ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 2, Issue 2, pp: (114-120), Month: October 2015 – March 2016, Available at: www.paperpublications.org

Secure Encrypted Data in Cloud Based Environment

K.R.Raji[1], Ms.Archana.V.Nair.S[2], Ms.K.R.Raghi[3]

PG Scholar[1]Assistant Professor of IT Department[2], Ponjesly College of Engineering, Nagercoil -3, India

Abstract: **Data Mining has wide applications in many areas such as banking, medicine, scientific research and among government agencies. Classification is one of the commonly used tasks in data mining applications. The cloud computing, users have the opportunity to outsource their data, in encrypted form, as well as the data mining tasks to the cloud. Since the data on the cloud is in encrypted form, existing privacy preserving classification techniques are not applicable. On solving the classification problem over encrypted data. A secure k-NN classifier over encrypted data in the cloud. The k-NN protocol protects the confidentiality of the data, user's input query, and data access patterns. To develop a secure k-NN classifier over encrypted data under the standard semi-honest model. Also, we empirically analyze the efficiency of our solution through various experiments.**

Keywords: **Security, k-NN Classifier, Outsourced Databases, Encryption.**

I. INTRODUCTION

Cloud computing paradigm is revolutionizing the organizations' way of operating their data particularly in the way they store, access and process data. As an emerging computing paradigm, cloud computing attracts many organizations to consider seriously regarding cloud potential in terms of its cost-efficiency, flexibility, and offload of administrative overhead.. The advantages that the cloud offers, privacy and security issues in the cloud are preventing companies to utilize those advantages. When data are highly sensitive, the data need to be encrypted before outsourcing to the cloud. However, when data are encrypted, irrespective of the underlying encryption scheme, performing any data mining tasks becomes very challenging without ever decrypting the data. The cloud can also derive useful and sensitive information about the actual data items by observing the data access patterns even if the data are encrypted. Therefore, the privacy/security requirements of the DMED problem on a cloud are threefold: (1) confidentiality of the encrypted data, (2) confidentiality of a user's query record, and (3) hiding data access patterns. Existing work on Privacy-Preserving Data Mining (either perturbation or secure multi-party computation based approach) cannot solve the DMED problem. Perturbed data do not possess semantic security, so data perturbation techniques cannot be used to encrypt highly sensitive data. Also the perturbed data do not produce very accurate data mining results. Secure multi-party computation based approach assumes data are distributed and not encrypted at each participating party. As a result, a novel methods to effectively solve the DMED problem assuming that the encrypted data are outsourced to a cloud. Specifically, we focus on the classification problem since it is one of the most common data mining tasks. Because each classification technique has their own advantage, to be concrete, this paper concentrates on executing the k-nearest neighbor classification method over encrypted data in the cloud computing environment.

II. EXISTING SYSTEM

The existing secret sharing techniques in SMC develop a PPkNN protocol. However, our work is different from the secret sharing based solution from the following two aspects. (i) Solutions based on the secret sharing schemes require at least three parties whereas our work requires only two parties. (ii) Hiding data access patterns is still an unsolved problem in the secret sharing based schemes, whereas our work protects data access patterns from both participating parties.

2.1 Privacy-Preserving Data Mining (PPDM):

Privacy Preserving Data Mining (PPDM) is defined as the process of extracting/deriving the knowledge about data without compromising the privacy of data. In the past decade, many privacy-preserving classification techniques have been proposed in the literature in order to protect user privacy. Agrawal and Srikant, Lindell and Pinkas introduced the notion of privacy-preserving under data mining applications. In particular to privacy preserving classification, the goal is to build a classifier in order to predict the class label of input data record based on the distributed training dataset without compromising the privacy of data.

A. Data Perturbation Method:

In these methods, values of individual data records are perturbed by adding random noise in such way that the distribution of perturbed data look very different from that of actual data. After such a transformation, the perturbed data is sent to the miner to perform the desired data mining tasks. Data perturbation techniques cannot be applicable for semantically secure encrypted data. Also, they do not produce accurate data mining results due to the addition of statistical noises to the data.

B. Data Distribution Methods:

Assume the dataset is partitioned either horizontally or vertically and distributed across different parties. During this process, data owned by individual parties is not revealed to other parties. Classification is one important task in many applications of data mining such as health-care and business. In cloud computing, data owner outsources his/her data to the cloud. The direct way to guard the outsourced data is to apply encryption on the data before outsourcing. The hosted data on the cloud is in encrypted form in our problem domain, the existing privacy preserving classification techniques are not sufficient and applicable to PPkNN due to the following reasons. (i) In existing methods, the data are partitioned among at least two parties, whereas in our case encrypted data are hosted on the cloud. (ii) Since some amount of information is loss due to the addition of statistical noises in order to hide the sensitive attributes, the existing methods are not accurate. (iii) Leakage of data access patterns: the cloud can easily derive useful and sensitive information about users' data items by simply observing the database access patterns.

2.2. Query processing over encrypted data:

PPkNN is a more complex problem than the execution of simple kNN queries over encrypted data. One, the intermediate k-nearest neighbors in the classification process should not be disclosed to the cloud or any users. Secondly, even if we know the k-nearest neighbors, it is still very difficult to find the majority class label among these neighbors since they are encrypted at the first place to prevent the cloud from learning sensitive information. Third, the existing work did not address the access pattern issue which is a crucial privacy requirement from the user's perspective. A novel secure k-nearest neighbor query protocol over encrypted data that protects data confidentiality, user's query privacy, and hides data access patterns. First we introduced new security primitives, namely secure minimum (SMIN), secure minimum out of n numbers (SMINn), secure frequency (SF), and proposed new solutions for them. Second, the work did not provide any formal security analysis of the underlying sub-protocols. On the other hand, this paper provides formal security proofs of the underlying sub-protocols as well as the PPkNN protocol under the semi-honest model. Third, our preliminary work in addresses only secure kNN query which is similar to Stage 1 of PPkNN. However, Stage 2 in PPkNN is entirely new. Finally, our empirical are based on a real dataset whereas the results are based on a simulated dataset. As mentioned earlier, one can implement the proposed protocols under secret sharing schemes. In this work, we only concentrate on the two party situations; thus, we adopted the Paillier cryptosystem. Two-party and multi-party (three or more parties) SMC protocols are complement to each other, and their applications mainly depend on the number of available participants.

III. THE PROPOSED PROTOCOL

A novel privacy-preserving k-NN classification protocol, denoted by PPkNN, which is constructed using the protocols as building blocks. The goal of the PPkNN protocol is to classify users' query records using D′ in a privacy-preserving manner. Consider an authorized user Bob who wants to classify his query record q = (q_1,…, qm) based on D′ in C1. The proposed PPkNN protocol mainly consists of the following two stages:

• Stage 1 - Secure Retrieval of k-Nearest Neighbors (SRkNN):

In this stage, Bob initially sends his query q (in encrypted form) to C1. After this, C1 and C2 involve in a set of sub-protocols to securely retrieve (in encrypted form) the class labels corresponding to the k-nearest neighbors of the input query q. At the end of this step, encrypted class labels of k-nearest neighbors are known only to C1.

• Stage 2 - Secure Computation of Majority Class (SCMCk):

Following from Stage 1, C1 and C2 jointly compute the class label with a majority voting among the k-nearest neighbors of q. At the end of this step, only Bob knows the class label corresponding to his input query record q. The main steps involved in the proposed PPkNN protocol are as shown in Algorithm. We now explain each of the two stages in PPkNN in detail.

The Paillier cryptosystem is an additive homomorphic and probabilistic asymmetric encryption scheme. The encryption scheme is semantically secure. We simply use the well-known Paillier scheme in our implementations. However, to be more specific, in this paper we use the original Paillier cryptosystem.

Key generation : Randomly generate the prime numbers p and q.

Calculate : n = p*q and phi= (p-1)*(q-1)
Find : g=n+1 and λ=lcm (p-1,q-1)
$\mu=(L(g^{\lambda} \bmod n^2))^{-1} \bmod n$
Where,
L (u) = u-1/n
$u = g^{\lambda} \bmod n^2$

Encryption : $C=g^m.r^n \bmod n^2$
Where,
C=cipher text
m=message

Decryption : $D=Lt*\mu \bmod n$
Where,
$t=c^{\lambda} \bmod n^2$
Lt= (t-1)/n

3.1. Stage 2: Secure Computation of Majority Class (SCMCk):

Without loss of generality, suppose Alice's dataset D consists of w unique class labels denoted by $c = (c_1, . . c_w)$.

We assume that Alice outsources her list of encrypted classes to C1. That is, Alice outsources $(E_{pk}(c_1), . , E_{pk}(c_w))$ to C1 along with her encrypted database D′ during the data outsourcing step. Note that, for security reasons, Alice may add dummy categories into the list to protect the number of class labels, i.e., w from C1 and C2. However, for simplicity, we assume that Alice does not add any dummy categories to c. During Stage 2, C1 with private inputs $\Lambda=(E_{pk}(c_1), . . . , E_{pk}(c_w))$ and $\Lambda' = (E_{pk}(c_1'), . . . , E_{pk}(c_k'))$, and C2 with sk securely compute $E_{pk}(c_q)$. Here c_q denotes the majority class label among $c_1', . . . , c_{k'}$. At the end of stage 2, only Bob knows the class label c_q. The overall steps involved in Stage 2 are shown in Algorithm. To start with, C1 and C2 jointly compute the encrypted frequencies of each class label using the k-nearest set as input. That is, they compute $E_{pk}(f(c_i))$ using(Λ,Λ') as C1's input to the secure frequency (SF) protocol, for $1 \le i \le w$. The output $(E_{pk}(f(c_1)), . . . , E_{pk}(f(c_w)))$ is known only to C1. Then, C1 with $E_{pk}(f(c_i))$ andC2 with sk involve in the secure bit decomposition (SBD) protocol to compute $[f(c_i)]$, that is, vector of encryptions of the individual bits of $f(c_i)$, for $1 \le I \le w$. After this, C1 and C2 jointly involve in the SMAXw protocol. Briefly, SMAXw utilizes the sub-routine SMAX to eventually compute $([f_{max}], E_{pk}(c_q))$ in an iterative fashion. Here $[f_{max}] =[max(f(c1), . . . , f(c_w))]$ and cq denotes the majority class out of Λ′. At the end, the output $([f_{max}], E_{pk}(c_q))$ is known only to C1. After this,C1 computes $q = E_{pk}(c_q + r_q)$,where r_q is a random number in Z_N known only to C1. Then, C1 sends q to C2 and r_q to Bob. Upon receiving q, C2 decrypts it to get the randomized majority class label ′q = $D_{sk}(q)$ and sends it to Bob. Finally, upon receiving r_q from C1 and ′q from C2, Bob computes the output class label corresponding to q as $c_q = q - r_q \bmod N$.

3.2. Security Analysis of PPkNN under the Semi-honest Model:

Here we provide a formal security proof for the proposed PPkNN protocol under the semi-honest model. Due to the encryption of q and by semantic security of the Paillier cryptosystem, Bob's input query q is protected from Alice, C1 and C2 in our PPkNN protocol.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 2, Issue 2, pp: (114-120), Month: October 2015 – March 2016, Available at: www.paperpublications.org

Algorithm:

SCMCk ($E_{pk}(c_1'), \ldots, E_{pk}(c_k')) \rightarrow c_q$

Require: ($E_{pk}(c1), \ldots, E_{ok}(c_w)$), ($E_{pk}(c_1'), \ldots, E_{pk}(c_k')$)are known only to C1; sk is known only to C2

1: C1 and C2:

(a). ($E_{pk}(f(c_1)), \ldots, E(f(c_w))) \leftarrow SF(\Lambda,\Lambda')$, where $\Lambda = (E_{pk}(c_1), \ldots, E_{pk}(c_w))$, $\Lambda' = (E_{pk}(c_1'), \ldots, E_{pk}(c_k'))$

(b). **for** i = 1 to w do:

• $[f(c_i)] \leftarrow SBD(E_{pk}(f(c_i)))$

(c). $([fmax], E_{pk}(c_q)) \leftarrow SMAXw(\Psi_1, \ldots, \Psi_w)$, where $\Psi_i = ([f(ci)], E_{pk}(c_i))$, for $1 \leq i \leq w$

2: C1:

(a). $\gamma_q \leftarrow E_{pk}(c_q) * E_{pk}(r_q)$, where $r_q \in R\ Z_N$

(b). Send γ_q to C2 and r_q to Bob

3: C2:

(a). Receive γ_q from C1

(b). $\gamma'_q \leftarrow D_{sk}(\gamma_q)$; send γ'_q to Bob

4: Bob:

(a). Receive r_q from C1 and γ'_q from C2

(b). $c_q \leftarrow \gamma'_q - r_q \bmod N$

The goal of PPkNN is to protect data confidentiality and hide data access patterns. In this paper, to prove a protocol's security under the semi-honest model, we adopted the well-known security definitions from the literature of secure multiparty computation (SMC).

IV. SYSTEM OVERVIEW

Read the car evaluation dataset from KDD. It consists of 1,728 records and six attributes. There is a separate class attribute and the dataset is categorized into four different classes. Encrypt the dataset attribute-wise, using the Bitwise XOR operation. The Bitwise XOR operation split the string data by removing comas. Strings are converted into binary format using ASCII keyword. Convert the binary values into encrypted format by using Paillier cryptosystem. The Paillier cryptosystem is a public-key encryption scheme. It consists of 3 schemes:

- Key generation
- Encryption
- Decryption

The encrypted data send to cloud. Then the cloud will search the similar data. That data's are grouped together, and then collect the features of neighboring data by using SRKNN. From that data, find the majority class by means of SCMC and produce the classify data. Then analyze the efficiency of our solution through various experiments.

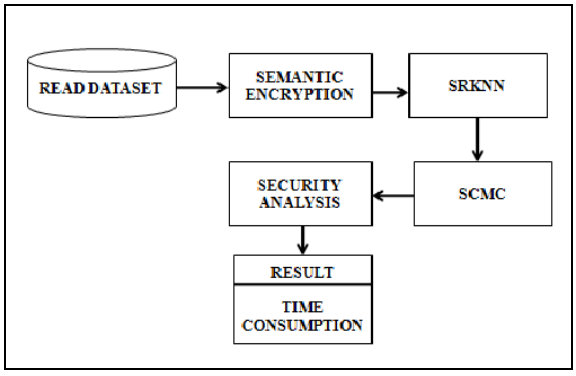


Fig: Architecture Design

V. COMPLEXITY ANALYSIS

The computation complexity of Stage 1 in PPkNN is bounded by O (n) instantiations of SBD and SSED, O (k) instantiations of SMINn, and O(n * k * l) instantiations of SBOR. We emphasize that the computation complexity of the SBD protocol proposed in is bounded by O (l) encryptions and O (l) exponentiations (under the assumption that encryption and decryption operations based on Paillier cryptosystem take similar amount of time). Also, the computation complexity of SSED is bounded by O (m) encryptions and O (m) exponentiations. In addition, the computation complexity of SMINn is bounded by O (l * n * log2 n) encryptions and O (l * n * log2 n) exponentiations. Here the computation complexity of SF is bounded by O (k * w) encryptions and O (k * w) exponentiations. Therefore, the total computation complexity of Stage 2 is bounded by O (w* (l + k +l * log2 w)) encryptions and exponentiations. In general, $w \ll n$, therefore, the computation cost of Stage 1 should be significantly higher than that of Stage 2.

VI. EMPIRICAL RESULTS

In this section, we discuss some experiments demonstrating the performance of our PPkNN protocol under different parameter settings. We used the Paillier cryptosystem as the underlying additive homomorphic encryption scheme and implemented the proposed PPkNN protocol in C. To the best of our knowledge, our work is the first effort to develop a secure k-NN classifier under the semi honest model. Thus, there is no existing work to compare with our approach. Therefore, we evaluate the performance of our PPkNN protocol under different parameter settings.

6.1. Dataset and Experimental Setup:

For our experiments, used the Car Evaluation dataset from the UCI KDD archive. The dataset consists of 1728 data records (i.e., n = 1728) with 6 input attributes (i.e., m = 6). Also, there is a separate class attribute and the dataset is categorized into four different classes (i.e., w = 4). We encrypted this dataset attribute-wise, using the Paillier encryption whose key size is varied in our experiments, and the encrypted data were stored on our machine. Based on our PPkNN protocol, we then executed a random query over this encrypted data.

6.2. Performance of PPkNN:

We first evaluated the computation costs of Stage 1 in PPkNN for varying number of k-nearest neighbors. Also, the Paillier encryption key size K is either 512 or 1024 bits. The results are shown in Figure 2(a). For K=512 bits, the Computation cost of Stage 1 varies from 9.98 to 46.16 minutes when k is changed from 5 to 25, respectively. On the other hand, when K=1024 bits, the computation cost of Stage 1 varies from 66.97 to 309.98 minutes when k is changed from 5 to 25, respectively. In either case, we observed that the cost of Stage 1 grows almost linearly with k. For any given k, we identified that the cost of Stage 1 increases by almost a factor of 7 whenever K is doubled. For example, when k=10, Stage 1 took 19.06 and 127.72 minutes to generate the encrypted class labels of the 10 nearest neighbors under K=512 and 1024 bits, respectively. Furthermore, when k=5, we observe that around 66.29% of cost in Stage 1 is accounted due to SMINn which is initiated k times in PPkNN

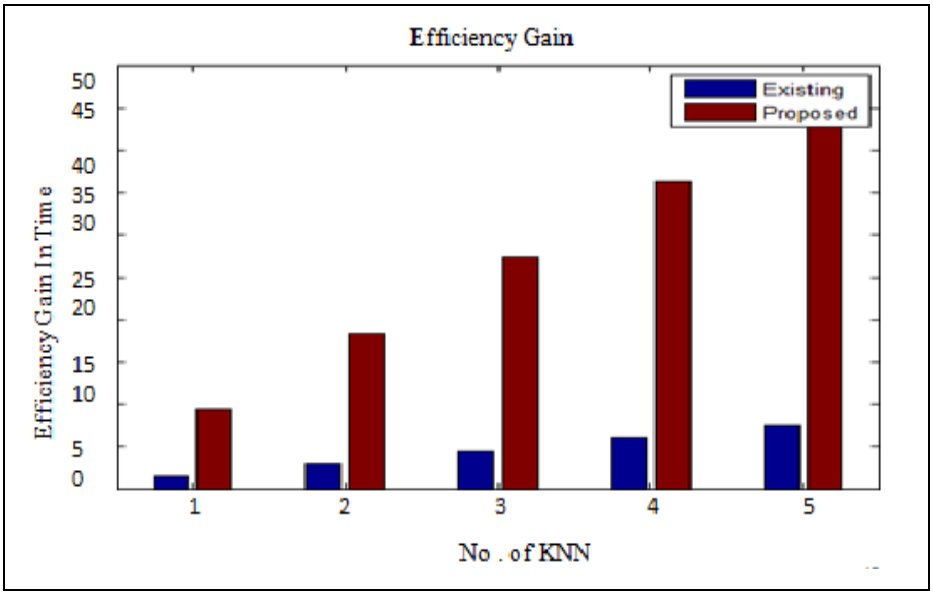
6.3. Performance Improvement of PPkNN:

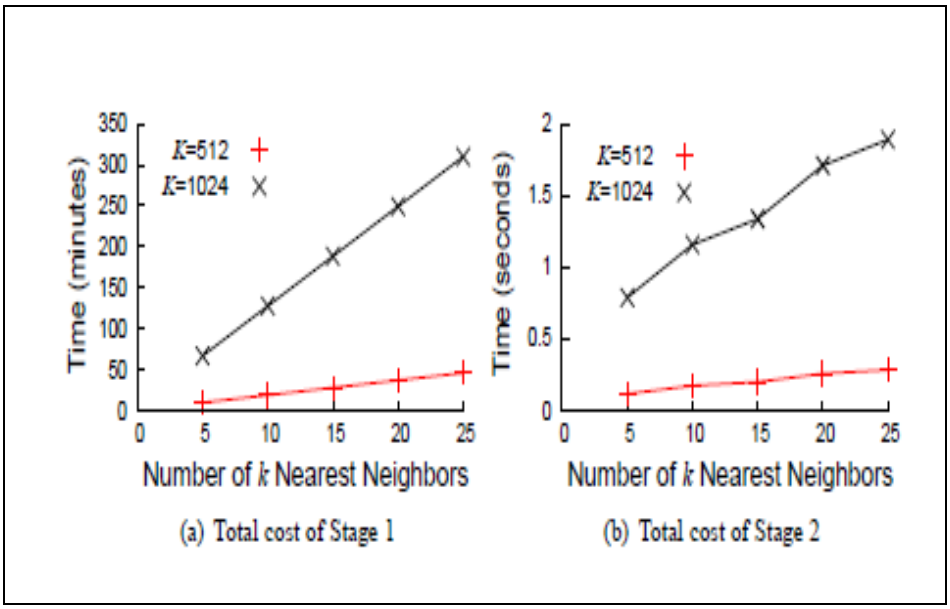
Two different ways to boost the efficiency of Stage 1 (as the performance of PPkNN depends primarily on Stage 1) and empirically analyze their efficiency gains. First, we observe that some of the computations in Stage 1 can be pre-computed. For example, encryptions of random numbers, 0s and 1s can be pre-computed (by the corresponding parties) in the offline phase. As a result, the online computation cost of Stage 1 (denoted by SRkNNo) is expected to be improved .To see the actual efficiency gains of such a strategy, we computed the costs of SRkNNo and compared them with the costs of Stage 1 without an offline phase (simply denoted by SRkNN) and the results for K = 1024 bits . Irrespective of the values of k, we observed that SRkNNo is around 33% faster than SRkNN. E.g., when k = 10, the computation costs of SRkNNo and SRkNN are 84.47 and 127.72 minutes, respectively (boosting the online running time of Stage 1 by 33.86%).PPkNN for k = 10 and K = 1024 bits Our second approach to improve the performance of Stage 1 is by using parallelism. Since operations on data records are independent of one another, we claim that most computations in Stage 1 can be parallelized. The efficiency of Stage 1 can indeed be improved significantly using parallelism. Moreover, we can also use the existing Map-reduce techniques to execute parallel operations on multiple nodes to drastically improve the performance further. Hence, the level of achievable performance in PPkNN actually depends on the implementation. On

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 2, Issue 2, pp: (114-120), Month: October 2015 – March 2016, Available at: www.paperpublications.org

the other hand, Bob's computation cost in PPkNN is mainly due to the encryption of his input query. In our dataset, Bob's computation cost is 4 and 17 milliseconds when K is 512 and 1024 bits, respectively.





VII. CONCLUSION

Classification is an important task in many data mining applications. To protect user privacy, various privacy-preserving classification techniques have been proposed in the literature for the past decade. The existing techniques are not applicable in outsourced database environment where the data resides in encrypted form on a third-party server. Proposed a novel privacy-preserving k-NN classification protocol over encrypted data in the cloud. Our protocol protects the confidentiality of the data, user's input query, and hides the data access patterns. We also evaluated the performance of our protocol under different parameter settings. Since improving the efficiency of SMINn is an important first step for improving the performance of our PPkNN protocol. We plan to investigate alternative and more efficient solutions to the SMINn problem in our future work. Also, in this paper, we used the well-known k-NN classifier and developed a privacy-preserving protocol for it over encrypted data. As a future work, we will investigate and extend our research to other classification algorithms.

REFERENCES

[1] C. C. Agrawal and P. S. Yu. A general survey of privacy-preserving data mining models and algorithms. Privacy-preserving data mining, pages 11–52, 2008.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In ACM SIGMOD, pages 563–574, 2004.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In ACM Sigmod Record, volume 29, pages 439–450.ACM, 2000.

[4] H. Hu, J. Xu, C. Ren, and B. Choi. Processing private queries over untrusted data cloud through privacy homomorphism. In IEEE ICDE, pages 601–612, 2011.

[5] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In IEE ICDE, pages 217–228, 2005.

[6] X. Xiao, F. Li, and B. Yao, "Secure nearest neighbor revisited," in Proc. IEEE Int. Conf. Data Eng., 2013, pp. 733–744.

[7] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in Proc. IEEE 30th Int. Conf. Data Eng., 2014, pp. 664–675.

[8] Y. Qi and M. J. Atallah, "Efficient privacy-preserving k-nearest neighbor search," in Proc. IEEE 28th Int. Conf. Distrib. Comput.Syst. 2008, pp. 311–319.

[9] M. Bohanec and B. Zupan. The UCI KDD Archive. University of California, Department of Information and Computer Science, Irvine, CA, 1997.

[10] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in Proc. 20th USENIX Conf. Security, 2011, pp. 35–35.