client's query vector while the client should only discover the vectors in the server's database that are within a predefined ℓ_2 distance from its query.

Unlike the earlier protocol with the star topology in Section IV-A, it is now necessary to use a homomorphic cryptosystem scheme such as Paillier cryptosystem [3] to perform simple operations in the encrypted domain. The additively homomorphic property of the Paillier cryptosystem ensures that $\xi_p(a)\xi_q(b) = \xi_{pq}(a + b)$ where a, b are integers in the message space, and $\xi(\cdot)$ is the encryption function. The integers p, q are randomly chosen encryption parameters which make the Paillier cryptosystem semantically secure, i.e., by choosing the parameters p, q at random, one can ensure that repeated encryptions of a given plaintext results in different ciphertexts, thereby protecting against chosen plaintext attacks (CPAs). For simplicity, we will drop the suffixes p, q from our notation. As a corollary to the additively homomorphic property, we have $\xi(a)^b = \xi(ab)$.

Inputs: The client has a query vector $\mathbf{x}$. The server has a database of N vectors $\mathbf{x}^{(i)}, i \in \mathcal{I} = \{1, 2, ..., N\}$. The server generates the triplet $(\mathbf{A}, \mathbf{w}, \boldsymbol{\Delta})$ and makes $\boldsymbol{\Delta}$ public.

Output: The client obtains $\mathcal{G}$, the set of approximate nearest neighbors of the query vector $\mathbf{x}$ within an ℓ_2 distance of D. If no such vectors exist, then the client obtains $\mathcal{G} = \emptyset$.

Protocol: The protocol transmissions are summarized in Fig. 6 and the steps are detailed below:

1) The client generates a public encryption key, pk, and secret decryption key, sk, for Paillier encryption. Then, it performs elementwise encryption of $\mathbf{x}$. Denote the elementwise encryption by $\xi(\mathbf{x}) = (\xi(x_1), \xi(x_2), ..., \xi(x_K))$. The client transmits $\xi(\mathbf{x})$ to the server.
2) The server uses the additively homomorphic property to compute $\xi(\mathbf{y}) = \xi(\mathbf{A}\mathbf{x} + \mathbf{w})$ and returns $\xi(\mathbf{y})$ to the client.
3) The client decrypts $\mathbf{y}$ and computes $\mathbf{q} = \boldsymbol{\Delta}^{-1}\mathbf{y}$. It then sends $\xi(\mathbf{q})$ to the server.
4) The server computes the hashes of its entire database, i.e., it obtains $\mathbf{q}^{(i)} = \boldsymbol{\Delta}^{-1}(\mathbf{A}\mathbf{x}^{(i)} + \mathbf{w})$ for all $i \in \mathcal{I}$.

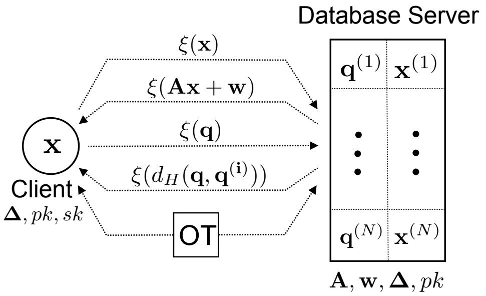

Fig. 6. A client can obtain the approximate nearest neighbors of the query vector $\mathbf{x}$ using the proposed locality-sensitive hashing scheme in conjunction with a additively homomorphic cryptosystem.

5) Now, the server uses homomorphic properties to compute the encryption of the Hamming distances between the quantized query vector and the quantized database vectors, i.e., it computes $d_H(\mathbf{q}, \mathbf{q}^{(i)})$ for all $i \in \mathcal{I}$ as follows:

$$\xi(M d_H(\mathbf{q}, \mathbf{q}_i)) = \xi(\sum_{m=1}^{M} q_m \oplus q_m^{(i)}) = \prod_{m=1}^{M} \xi(q_m \oplus q_m^{(i)})$$
$$= \prod_{m=1}^{M} \xi(q_m)\xi(q_m^{(i)})\xi(q_m)^{-2q_m^{(i)}}$$

The server sends the encrypted distances to the client.

6) The client decrypts $d_H(\mathbf{q}, \mathbf{q}^{(i)})$ for all $i \in \mathcal{I}$ and obtains the set $\mathcal{D} = \{i \,|\, d_H(\mathbf{q}, \mathbf{q}^{(i)}) \leq D_H \;\forall\; i \in \mathcal{I}\}$.

7) If $\mathcal{D} = \emptyset$, the protocol concludes. If not, the client performs a $|\mathcal{D}|$-out-of-N oblivious transfer (OT) protocol with the server to retrieve $\mathcal{G} = \{\mathbf{x}^{(i)} \,|\, i \in \mathcal{I} \bigcap \mathcal{D}\}$. OT (See [14], [15]) guarantees that the client does not discover any $\mathbf{x}^{(i)}$ such that $i \notin \mathcal{D}$ while ensuring that the query set $\mathcal{D}$ is not revealed to the server.

From Theorem 3.2, the set $\mathcal{G}$ contains the approximate ℓ_2 nearest neighbors of the query vector $\mathbf{x}$. Consider the advantages of computing distances in the hash subspace versus encrypted-domain computation of distance between the underlying vectors. For a database of size N, computing the distances between the vectors would reveal all N distances $\| \mathbf{x} - \mathbf{x}^{(i)} \|_2, i = 1, 2, ..., N$. A separate sub-protocol is necessary to ensure that only the distances corresponding to the nearest neighbors, i.e., the local distribution of the distances, is revealed to the client. In contrast, our proposed protocol naturally reveals the distances only when $\| \mathbf{x} - \mathbf{x}^{(i)} \|_2 \leq D$. If $\| \mathbf{x} - \mathbf{x}^{(i)} \|_2 > D$, then the Hamming distances between the hashes are no longer proportional to the true distances. This prevents the client from knowing the global distribution of the vectors in the server's database, while only revealing the local distribution of vectors close to the query vector.

V. Conclusions

We presented a secure binary embedding scheme using quantized random projections, which preserves the distances between vectors in a special way: So long as one vector is within a pre-specified distance d from another vector, the normalized Hamming distance between their two quantized projections is approximately proportional to the ℓ_2 distance between the two vectors. However, as the distance between the two vectors increases beyond d, then the Hamming distance between their projections becomes independent of the distance between the vectors. The embedding further exhibits some useful privacy properties: The mutual information between any two hashes decays towards zero exponentially fast as a function of the ℓ_2 distance between the two underlying signals.

We use this embedding approach to perform efficient privacy-preserving nearest neighbor search. Most privacy-preserving nearest neighbor searching algorithms are carried out using the original vectors, which must be encrypted in order to satisfy privacy constraints. On the other hand, because of the aforementioned properties, the proposed hashes can be used instead of the original vectors to implement privacy-preserving nearest neighbor search at significantly lower complexity or higher speed. To motivate this, we presented protocols in low-complexity clustering, and server-based authentication, though many other applications are possible.

References

[1] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley New York, 2001, vol. 2.

[2] J. Benaloh, "Dense Probabilistic Encryption," in *Proc. Workshop on Selected Areas of Cryptography*, Kingston, ON, Canada, May 1994, pp. 120–128.

[3] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology, EUROCRYPT 99*, vol. 1592. Springer-Verlag, Lecture Notes in Computer Science, 1999, pp. 233–238.

[4] I. Damgård and M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System," in 4th *Intl. Workshop on Practice and Theory in Public Key Cryptosystems*, Cheju Island, Korea, Feb. 2001, pp. 119–136.

[5] M. Shaneck, Y. Kim, and V. Kumar, "Privacy preserving nearest neighbor search," in *Proc. of the Sixth IEEE Intl. Conf. Data Mining - Workshops*, Washington, DC, USA, 2006, pp. 541–545.

[6] Y. Qi and M. J. Atallah, "Efficient privacy-preserving k-nearest neighbor search," *Intl. Conference on Distributed Computing Systems*, vol. 0, pp. 311–319, 2008.

[7] T. Seidl and H. P. Kriegel, "Optimal multi-step k-nearest neighbor search," in *Proc. 1998 ACM SIGMOD Intl. Conf. Management of Data*, 1998, pp. 154–165.

[8] P. T. Boufounos, "Universal rate-efficient scalar quantization," 2010, preprint, http://arxiv.org/abs/1009.3145, Submitted.

[9] L. Jacques, J. Laska, P. Boufounos, and R. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," 2011, preprint, http://arxiv.org/abs/1104.3160.

[10] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[11] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," *The Neural Information Processing Systems*, vol. 22, 2009.

[12] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math*, vol. 26, pp. 189–206, 1984.

[13] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, March 1963.

[14] M. O. Rabin, "How to exchange secrets with oblivious transfer," Cryptology ePrint Archive, Report 2005/187, 2005, http://eprint.iacr.org/.

[15] C. Chu and T. Tzeng, "Efficient k-out-of-n oblivious trans- fer schemes with adaptive and non-adaptive queries," in *Intl. Workshop on Practice and Theory in Public Key Cryptography*, vol. 3386. Les Diablerets, Switzerland: Springer-Verlag, Lecture Notes in Computer Science, Jan. 2005, pp. 172–183.