Privacy-Preserving Computation of the Earth Mover's Distance

Alberto Blanco-Justicia[0000−0002−1108−8082] and Josep Domingo-Ferrer[0000−0001−7213−4962]

Universitat Rovira i Virgili
Departament of Computer Engineering and Mathematics
UNESCO Chair in Data Privacy
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, Tarragona, Catalonia
{alberto.blanco,josep.domingo}@urv.cat

Abstract. The Wasserstein distance, also known in computer science as the *Earth Mover's Distance* (EMD) is a distance metric between two probability distributions. EMD has often been used as a distance metric to compare images and documents, and is central to privacy models such as t-closeness. In this work, we show that, given one-dimensional discrete probability distributions, the computation of EMD can be reduced to the computation of the cardinality of the intersection of two sets. We then use a private matching scheme to create a privacy-preserving computation protocol for EMD: two parties can compute EMD between their privately-owned documents without revealing them to the other party. We demonstrate our proposal by implementing a privacy-preserving reverse image search, where images are kept encrypted at an external server.

Keywords: Secure Multiparty Computation · Private Matching Schemes · Searchable Encryption · Earth Mover's Distance.

1 Introduction

The Wasserstein distance, better known in computer science as the Earth Mover's Distance (EMD), is a distance metric between two distributions that measures the minimum amount of work needed to transform one distribution into the other. If we imagine distributions as heaps of earth —hence the name of the distance—, the cost of transforming one heap of earth into another is equal to the amount of earth we need to move times the distance we need to move it [18]. EMD is a special case of a transportation optimization problem and, for discrete one-dimensional distributions (such as a histogram), there exist efficient algorithms to compute it.

EMD can be used to compare images according to their distribution of colors and textures [18,19], and to semantically compare documents [22,14]. Thus, EMD can be used as a building block to develop databases with reverse search support. EMD has also been used in microdata anonymization: the t-closeness privacy model [15], which is an extension of k-anonymity [20], requires confidential attributes to have a distribution

within each k-anonymous class that is close to their distribution on the whole microdata file. This closeness is measured as the EMD.

In this work, we propose a mechanism for privacy-preserving computation of EMD. Two individuals, each of them holding a private file, want to compute the EMD of their files without any of them revealing their own file. Our proposed mechanism is based on two-party computation of the set intersection cardinality [9]. It might be used as a building block for the implementation of searchable encryption schemes for reverse image and document search or to develop distributed anonymization mechanisms.

In Section 2, we present the Earth Mover's Distance and give an overview of methods for secure multiparty computation of the cardinality of the intersection of private sets. In Section 3 we demonstrate that, for discrete one-dimensional distributions, the computation of EMD can be reduced to the computation of the cardinality of the union or the intersection of two sets. Section 4 presents our experimental work, in which we apply our EMD computation mechanism to a reverse image search service for encrypted outsourced images. In Section 5 we summarize the conclusions of the work.

2 Background

2.1 The Earth Mover's Distance

Given two probability distributions A and B, EMD measures the distance between such distributions [18]. In the general case, EMD can be obtained by solving a transportation optimization problem. In the special case of discrete one-dimensional distributions, or relative frequency histograms $A = \{a_0, \ldots, a_{n-1}\}$ and $B = \{b_0, \ldots, b_{n-1}\}$, EMD can be iteratively computed using Algorithm 1.

Algorithm 1: Earth Mover's Distance between two one-dimensional discrete distributions

Input: $A = \{a_0, \ldots, a_{n-1}\}$, $B = \{b_0, \ldots, b_{n-1}\}$
Output: $\text{EMD}(A, B)$

$w_0 = 0$
for $i = 1 \rightarrow n$ **do**
  $w_i = a_{i-1} - b_{i-1} + w_{i-1}$
end
return $\sum_{i=0}^{n} |w_i|$

2.2 Private computation of the size of the intersection of two sets

In this section we present an overview of secure multiparty computation protocols for computing the cardinality of the intersection of privately held sets. These mechanisms are called *Private Set Intersection – Cardinality* (PSI-CA) and are a recurring primitive for secure protocols in privacy-preserving data mining. Their main goal is to allow data

sharing among entities that do not fully trust each other. Instead of sharing all data held by each of them, the entities can start by sharing information about the data they have in common.

More formally, parties P_1 and P_2, with private inputs x and y, respectively, want to compute a function $f(x, y)$ without revealing any information on x, resp. y, to the other party. In our case, the private inputs x and y are sets of elements in a common universe, and the function f is $|x \cap y|$. A typical security requirement on PSI-CA (but also on general multiparty computation schemes) is that parties P_1 and P_2 do not learn more information about the other party's input than the information they would obtain if a trusted third party collected their inputs, computed f and distributed the output to the participating parties.

An example application of PSI-CA would be several institutions co-operating in searching and detecting fraudulent financial movements of the customers they have in common. Other uses of PSI-CA include searching keywords in outsourced document databases or matching social medial users according to their shared contacts.

One of the most influential works on PSI-CA is that of Freedman, Nissim and Pinkas [9]. This work presents several secure protocols to compute different set operations, including the set intersection cardinality. All of them are based on the same primitive, namely oblivious polynomial evaluation (OPE). One of the parties builds a polynomial whose roots are the elements of her privately owned set. The other party evaluates such a polynomial for all elements in her own set. All elements whose evaluation is 0 are elements that are shared by both parties. Using additively-homomorphic encryption (*e.g.* the Paillier cryptosystem [16]), the second party can evaluate the polynomial from its encrypted coefficients. Unfortunately, these protocols require the computation of expensive cryptographic operations and are not suitable for large datasets, although [10] describes some optimizations.

Another frequently used construction in PSI-CA protocols are Bloom filters. Dong, Chen and Wen [6], use modified Bloom filters and oblivious transfer protocols to compute multi-set operations. Pinkas, Schneider and Zohner [17] proposed a scalability improvement of this protocol.

PSI-CA based on oblivious polynomial evaluation. Next we briefly recall the computation of the size of the intersection of two sets using oblivious polynomial evaluation, as per [9]. Let $\mathcal{C}$ and $\mathcal{S}$ be the two participants in the protocol, with $\mathcal{C}$ bringing in the set $A = \{a_0, \ldots, a_{n-1}\}$ and $\mathcal{S}$ the set $B = \{b_0, \ldots, b_{m-1}\}$. Let Enc be the encryption under the Paillier cryptosystem with $\mathcal{C}$'s public key. The Paillier cryptosystem is additively homomorphic, that is, it satisfies the following two properties:

$$Enc(x) \cdot Enc(y) = Enc(x + y);$$
$$Enc(x)^k = Enc(kx).$$

Protocol 1 *PSI-CA-OPE*

1. *$\mathcal{C}$ computes* $P(x) = \sum_{i=0}^{n} p_i x^i = \prod_{j=0}^{n-1}(x - a_i)$.
2. *$\mathcal{C}$ sends* $Enc(p_0), \ldots, Enc(p_n)$ *to* $\mathcal{S}$.

3. *$\mathcal{S}$ chooses random integers $r_j \in \mathbb{Z}_n$ for $1 \leq j \leq m-1$. $\mathcal{S}$ computes $Enc(r_j \cdot p(b_j) + k)$ for all $1 \leq j \leq |B|$. Next, $\mathcal{S}$ returns these ciphertexts to $\mathcal{C}$.*
4. *$\mathcal{C}$ decrypts the obtained ciphertexts. The result of each decryption is either k or a random number.*
5. *The number of obtained k's equals $|A \cap B|$.*

This protocol is secure against semi-honest adversaries. After this protocol is executed, $\mathcal{C}$ obtains $|B|$ from the number of ciphertexts received in Step 3, $\mathcal{S}$ obtains $|A|$ from the degree n of $P(x)$, and $\mathcal{C}$ obtains $|A \cap B|$. For $\mathcal{S}$ to also obtain $|A \cap B|$, the protocol has to be run a second time, with $\mathcal{C}$ and $\mathcal{S}$ inverting their roles.

PSI-CA based on Bloom filters. We summarize here the properties of Bloom filters and how they are used as a building block of PSI-CA protocols. Bloom filters [2] are a probabilistic data structure used to encode sets that allows membership queries. Membership queries for elements in the original set will test positive with 100% probability, while for elements not in the set there is a fixed probability of false positives.

A Bloom filter is a vector $B \in \mathbb{Z}_2^m$, with all its elements initialized to 0, and equipped with $k \ll m$ hash functions $H_i : \{0,1\}^* \rightarrow \{0, \ldots, m-1\}$. To insert an element e, we compute a list of indices $(i_0, \ldots, i_{k-1}) = (H_0(e), \ldots, H_{k-1}(e))$ and assign a value 1 to the corresponding positions in B. Similarly, a membership query for element e is resolved by checking whether the corresponding k positions in the vector are all set to 1. The false positive probability for membership queries in a Bloom filter containing n elements is given by the expression $(1-(1-\frac{1}{m})^{kn})^k$, which is the probability that the k indices of the element e that is tested are set to 1 when the element is not encoded in the filter. This may happen if an element $e' \neq e$ shares the same indices of e (which is unlikely if cryptographic hashes are used) or if a combination of other elements set these indices to 1.

The approximate number of elements inserted in a Bloom filter is

$$|A| \approx -\frac{m}{k} \ln \left(1 - \frac{\mathcal{H}(B_A)}{m}\right), \tag{1}$$

where A is the encoded set, B_A is the Bloom filter containing the elements of A, $\mathcal{H}(\cdot)$ is the Hamming weight, m is the length of the Bloom filter, and k is the number of hash functions.

A Bloom filter encoding the union, resp. the intersection, of two (or more) sets can be easily obtained by applying the bitwise $\vee$ (logic "or"), resp. $\wedge$ (logic "and") operation. By applying Expression (1) to the resulting Bloom filters, we obtain an estimate of the cardinality of the union, resp. the intersection, of the sets.

Bloom filters, by themselves, do not offer enough protection to the encoded elements to be directly used in PSI-CA protocols. If both parties share the same universe of elements, any of them might obtain all elements from the other party's Bloom filter with a brute force attack. Using oblivious transfer protocols, one of the parties can request the other party for all indices that are set to 1 in her own Bloom filter and thus obtain the intersection of the sets [6]. An alternative option is to encrypt all the Bloom

filter positions with a cryptosystem which is homomorphic for the binary sum, such as Goldwasser-Micali, and sum the filters index-wise. Positions where both filters agree will result in 0s and the rest as 1s [13]. Another option is to alter the Bloom filter's values with some probability, as in randomized response [7]. Finally, using keyed hash functions, such as HMAC, we can control who is able to insert elements into the Bloom filter and who is able to query it, which allows us to build indices for symmetric searchable encryption schemes as we do in our experiments.

3 The EMD as a PSI-CA problem

In this section, we show how the computation of EMD between two discrete one-dimensional distributions A and B can be reduced to operations between the cardinality of two sets that encode such distributions and the cardinality of their intersection. This reduction will allow us to describe a protocol for secure two-party computation of the EMD. To this end, we first show that Algorithm 1 is equivalent to the Manhattan distance between two cumulative distributions.

Theorem 1. *Let $\tilde{A}$ (resp. $\tilde{B}$) be the cumulative distribution, or cumulative sum of A (resp. B). Then, the distance $EMD(A, B)$ can be computed as the 1-norm, or Manhattan distance, of $\tilde{A}$ and $\tilde{B}$:*

$$EMD(A, B) = \|\tilde{A} - \tilde{B}\|_1. \tag{2}$$

Proof. Given the discrete one-dimensional distribution $A = \{a_0, \ldots, a_{n-1}\}$, the cumulative distribution of A is

$$\tilde{A} = \{\tilde{a}_i = \sum_{j=0}^{i} a_j \ : \ 0 \leq i \leq n-1\}. \tag{3}$$

On the other hand, the 1-norm or Manhattan distance is defined as

$$\|A - B\|_1 = \sum_{i=0}^{n-1} |a_i - b_i| \,. \tag{4}$$

By developing line 3 of Algorithm 1, which refers to the computation of w_i, and reordering the factors, we obtain for $i > 0$ and with $w_0 = 0$

$$\begin{aligned} w_i &= a_{i-1} - b_{i-1} + w_{i-1} \\ &= a_{i-1} - b_{i-1} + a_{i-2} - b_{i-2} + w_{i-2} \\ &\cdots \\ &= a_{i-1} - b_{i-1} + a_{i-2} - b_{i-2} + \cdots + a_0 - b_0 \\ &= (a_{i-1} + \cdots + a_0) - (b_{i-1} + \cdots + b_0) \\ &= \sum_{j=0}^{i-1} a_j - \sum_{j=0}^{i-1} b_j. \end{aligned} \tag{5}$$

According to line 5 of Algorithm 1 and using Expressions (3), (4) and (5), we obtain

$$\begin{aligned} EMD(A,B) &= \sum_{i=0}^{n} |w_i| = w_0 + \sum_{i=1}^{n} |w_i| = \sum_{i=1}^{n} |w_i| \\ &\overset{(5)}{=} \sum_{i=1}^{n} \left| \sum_{j=0}^{i-1} a_j - \sum_{j=0}^{i-1} b_j \right| \overset{(3)}{=} \sum_{i=1}^{n} |\tilde{a}_{i-1} - \tilde{b}_{i-1}| \\ &= \sum_{i=0}^{n-1} |\tilde{a}_i - \tilde{b}_i| \overset{(4)}{=} \|\tilde{A} - \tilde{B}\|_1. \end{aligned}$$

□

Next, we prove that the Manhattan distance can be obtained from the cardinality of the intersection of two sets. This result comes from [1], and we elaborate on it.

First, we define an encoding function f, which takes a list $A = \{a_0, \ldots, a_{n-1}\}$ of nonnegative integers and returns a set of the form:

$$\mathcal{A} = f(A) = \{(i,j) : a_i > 0,\ 1 \leq j \leq a_i\}.$$

For example, given the list $A = \{2,1,3\}$, the resulting set $\mathcal{A}$ is $\{(1,1),(1,2),(2,1),(3,1),(3,2),(3,3)\}$. Figure 1 shows a graphical example of the encoding function.

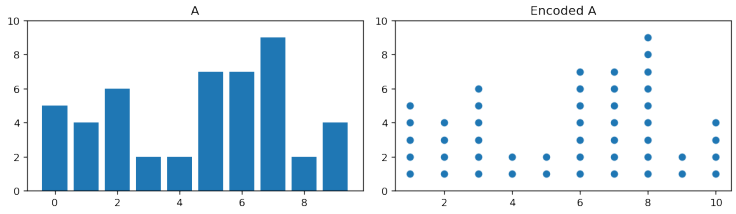


Fig. 1. The encoding function f encodes the histogram A as a set of two-dimensional points $\mathcal{A}$

Theorem 2. *Given the encoding function f, two lists A and B containing the same number n of nonnegative integers, and their respective encodings $\mathcal{A}$ and $\mathcal{B}$, the following holds:*

$$\|A - B\|_1 = |\mathcal{A}| + |\mathcal{B}| - 2|\mathcal{A} \cap \mathcal{B}|.$$

Proof. First, we rewrite the expression of the Manhattan distance between lists A and B as the difference between the sum of the element-wise maximum values minus the

sum of the element-wise minimum values of A and B:

$$\begin{aligned}\|A-B\|_1 &= \sum_{i=0}^{n-1} |a_i - b_i| \\ &= \sum_{i=0}^{n-1} \max(a_i, b_i) - \min(a_i, b_i) \\ &= \sum_{i=0}^{n-1} \max(a_i, b_i) - \sum_{i=0}^{n-1} \min(a_i, b_i).\end{aligned}$$

We also observe that the cardinality of the set resulting from applying the encoding function f to a list equals the sum of the values of such list:

$$|\mathcal{A}| = \sum_{i=0}^{n-1} a_i.$$

The encoding function f also has the following property: the size of the union of the encodings of A and B is the sum of the element-wise maximum values of A and B:

$$\begin{aligned}|\mathcal{A} \cup \mathcal{B}| &= |\{(i,j) : (i,j) \in \mathcal{A} \text{ or } (i,j) \in \mathcal{B}\}| \\ &= |\{(i,j) : a_i, b_i > 0,\ 1 \leq j \leq \max(a_i, b_i)\}| \\ &= \sum_{i=0}^{n-1} \max(a_i, b_i).\end{aligned}$$

Likewise, the size of the intersection equals the sum of the element-wise minimum values:

$$\begin{aligned}|\mathcal{A} \cap \mathcal{B}| &= |\{(i,j) : (i,j) \in \mathcal{A} \text{ and } (i,j) \in \mathcal{B}\}| \\ &= |\{(i,j) : a_i, b_i > 0,\ 1 \leq j \leq \min(a_i, b_i)\}| \\ &= \sum_{i=0}^{n-1} \min(a_i, b_i).\end{aligned}$$

Figure 2 shows a graphical example of these union and intersection properties. In the top row, we show the encodings of histograms A and B. In the bottom row, we show the sets representing the intersection and the union of the encoded datasets. It can be observed that the resulting sets represent the element-wise minimum and maximum, respectively, of the two histograms.

Therefore, the Manhattan distance between A and B results from the difference $|\mathcal{A} \cup \mathcal{B}| - |\mathcal{A} \cap \mathcal{B}|$. Given the identity $|\mathcal{A} \cup \mathcal{B}| = |\mathcal{A}| + |\mathcal{B}| - |\mathcal{A} \cap \mathcal{B}|$, this results in

$$\|A-B\|_1 = |\mathcal{A}| + |\mathcal{B}| - 2|\mathcal{A} \cap \mathcal{B}|.$$

□

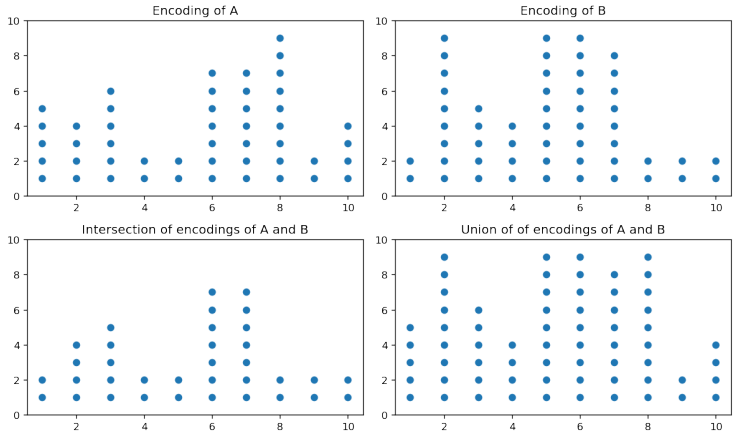


Fig. 2. The intersection and union of $\mathcal{A}$ and $\mathcal{B}$ result in the element-wise minimum and maximum, respectively, of A and B

From Theorems 1 and 2 it follows that the computation of EMD can be expressed as

$$EMD(A, B) = |\tilde{\mathcal{A}}| + |\tilde{\mathcal{B}}| - 2|\tilde{\mathcal{A}} \cap \tilde{\mathcal{B}}|. \tag{6}$$

In Expression (6), $|\tilde{\mathcal{A}}|$, $|\tilde{\mathcal{B}}|$ and $|\tilde{\mathcal{A}} \cap \tilde{\mathcal{B}}|$ can be privately computed by both parties holding A and B, respectively, using the PSI-CA protocols discussed in Section 2.2, in particular Protocol 1.

3.1 Message expansion

Let the size of set A be $|A| = n$ and its sum be $s = \sum_{i=0}^{n-1} a_i$. The message expansion of set $\tilde{\mathcal{A}}$, resulting from computing the cumulative sum $\tilde{A}$ and encoding it using function f is:

- In the best case, when $a_0 = \cdots = a_{n-2} = 0$ and $a_{n-1} = s$, $|\tilde{\mathcal{A}}| = s$;
- In the worst, case, when $a_0 = s$ and $a_1, \cdots, a_{n-1} = 0$, $|\tilde{\mathcal{A}}| = sn$;
- In the average case $|\tilde{\mathcal{A}}| = (1 + n)s/2$.

Thus, the spatial complexity of our proposal is $\mathcal{O}(sn)$.

4 Experimental results

In our experimental work, we simulated a symmetric searchable encryption (SSE) scheme that allows the data owner to conduct reverse image search queries on an outsourced database of encrypted images. We use our proposed scheme to privately compute the EMD between two distributions to build the search index of an SSE scheme. Typical SSE schemes consist of two parts: on the one hand, the information is encrypted using some standard symmetric encryption scheme (*e.g.* AES), and, on the other hand, a

search index is built such that only the parties holding some private information (*i.e.* a trapdoor generated with the symmetric secret key) can query it [5,4,12,21,3]. In this experimental work we focus on the search index to demonstrate the private computation of the EMD.

The experiments were implemented in Python in a Jupyter Notebook, which is available in Github[1]. The images for the experiments were obtained from [8], which contains 9144 images of 101 different categories.

We took half of the images (4572) from the database to build our search index. For each image, we obtained its gray level histogram with $n = 16$ bins of 16 values each, and we re-scaled it so that its cumulative sum was $s = 100$. The maximum number of elements to be included in each of the Bloom filters after encoding the histograms using function f was, as per Section 3.1, $sn = 1600$. The implementation of the Bloom filters used keyed hash functions (HMAC) so that only those who owned the key were able to add images to the database and query it. To query the database, we encoded the image to be searched in the same way as the index images, and we computed its EMD with respect to all images in the search index.

We first studied the impact of the Bloom filters' false positive rate on the overall performance of the reverse image search service. The optimal values for m and k can be computed as a function of the maximum number sn of allowed item insertions, and a fixed false positive rate ρ [11]:

$$m = -\frac{sn \ln \rho}{(\ln 2)^2}, \qquad k = \frac{m}{sn} \ln 2. \tag{7}$$

For a maximum number of $sn = 1600$ elements, the optimal bit-length of the Bloom filters m and number of hashes k with respect to the false positive rate ρ, going from $\rho = 0.0001$ to $\rho = 0.5$, are shown in Figure 3.

The false positive rate, thus, directly affected the size of the search index, from $10.06MB$ for a false positive rate of 50% to $133.73MB$ for a false positive rate of 0.01%. Next, Figure 4 shows how the choice of ρ affected the time to build the search index and the individual query times.

Some error in the computed EMD was to be expected for the following reasons. On the one hand, we forced all gray level histograms to have the same cumulative sum and we could only operate on integer values. Therefore, some error had to come from the rounding of values. On the other hand, the false positive rate of the Bloom filters affects the collisions in the membership queries, and therefore also the computation of the intersection, the estimation of encoded elements and, thus, the computation of the EMD. Figure 5 shows the Mean Absolute Error of the EMD computed on the cleartext images versus the EMD computed using our mechanism for different values of ρ.

The pairwise distances of the first 1024 images in the search index, computed in the clear and using our method with Bloom filters with $m = 8000$, $k = 4$ and $\rho \leq 0.09$ are shown as distance matrices in Figures 6 and 7.

Next, we generated queries for all 9144 images in the original database against the search index including 4572 images. All images whose distance was less than a

[1] `https://github.com/ablancoj/privateEMD/blob/master/EMD%20Bloom%20images.ipynb`

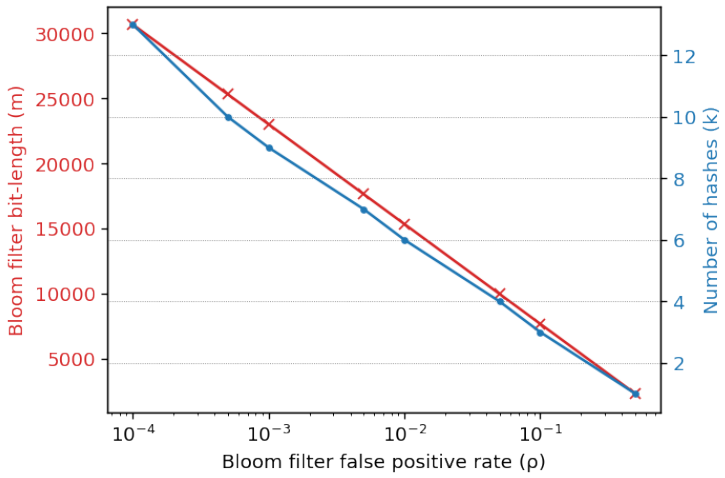


Fig. 3. Size of the Bloom filters (m) and number of hashes (k) as a function of the false positive rate, for a fixed maximum number of elements equal to 1600

threshold 10 were returned as the result. All search queries for the 4572 images in the database returned the correct result, although some queries returned more than one result. Figure 8 shows one of such false positives.

Regarding the 4572 images *not* included in the database, a few queries returned a result. Figure 9 shows one such false positive.

Figure 10 shows the ratio of images not in the search index that returned some result and the ratio of images in the search index that returned more than one result, as a function of the Bloom filter's false positive rate ρ.

Additionally, we generated queries on modified images. In particular we applied random anisotropic scaling to images (randomly scaling each of the axes by $x0.5$ to $x1.5$) both in the search index and out of it before querying for them. In this scenario, we built a search index consisting of Bloom filters with $\rho = 0.0001$, $m = 30672$, and $k = 13$. The number of bins $n = 16$ and maximum cumulative sum $s = 100$ remained the same. The distance threshold to accept an image in the index as a result was set to 15. The results were as follows:

- For images in the search index, 42.5% returned a correct result.
- For images not in the search index, 8.9% returned some result.

Increasing the maximum cumulative sum to $s = 500$ (which increased the maximum number of elements to $sn = 8000$ and changed the Bloom filter parameters to $m = 115020$ and $k = 9$ for a false positive rate of $\rho = 0.001$) and increasing the distance threshold to 100 improved the matching results to 72.7% correct matches for images in the search index, but increased the false positives for images not in the search index to 21%, at the additional cost of increasing the search index size to $500MB$ and the query time to 381 ms.

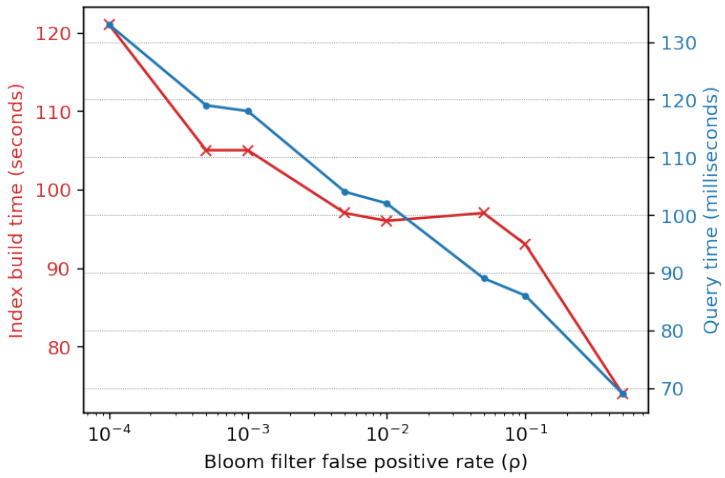


Fig. 4. Index build time and image query time as a function of the false positive rate, for a fixed maximum number of elements equal to 1600

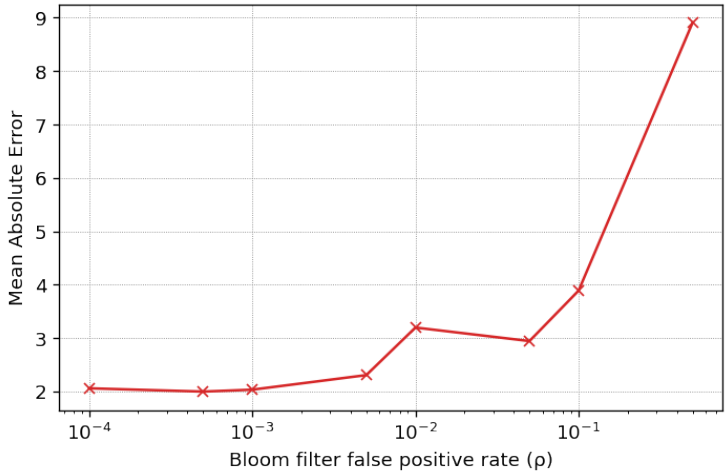


Fig. 5. Mean Absolute Error as a function of the false positive rate, for a fixed maximum number of elements equal to 1600.

5 Conclusions

In this work we have proposed a secure multiparty computation scheme to calculate the Earth Mover's distance between one-dimensional discrete distributions. To do so, we have proved that computing EMD in such cases can be reduced to finding the size of the intersection of sets encoding these distributions. From these results, we propose to use existing Private Set Intersection—Cardinality schemes to obtain EMD between two

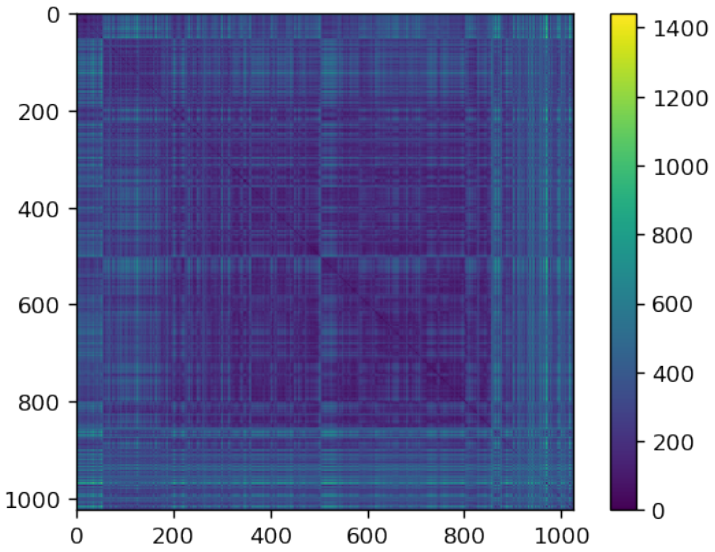


Fig. 6. Pairwise EMD computed from cleartext images

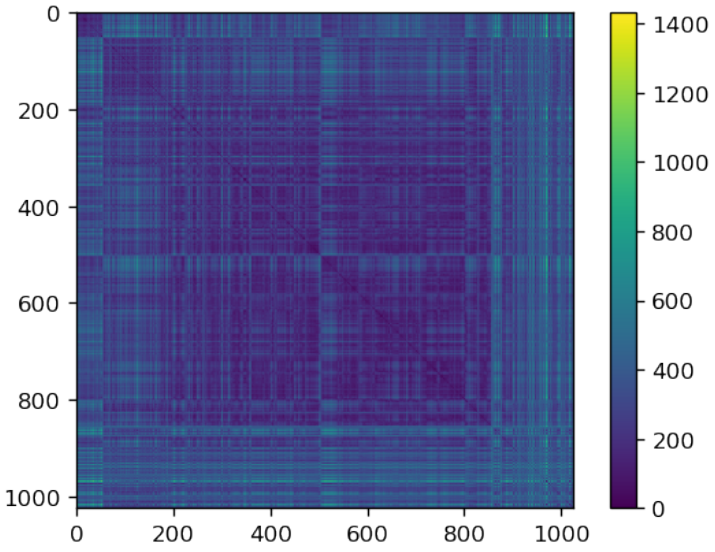


Fig. 7. Pairwise EMD computed using our mechanism

distributions. We have demonstrated our scheme with a symmetric searchable encryption scheme that supports reverse image search.

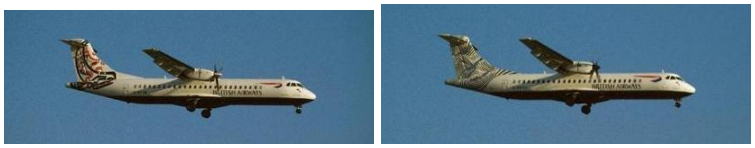


Fig. 8. False positive: querying for the left image returned both that image and the image on the right

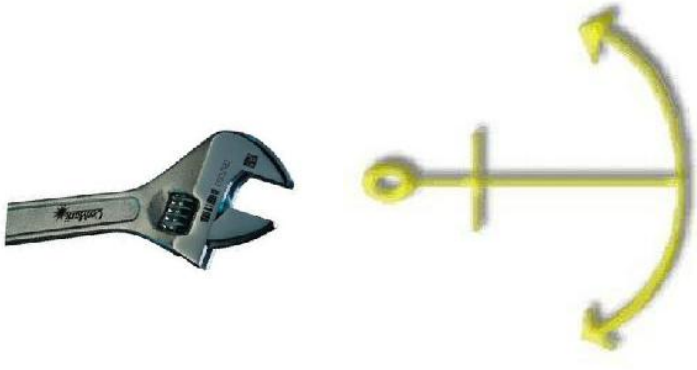


Fig. 9. False positive: querying for the image on the left resulted in the image on the right

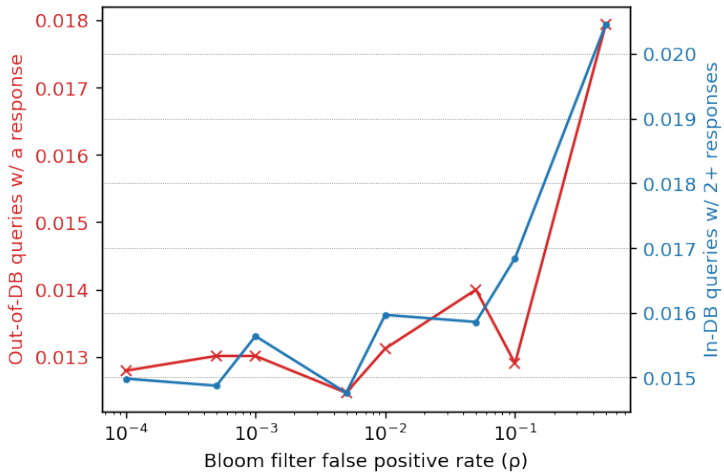


Fig. 10. Left: ratio of queries for images not in the search index that returned some result. Right: ratio of queries for images in the search index that returned more than one result.

Acknowledgements and disclaimer

We acknowledge support from: European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), Government of Catalonia (ICREA

Acadèmia Prize to the second author and grant 2017 SGR 705) and Spanish Government (project RTI2018-095094-B-C21). The authors are with the UNESCO Chair in Data Privacy, but their views here are not necessarily shared by UNESCO.

References

1. A. Blanco-Justicia, J. Domingo-Ferrer, O. Farràs, and D. Sánchez, "Distance computation between two private preference functions," in *IFIP International Information Security Conference*. Springer, 2014, pp. 460–470.
2. B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
3. R. Bost, and P.-A. Fouque, "Security-efficiency tradeoffs in searchable encryption," in *Proceedings of Privacy Enhancing-Technologies*, vol. 4, pp. 132–151, 2019.
4. R. Bost, P.-A. Fouque, and D. Pointcheval, "Verifiable dynamic symmetric searchable encryption: optimality and forward security," *IACR Cryptol. ePrint Arch*, 2016, 62.
5. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.
6. C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: an efficient and scalable protocol," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*, 2013, pp. 789–800.
7. Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14)*, 2014, pp. 1054–1067.
8. L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE, 2004, pp. 178–178.
9. M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology (EUROCRYPT 2004)*. Springer, 2004, pp. 1–19.
10. M. J. Freedman, C. Hazay, K. Nissim, and B. Pinkas, "Efficient set intersection with simulation-based security," *Journal of Cryptology*, vol. 29, no. 1, pp. 115–155, 2016.
11. T. Gerbet, A. Kumar, and C. Lauradoux, "The power of evil choices in Bloom filters," in *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2015)*. IEEE, 2015, pp. 101–112.
12. C. Guo, X. Chen, Y. Jie, F. Zhangjie, M. Li, and B. Feng, "Dynamic multi-phrase ranked search over encrypted data with symmetric searchable encryption," *IEEE Transactions on Services Computing*, 2017. Early access: `https://ieeexplore.ieee.org/abstract/document/8089767` Checked September 12, 2020.
13. F. Kerschbaum, "Outsourced private set intersection using homomorphic encryption," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*, 2012, pp. 85–86.
14. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International Conference on Machine Learning*, 2015, pp. 957–966.
15. N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.
16. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology (EUROCRYPT 1999)*. Springer, 1999, pp. 223–238.

17. B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on {OT} extension," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 797–812.
18. Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *6th International Conference on Computer Vision*. IEEE, 1998, pp. 59–66.
19. Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.
20. P. Samarati, and L. Sweeney, *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression,* Research Report, SRI International, 1998.
21. S.-F. Sun, X. Yuan, J.K. Liu, R. Steinfeld, A. Sakzad, V. Vo, and S. Nepal, "Practical backward-secure searchable encryption from symmetric puncturable encryption," In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 763–780, 2018.
22. X. Wan and Y. Peng, "The earth mover's distance as a semantic measure for document similarity," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 2005, pp. 301–302.