MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Privacy-Preserving Approximation Of L1 Distance for Multimedia Applications

Shantanu Rane, Wei Sun, Anthony Vetro

TR2010-077 October 2010

Abstract

Alice and Bob posses sequences x and y respectively and would like to compute the — distance, namely — under privacy and communication constraints. The privacy constraint requires that Alice and Bob do not reveal their data to each other. The communication constraint requires that they accomplish the secure distance calculation with a small number of protocol transmissions and key exchanges. This paper describes and analyzes a privacy-preserving approximation protocol for the — distance that keeps the communication overhead manageable by performing a Johnson-Lindenstrauss embedding into the — space. Then, it performs secure two-party computation of — distance using Paillier homomorphic encryption. The protocol is implemented for private querying of face images, while maintaining a low communication overhead between the querying party and a remote database of face feature vectors.

IEEE International Conference on Multimedia and Expo

PRIVACY-PRESERVING APPROXIMATION OF L1 DISTANCE FOR MULTIMEDIA APPLICATIONS

Shantanu Rane, Wei Sun and Anthony Vetro

Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
{rane,weisun,avetro}@merl.com

ABSTRACT

Alice and Bob possess sequences $\mathbf{x}$ and $\mathbf{y}$ respectively and would like to compute the ℓ_1 distance, namely $\| \mathbf{x} - \mathbf{y} \|_1$ under privacy and communication constraints. The privacy constraint requires that Alice and Bob do not reveal their data to each other. The communication constraint requires that they accomplish the secure distance calculation with a small number of protocol transmissions and key exchanges. This paper describes and analyzes a privacy-preserving approximation protocol for the ℓ_1 distance that keeps the communication overhead manageable by performing a Johnson-Lindenstrauss embedding into the ℓ_2 space. Then, it performs secure two-party computation of ℓ_2 distances using Paillier homomorphic encryption. The protocol is implemented for private querying of face images, while maintaining a low communication overhead between the querying party and a remote database of face feature vectors.

***Keywords*—** Homomorphic Encryption, Secure Multiparty Computation, Johnson-Lindenstrauss embedding

1. INTRODUCTION

Consider a private querying system in which a user wants to check if his query image matches one or more images in a database of images. The querying device (Alice), can extract features from the user's image and match these against a feature vector database (Bob). A common distance metric for matching image features is the Manhattan distance, or ℓ_1 distance. Suppose that the matching criterion is that the ℓ_1 distance between Alice's feature vector and one of Bob's feature vectors is below a certain threshold. However, private querying imposes some constraints on how this matching is performed. Firstly, for the user's privacy, Bob must not know anything about Alice's feature vector. Secondly, for security of images in the database, Alice must not find out anything about Bob's feature vectors. Thirdly, for practical usage, the communication protocol employed by Alice and Bob should not incur a large transmission overhead. We propose a secure approximation protocol to address privacy and communication constraints in problems of this kind.

There has been significant work on the approximation of distances in different metric spaces, resulting in elegant solutions based on low-dimensionality embeddings. Du and Atallah [1] used Monte-Carlo techniques to achieve approximation of ℓ_1 and ℓ_2 distances between two signals. In a streaming scenario, Feigenbaum *et al.* used range-summable random variables to approximate the ℓ_1 distance between two massive data streams [2]. This computation is not secure by design, but in our opinion, it can be made secure using oblivious transfer and secure dot product protocols of Yao [3]. However, this would incur a significant communication overhead between Alice and Bob. Indyk [4] used stable distributions to map the points $\mathbf{x}$ and $\mathbf{y}$ in the original space to points $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ in a lower-dimensional space while approximately preserving the ℓ_1 distance. If Alice and Bob were each to apply this low-dimensional embedding, they still need a protocol to compute the $\| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_1$ with privacy. Unfortunately, a privacy preserving two-party protocol to compute $\| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_1$ with low communication overhead is not available.

To circumvent the above difficulty, we propose a three-step approach in which the original points $\mathbf{x}$ and $\mathbf{y}$ are first mapped into new binary vectors $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ which reside on a higher dimensional Hamming cube. Next, using Johnson-Lindenstrauss (JL) embedding, these new points are further mapped into $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$. The mappings ensure that $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2 \approx \| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_2^2 = \| \mathbf{x} - \mathbf{y} \|_1$. The reduction in dimension via JL embedding partially compensates for the increase in dimensionality in the binarization stage. The problem is then reduced to privacy-preserving computation of squared ℓ_2 distance in a space of slightly larger dimension than the original space. For computing the ℓ_2 distance, we propose an efficient two-party protocol using Paillier homomorphic encryption. The protocol is secure for computationally bounded Alice and Bob, and incurs much lower communication overhead compared to oblivious transfer on individual components of $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$.

The remainder of this paper is organized as follows: Section 2 describes the first two steps in which the problem of computing ℓ_1 distance is reduced to one of computing ℓ_2 distance. Section 3 reviews Paillier homomorphic encryption and describes the protocol used for secure computation of ℓ_2 distance. In Section 4, the operations of the previous two sections are combined to construct the proposed privacy-preserving ℓ_1 distance approximation protocol. In Section 5, this protocol is implemented in a private image querying system that uses ℓ_1 distance as a matching criterion for face features. It is shown that, the approximate ℓ_1 distance computed by the protocol is accurate enough to retain the security-robustness properties of

978-1-4244-7493-6/10/$26.00

the inherent matching algorithm. Further, private querying is feasible at manageable communication overhead of a few hundred kilobytes per querying instance, including the bit rate expansion resulting from homomorphic encryption.

2. APPROXIMATION OF ℓ_1 DISTANCE

Let Alice and Bob possess two integer sequences of length n, viz., $\mathbf{x} = (x_1, x_2, ..., x_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_n)$, where $x_i, y_i \in \{0, 1, ..., M-1\}$ for all $i \in \{1, 2, ..., n\}$. Define a function $f : \{0, 1, ..., M-1\} \longrightarrow \{0,1\}^{M-1}$, such that $f(u)$ is a binary vector containing 1's as its first u entries and 0's as the following $M-1-u$ entries. For vector arguments, define $\widetilde{\mathbf{x}} = f(\mathbf{x}) = (f(x_1), f(x_2), ..., f(x_n))$ and $\widetilde{\mathbf{y}} = f(\mathbf{y}) = (f(y_1), f(y_2), ..., f(y_n))$. Clearly,

$$\| \mathbf{x} - \mathbf{y} \|_1 = \| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_1 = \| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_2^2 \quad (1)$$

The vectors $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ both have length $n*(M-1)$. There are M^n such vectors[1] and they reside in $\{0,1\}^{(M-1)n}$. Using (1), Alice and Bob can employ the secure ℓ_2 distance protocol explained in Section 3 and obtain $\| \mathbf{x} - \mathbf{y} \|_1$ exactly, but this will incur a very high communication overhead. Therefore, we propose to embed $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ into a lower dimensional space such that the ℓ_2 distance between them is approximately preserved. For this, we use the Johnson Lindenstrauss (JL) Lemma [6].

Lemma 1 *[6] Given $\epsilon > 0$ and an integer s, let k be a positive integer such that $k \geq k_0 = O(\epsilon^{-2} \log s)$. For every set P of s points in $\mathbb{R}^d$ there exists $g : \mathbb{R}^d \longrightarrow \mathbb{R}^k$ such that for all $\boldsymbol{u}, \boldsymbol{v} \in P$*

$$(1-\epsilon)||\boldsymbol{u} - \boldsymbol{v}||_2^2 \leq ||g(\boldsymbol{u}) - g(\boldsymbol{v})||_2^2 \leq (1+\epsilon)||\boldsymbol{u} - \boldsymbol{v}||_2^2 \quad (2)$$

Roughly speaking, using the embedding function g the ℓ_2 distance between any two points $\mathbf{u}, \mathbf{v}$ in a high dimensional space is approximately equal to the ℓ_2 distance between $g(\mathbf{u}), g(\mathbf{v})$ in a lower dimensional space with high probability. Recently, Achlioptas [7] gave a constructive proof of JL Lemma, showing that for given $\epsilon, \beta > 0$, the inequality (2) holds with probability at least $1 - s^{-\beta}$ if $k \geq k_0 = \frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3} \log s$. The embedding g is a linear transformation given by $g(\mathbf{u}) = \frac{1}{\sqrt{k}} \mathbf{Ru}$, where each entry of the $k \times d$ matrix $\mathbf{R}$ can be generated i.i.d. from a ± 1-valued Bernoulli-0.5 distribution or from a normal distribution. For ease of computation, the ± 1-valued Bernoulli-0.5 distribution will be employed in this paper.

To determine the number of reduced dimensions after JL embedding, set $s = M^n$ in the JL Lemma. Then, for parameters $\epsilon, \beta > 0$, the minimum number of random projections is given by:

$$k = \frac{(4+2\beta)n}{\epsilon^2/2 - \epsilon^3/3} \log M \quad (3)$$

[1] A similar mapping recently appeared in [5] for private image retrieval.

The parameters ϵ and β are useful in analyzing the fidelity of the approximation of ℓ_2 distance achieved by JL. In a practical implementation, we can simplify (3) to $k = \alpha n \log_w M$ where the constant α captures the parameters β and ϵ in addition to the change in the base of the logarithm. Increasing α increases the number of random projections required, thereby improving the accuracy of the embedding.

Recall from above, that the entries of the matrix $\mathbf{R}$ are i.i.d. Bernoulli-0.5 random variables and take values +1 or -1. The seed used to generate the entries of $\mathbf{R}$ is assumed to be shared between Alice and Bob. Now define the linear JL embedding function $g : \mathbb{R}^{n(M-1)} \longrightarrow \mathbb{R}^k$ as

$$\widehat{\mathbf{x}} = g(\widetilde{\mathbf{x}}) = \frac{1}{\sqrt{k}} \mathbf{R}\widetilde{\mathbf{x}} \ , \ \widehat{\mathbf{y}} = g(\widetilde{\mathbf{y}}) = \frac{1}{\sqrt{k}} \mathbf{R}\widetilde{\mathbf{y}}$$

The vectors $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$ have length k. Therefore, by the J-L Lemma and (1),

$$\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2 \approx \| \widetilde{\mathbf{x}} - \widetilde{\mathbf{y}} \|_2^2 = \| \mathbf{x} - \mathbf{y} \|_1 \quad (4)$$

From the point of view of implementation, note that $\widehat{\mathbf{x}}, \widehat{\mathbf{y}}$ are not integer sequences. It is much more convenient to use integer vectors for the next stage of the algorithm, namely the protocol for secure computation of ℓ_2 distance. To accomplish this, we note that $\sqrt{k}\,\widehat{\mathbf{x}}, \sqrt{k}\,\widehat{\mathbf{y}}$ are integer sequences, so without loss of generality, the secure ℓ_2 distance protocol can operate on $\sqrt{k}\,\widehat{\mathbf{x}}, \sqrt{k}\,\widehat{\mathbf{y}}$ and the division by $\sqrt{k}$ can be handled later on. Another approach is to quantize each component of $\widehat{\mathbf{x}}, \widehat{\mathbf{y}}$ to the closest integer. In the sequel, assume that Alice and Bob have respectively and separately computed $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$ which are "good" embeddings in the sense of (4).

3. SECURE COMPUTATION OF ℓ_2 DISTANCE

We now describe secure computation of exact squared ℓ_2 distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ of length t using Paillier homomorphic encryption. The Paillier cryptosystem [8] is reviewed briefly below.

- **Configuration:** Choose two large prime numbers p, q, and let $N = pq$. Denote by $\mathbb{Z}_{N^2}^* \subset \mathbb{Z}_{N^2} = \{0, 1, ..., N^2 - 1\}$ the set of non-negative integers that have multiplicative inverses modulo N^2. Select $g \in \mathbb{Z}_{N^2}^*$ such that $\gcd(L(g^\lambda \mod N^2), N) = 1$, where $\lambda = \text{lcm}(p-1, q-1)$, and $L(x) = \frac{x-1}{N}$. Let (N, g) be the public key, and (p, q) be the private key.
- **Encryption:** Let $m \in \mathbb{Z}_N$ be a plaintext. Then, the ciphertext is given by

$$\xi_r(m) = g^m \cdot r^N \mod N^2 \quad (5)$$

  where $r \in \mathbb{Z}_N^*$ is a number chosen at random.
- **Decryption:** Let $c \in \mathbb{Z}_{N^2}$ be a ciphertext. Then, the corresponding plaintext is given by

$$\psi(\xi_r(m)) = \frac{L(c^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} = m \mod N \quad (6)$$

Note that decryption works irrespective of the value of r used during encryption. Since r can be chosen at random for every encryption, the Paillier cryptosystem is probabilistic, and therefore semantically secure. It can now be verified that the following homomorphic properties hold for the mapping (5) from the plaintext set $(\mathbb{Z}_N, +)$ to the ciphertext set $(\mathbb{Z}^*_{N^2}, \cdot)$,

$$\psi(\xi_r(m_1)\xi_r(m_2) \mod N^2) = m_1 + m_2 \mod N \quad (7)$$

$$\psi([\xi_r(m_1)]^{m_2} \mod N^2) = m_1 m_2 \mod N \quad (8)$$

Now, suppose that Alice and Bob own two integer sequences $\mathbf{a} = \{a_1, a_2, ..., a_t\}$ and $\mathbf{b} = \{b_1, b_2, ..., b_t\}$ respectively and let $t \ll N$.

$$\begin{aligned} \| \mathbf{a} - \mathbf{b} \|_2^2 &= \sum_{i=1}^{t} (a_i - b_i)^2 = \sum_{i=1}^{t} (a_i^2 + b_i^2 - 2a_i b_i) \\ &= A + B + C \end{aligned} \quad (9)$$

$$\text{where } A = \sum_{i=1}^{t} a_i^2,\ B = \sum_{i=1}^{t} b_i^2,\ C = -\sum_{i=1}^{t} 2a_i b_i$$

Observe that Alice knows A, Bob knows B, but C contains the cross terms and is unknown to both of them. For secure computation, Alice generates a public/private key pair and shares only the public key with Bob. We assume that Alice and Bob are honest but curious, i.e., each of them will follow the steps of the protocol but will attempt to extract as much information as possible from the data made available to them by the protocol. Now, the protocol for secure computation of the squared ℓ_2 distance is as follows:

1. For each $i \in 1, 2, ..., t$, Alice encrypts a_i into $\xi_{r_i}(a_i)$ according to (5). Here, r_i is chosen randomly from $\mathbb{Z}^*_N$. She transmits the encrypted results to Bob.

2. For each $i \in 1, 2, ..., t$, Bob computes

$$\tilde{b}_i = -2b_i \mod N$$

$$\xi_{r_i}(-2a_i b_i) \equiv [\xi_{r_i}(a_i)]^{\tilde{b}_i} \mod N^2$$

3. Bob computes

$$\xi_{r_C}(C) \equiv \xi_{r_C}(-\sum_{i=1}^{t} 2a_i b_i) \equiv \prod_{i=1}^{t} \xi_{r_i}(-2a_i b_i) \bmod N^2$$

where $r_C = \prod_{i=1}^{t} r_i \mod N \in \mathbb{Z}^*_N$. Note that Bob operates solely in the encrypted domain in this step, so the values of C and r_C are unknown to him.

4. Bob chooses $r_B \in \mathbb{Z}^*_N$ at random and computes

$$\xi_{r_D}(B + C) \equiv \xi_{r_B}(B)\, \xi_{r_C}(C) \mod N^2$$

where $r_D = r_B r_C \mod N \in \mathbb{Z}^*_N$. Bob transmits this result to Alice. The value of r_D is implicit in the encryption result but is unknown to Bob.

5. Alice chooses $r_A \in \mathbb{Z}^*_N$ at random and computes

$$\begin{aligned} \xi_r(\| \mathbf{a} - \mathbf{b} \|_2^2) &= \xi_r(A + B + C) \\ &\equiv \xi_{r_A}(A)\, \xi_{r_D}(B + C) \mod N^2 \end{aligned}$$

where $r = r_A r_D \mod N \in \mathbb{Z}^*_N$. Again, the value of r is implicit in the encryption result but unknown to Alice because she does not know r_D.

6. Using the private key, Alice decrypts $A + B + C = \| \mathbf{a} - \mathbf{b} \|_2^2$ according to (6). If required by the application, this result is transmitted to Bob. The decrypted ℓ_2 distance requires far fewer bits to transmit compared to the encrypted transmissions in the previous steps, so the communication overhead of this last step is neglected.

Note that, by design, the protocol does not reveal **a** to Bob or **b** to Alice. In order to know a_i, Bob must decrypt Alice's transmissions in the absence of the decryption key. Since he is computationally bounded, Alice's inputs are computationally secure. Since Paillier encryption is semantically secure, repeated encryptions of a bit value (0 or 1) will result in a different ciphertext every time, dictated by the random choice of r in (5). If Alice wants to find out Bob's inputs, she must decrypt $\xi_{r_D}(B + C) = \xi_{r_D}(\sum_{i=1}^{t}(b_i^2 - 2a_i b_i))$ which gives her 1 equation and t unknowns. Thus, for $t \geq 2$, Bob's data is secure. In terms of the communication overhead, Alice transmits a maximum of $t \log N^2$ bits in Step 1. Thus Alice's maximum communication overhead is $t \log N^2$ bits, while Bob transmits a maximum of $\log N^2$ bits. Since N is a fixed constant based on the desired security of the encryption algorithm, the communication complexity, in terms of the vector length t, is $O(t)$ for Alice and $O(1)$ for Bob. For more details on the computational overhead incurred by Alice and Bob, the reader is referred to [9].

4. SECURE COMPUTATION OF APPROXIMATE ℓ_1 DISTANCE

Finally, we cascade the operations in Section 2 and Section 3 and generate a secure two-party approximation of the ℓ_1 distance between **x** and **y**. In the setup phase, Alice and Bob share a randomly picked seed that they use to generate the matrix **R** for JL embedding. Also, Alice generates a public/private key pair for Paillier encryption and shares the public key with Bob[2]. Now, the steps of the combined protocol are as follows:

1. Starting with **x**, Alice obtains $\widehat{\mathbf{x}}$ using binarization followed by JL embedding as explained in Section 2. Similarly, starting with **y**, Bob obtains $\widehat{\mathbf{y}}$. These $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$ satisfy (4).

2. Alice sets $\mathbf{a} = \widehat{\mathbf{x}}$. Bob sets $\mathbf{b} = \widehat{\mathbf{y}}$. The two parties then employ the protocol in Section 3. After one round of

[2]The communication overhead for sharing the random seed to generate **R** is assumed negligible. The overhead for transmission of the public encryption key with Bob can be folded into the first step of the secure ℓ_2 distance protocol.

communication, Alice obtains the value of $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2 \approx \| \mathbf{x} - \mathbf{y} \|_1$. If necessary, this value can be transmitted to Bob.

When Alice and Bob are computationally bounded, the protocol from Section 3 ensures that the privacy constraints in the two-party computation are satisfied. Let e be the absolute value of the error between the ℓ_1 distance and its ℓ_2 approximation. Then, from the JL Lemma, we have with probability at least $\rho(s,\beta) = 1 - s^{-\beta}$, the absolute error $e \leq \epsilon \| \mathbf{x} - \mathbf{y} \|_1$. Using (3), it is clear that reduction in ϵ and increase in $\rho(s,\beta)$ are achieved at the expense of an increased number of random projections, i.e., the number of rows of $\mathbf{R}$. Calculating the number of projections by choosing β and ϵ values using (3) gives a conservative (hence large) estimate of the required number of projections. We find that a much smaller number of projections is sufficient in practice. For instance, in the implementation of Section 5, we use $M = 256, n = 900, w = 2, \alpha = 1$ resulting in an embedding into a space with $k = \alpha n \log_w M = 7200$ dimensions. It is especially important to keep the number of projections small because, in the secure computation protocol, the projections have to be encrypted and the resulting ciphertexts, being large, incur a significant transmission penalty.

5. PRIVATE QUERYING OF FACE DATABASES

We now consider an application of the proposed protocol for private querying of face images that was introduced at the beginning of the paper. As shown in Fig. 1, a photograph of the query face is presented to the querying device (Alice), which performs face detection and extracts an integer feature vector $\mathbf{x}$ from it. This integer feature vector must now be compared with a remote database (Bob) containing face feature vectors $\mathbf{y}_i$ of a large group of people, such as employees in an organization, suspects in crime scenes, and so on. The query will be deemed successful if $\| \mathbf{x} - \mathbf{y}_i \|_1 \leq D_{th}$ for some $\mathbf{y}_i$, where D_{th} is some threshold agreed upon by Alice and Bob. It is desired that querying be accomplished such that (a) The query face features are not revealed to the database server, and (b) The querying device gets no information about the $\mathbf{y}_i$ in the database except what is conveyed by $\| \mathbf{x} - \mathbf{y}_i \|_1$ for all i. This requires that the database vectors are not indexed by the name of the legitimate user. This is accomplished by using a different random ordering of the $\mathbf{y}_i$ for every querying instance, and approximating $\| \mathbf{x} - \mathbf{y}_i \|_1$ according to that ordering[3].

Let $n > 2$ be the length of the integer feature vectors, and U be the number of feature vectors in the database. We now present a scheme which achieves these objectives with $O(n)$ and $O(U)$ communication overhead from Alice and Bob respectively. The querying device (Alice) possesses a public/private key pair, while the database (Bob) possesses only the public key but not the private key. Using the JL embedding in Section 2, the vectors $\mathbf{x}$ and $\mathbf{y}_i$, $i = 1, 2, ..., U$ are transformed to $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}_i$ such that $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2 \approx \| \mathbf{x} - \mathbf{y} \|_1$. By following the secure ℓ_2 distance protocol in Section 3, Alice obtains the values of $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}}_i \|_2^2$ for $i = 1, 2, ..., U$, and verifies whether $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}}_i \|_2^2 \leq D_{th}$ for some i.

We first test whether the approximation resulting from JL embedding compromises the accuracy of matching faces based on the ℓ_1 distance between their corresponding integer feature vectors. To perform this test, we used faces from the Multiple Biometric Grand Challenge (MBGC) database [10] from NIST. Note that, even though the underlying data is a public biometric database, we are using it for a database querying application and not for biometric authentication. Thus, when considering potential attacks on the system, we should be concerned with attempts to discover $\mathbf{x}$ and $\mathbf{y}$, not with attempts to gain unauthorized entry by subverting a biometric access control system.

To facilitate querying, integer feature vectors with length $n = 900$ are extracted from two-dimensional Haar-like features applied to the face images using the algorithm proposed in [11]. Owing to large differences in illumination, pose and expression, the MBGC dataset is known to be a very challenging dataset for face recognition with most feature extraction algorithms, including the one that we adopt for these simulations. However, for the purposes of this paper, we are not concerned with the goodness of a particular feature extraction algorithm; our objective is only to test whether our approximation via JL embedding preserves the fidelity of the feature matching algorithm. To that end, Fig. 2(a) plots the intra-user and inter-user distances calculated between pairs of integer feature vectors for 100 users, each having between 2 and 20 faces. The solid lines depict the distribution of the exact ℓ_1 intra-user and inter-user distances while the dashed lines depict the distribution of the respective approximate distances using our protocol. It is clear that the distributions are almost identical, suggesting that the approximation is accurate. The distributions are nearly Gaussian as a consequence of the Central Limit Theorem; the individual face features are nearly independent and uniformly distributed in [0, 255], so the calculation of the ℓ_1 distance results in addition of a large number of nearly i.i.d random variables. A histogram of the approximation error has been plotted in Fig. 2(b) for 6000 randomly chosen pairs of faces. The mean and standard deviation are both very small, providing further evidence that the approximations are very accurate.

Having verified the accuracy of our approximation scheme, we now turn to secure calculation of $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2$. This is done according to the protocol described in Section 3. Both parties use Paillier homomorphic encryption with encryption parameters p, q and g. Thus, the public encryption key is $(N = pq, g)$ and the private decryption key is (p, q). We used 100-bit prime numbers[4] for p and q but they could be larger if higher computational security is desired. As noted earlier, the encryption key is known to both Alice and Bob, but the decryption key is known only to the Alice. We verified that the squared ℓ_2 distance calculated using the protocol is exactly $\| \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \|_2^2$. Now, for the final

[3]While we are concerned only with privacy-preserving querying, many further actions are possible in the event of a successful query. One example is that, if a query is successful, Alice and Bob may later decide to share the relevant images.

[4]For e.g., $p = 1267650600228229401496704256919$, $q = 1267650600228229401496705310003$, $g = 2$.

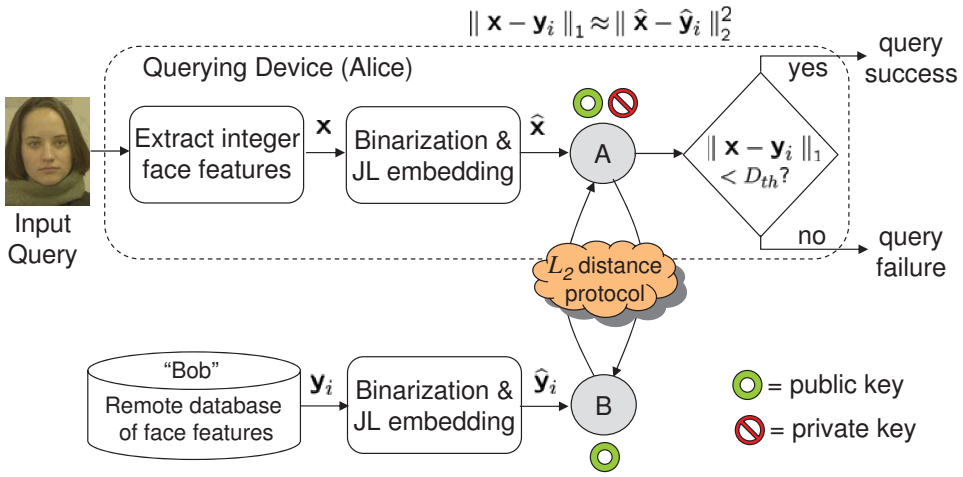


Fig. 1. A querying system based on ℓ_1 distance between integer face feature vectors. The querying device (Alice) determines whether an input image is similar to one or more images in the database (Bob) without revealing the query and without knowing the images in the database.

step in querying, it is left to choose a value for the distortion threshold D_{th}. This depends on the distribution of intra-user and inter-user distances in Fig. 2 and the desired querying accuracy. Clearly, the overlap between the distributions leads to false positives and false negatives. Since our approximation is very accurate, the probability of false positives and false negatives depends only on the underlying feature extraction algorithm, which is not the focus of this paper.

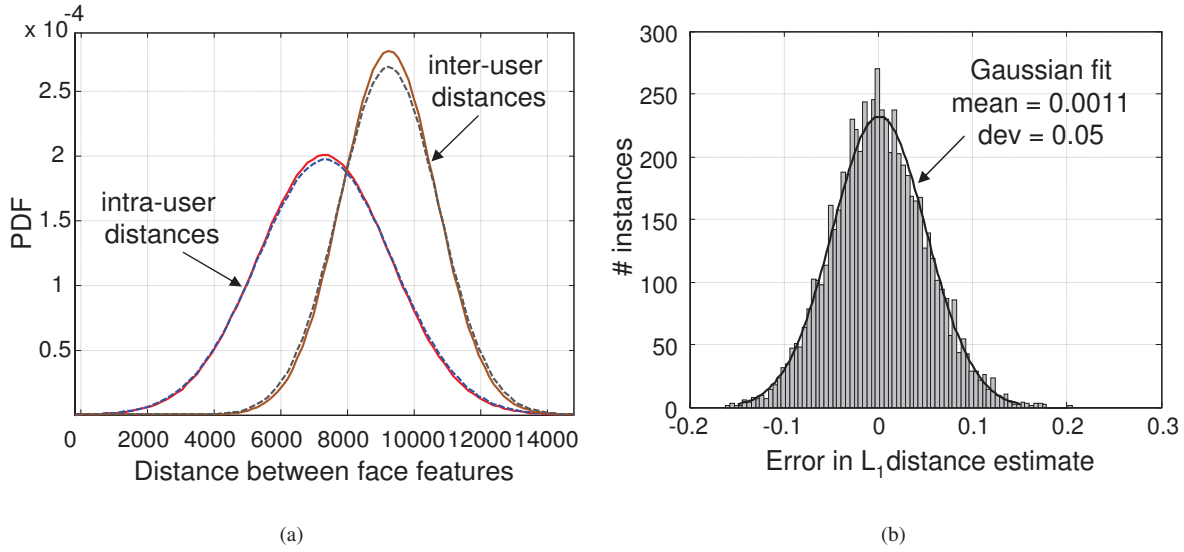
It is more pertinent to compute the communication overhead incurred by our protocol. The original length of the feature vectors possessed by Alice and Bob is $n = 900$. After binarization, this results in binary vectors of length $(M-1)n = 255 \times 900 = 229500$. JL embedding of the binary vectors results in integer vectors of length $O(\log M^n)$. Using base-2 logarithms and absorbing the conversion into the multiplicative constant for the order, we choose to embed the binary vectors into an ℓ_2 space of dimension $\log_2 M^n = 900 \log_2 256 = 7200$. Finally, in the secure ℓ_2 distance protocol, Alice transmits a maximum of $7200 \times \log N^2 = 7200 \times 400 \approx 360$ kilobytes to Bob, where $N = pq$ is a 200-bit number. Then, for each user i in the database, Bob transmits $\log N^2$ bits to Alice. Thus, the total communication overhead for Bob is $U \log N^2 = 100 \times 400 \approx 5$ kilobytes. If greater computational security is desired, N is increased, thereby increasing the computational overhead for Alice and Bob. Also, if an even closer approximation to the original ℓ_1 distance is desired, the JL embedding can embed into a ℓ_2 space of larger dimension, which would also increase the communication overhead for Alice but not for Bob. This may be important because the database (Bob) could conceivably be handling requests from multiple querying devices ("Alices") at the same time.

At the end of the protocol, Bob only has access to encrypted integers $\tilde{x}_j, j = 1, 2, ..., 7200$ from Alice, which he cannot decrypt because he does not possess the private key. Bob does not even find out whether the query succeeded or failed. Alice only has access to the approximate distances $\| \hat{\mathbf{x}} - \hat{\mathbf{y}}_i \|_2^2$, but does not know anything else about the $\mathbf{y}_i, i = 1, 2, ..., 100$. Since we consider honest but curious parties, we have not explicitly addressed the case in which Alice and/or Bob use the malleability of homomorphic encryption to modify the encrypted transmissions and force incorrect results from the protocol. A feature of the proposed protocol is that, even though such malicious actions can result in erroneous ℓ_1 distances, they cannot compromise the privacy of the innocent party.

We have presented one realization of privacy-preserving querying based on ℓ_1 and ℓ_2 distances, and many refinements are possible that are outside our current scope. For example, it may be desirable to protect Bob's database from repeated queries by Alice because such repeated queries would allow Alice to gain some information about the distribution of faces in Bob's database. In such cases, it is not advisable to present Alice with a vector containing distance values $\| \hat{\mathbf{x}} - \hat{\mathbf{y}}_i \|_2^2$ for all i. Instead, Alice and Bob could engage in a minimum-finding protocol, at the end of which, only the smallest distance $d = \min_i \| \hat{\mathbf{x}} - \hat{\mathbf{y}}_i \|_2^2 \approx \min_i \| \mathbf{x} - \mathbf{y}_i \|_1$ corresponding to the closest face is disclosed to Alice. Then, Alice's query is successful if and only if $d \leq D_{th}$.

6. SUMMARY

A two-party protocol for privacy-preserving approximation of ℓ_1 distance was presented in this paper. Using Johnson-Lindenstrauss embeddings, the two parties map their vectors $\mathbf{x}$ and $\mathbf{y}$ into new vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ such that $\| \hat{\mathbf{x}} - \hat{\mathbf{y}} \|_2^2 \approx \| \mathbf{x} - \mathbf{y} \|_1$. Then, they use Paillier homomorphic encryption for secure computation of $\| \hat{\mathbf{x}} - \hat{\mathbf{y}} \|_2^2$. The security and communication overhead of the two untrusting parties is analyzed. The approximation protocol is employed in a private face querying system, where a querying device can interact with a remote database without revealing its input image, and none of the face features



(a) (b)

Fig. 2. (a) The probability distributions of the inter-user and intra-user distances calculated using the proposed approximation scheme are very close to the true probability distributions.(b) A histogram of the error between the actual ℓ_1 distance and its secure approximation computed for 6000 randomly chosen pairs. The mean error is 0.0011 and its standard deviation is 0.05.

in the database are compromised. In experiments with a public database, it is found that the protocol closely approximates the ℓ_1 distance while preserving the privacy and security requirements.

7. ACKNOWLEDGEMENTS

The authors are grateful to Petros Boufounos for introducing them to the JL Lemma, and to Kuntal Sengupta and Mike Jones for providing the face feature extraction software.

8. REFERENCES

[1] W. Du, M. Atallah, and F. Kerschbaum, "Protocols for secure remote database access with approximate matching," in *Seventh ACM Conference on Computer and Communications Security*, 2000, pp. 523–540.

[2] J. Feigenbaum, S. Kannan, M. J. Strauss, and M. Viswanathan, "An approximate l^1-difference algorithm for massive data streams," *SIAM J. COMPUT.*, vol. 32, no. 1, pp. 131–151, 2002.

[3] A. C-C. Yao, "How to Generate and Exchange Secrets," in *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (SFCS)*, Washington, DC, USA, 1986, pp. 162–167, IEEE Computer Society.

[4] P. Indyk, " Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation," in *IEEE Symposium on Foundations of Computer Science*, Redondo Beach, CA, Nov. 2000, pp. 189–197.

[5] W. Lu, A. Varna, and M. Wu, "Secure Image Retrieval through Feature Detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, 2009.

[6] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz Mapping Into Hilbert Space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[7] D. Achlioptas, "Database-friendly Random Projections: Johnson-lindenstrauss With Binary Coins," *Journal of Computer and System Sciences*, vol. 66, pp. 671–687, 2003.

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

[9] S. Rane, W. Sun, and A. Vetro, "Secure Distortion Computation in the Encrypted Domain Using Homomorphic Encryption," in *Proc. IEEE International Conference on Image Processing*, Cairo, Egypt, Oct. 2009.

[10] National Institute of Standards and Technology (NIST), "Multiple biometric grand challenge database," http://face.nist.gov/mbgc/.

[11] M. Jones and P. Viola, "Face recognition using boosted local features," *MERL Technical Report, TR2003-25*, May 2003.