Third Party Data Clustering over Encrypted Data Without Data Owner Participation: Introducing The Encrypted Distance Matrix

Nawal Almutairi[1,2], Frans Coenen[1], and Keith Dures[1]

[1] Department of Computer Science, The University of Liverpool, Liverpool, UK
[2] Department of Information Technology, King Saud University, Riyadh, KSA
{n.m.almutairi,coenen,dures}@liverpool.ac.uk

Abstract. The increasing demand for Data Mining as a Service, using cloud storage, has raised data security concerns. Standard data encryption schemes are unsuitable because they do not support the mathematical operations that data mining requires. Homomorphic and Order Preserving Encryption provide a potential solution. Existing work, directed at data clustering, has demonstrated that using such schemes provides for secure data mining. However, to date, all proposed approaches have entailed some degree of data owner participation, in many cases the amount of participation is substantial. This paper proposes an approach to secure data clustering that does not require any data owner participation (once the data has been encrypted). The approach operates using the idea of an Encrypted Distance Matrix (EDM) which, for illustrative purposes, has been embedded in an approach to secure third-party data clustering - the Secure Nearest Neighbour Clustering (SNNC) approach, that uses order preserving and homomorphic encryption. Both the EDM concept and the SNNC approach are fully described.

Keywords: Privacy preserving data mining, Secure nearest neighbour clustering, Order preserving encryption, Homomorphic encryption.

1 Introduction

Data Mining as a Service (DMaaS), using cloud storage, provides data owners with a set of useful tools for data analytics. Although cloud services provide a reliable infrastructure to host data and the potential for third party analytics, usage of such services entails issues of data confidentiality and security, and unauthorised data accesses (data leakage). Consequently, Privacy Preserving Data Mining (PPDM) approaches have been proposed to address these issues [1]. The typical PPDM approach is to provide the third party data miner with a version of the data where sensitive data attributes have been either removed or modified using data transformation methods, such as data obfuscation, perturbation and anonymization. These methods tend to operate by introducing "statistical noise" to the sensitive attribute-values. This can then compromise the effectiveness of the data analysis; whilst, at the same time, it might still be possible to "reverse engineer" the original data values. Hence, data confidentiality cannot be guaranteed.

Data encryption can substantially guarantee data privacy and significantly mitigate against the risk of unauthorised data accesses. However, data encryption, in its standard form, prevents the application of any form of data mining, rendering it unsuited in the context of DMaaS. Data mining activities require data manipulation and data comparison of some form. A potential solution is Homomorphic Encryption (HE) [7], a form of encryption that supports a limited number of mathematical operations. HE schemes have been proposed that support addition, subtraction, multiplication and division of various forms. However, although the operations provided by HE schemes go some way to support DMaaS, they do not provide an entire solution; for example they do not support logical operations (over cypher-text) that data mining algorithms frequently require. One proposed solution [4, 8] is to incorporate periodic recourse to data owners during the data mining process, so that the data owners can perform the data operations (on unencrypted data) that the adopted HE scheme does not support. For example, in the context of data clustering, the comparison of records. However, using this approach the amount of data owner participation is significant, calling in to question the advantages that DMaaS has to offer; although the approach does provide a suitable mechanism for collaborative secure data mining [11] and therefore does have merit. There has been some work that seeks to reduce the amount of data owner participation, of note is the idea of a 3-D Updateable Distance Matrix (UDM) presented in [2], but this still does not resolve the data owner participation issue. An alternative solution, in the context of collaborative data mining, is "secret sharing" [14], which aims to minimise data owner participation by introducing semi-honest and non-colluding third parties that decrypt, perform operations on "data pieces" (called shares) that hold no comprehensive information, and then re-encrypt, on behalf of the data owner. The global results can be reconstructed by knowing the individual results from several parties. Therefore, this again does not guarantee data confidentiality, whilst issues with data leakage remain.

From the foregoing, research directed at secure DMaaS has been predominantly focused on involving data owners, or constructing complex models to share secret keys, so as to resolve the current security issue associated with DMaaS. As noted above, these have significant limitations. Ideally, data owners should be able to package their data so that it is secure, send it to a third party for storage and analysis, and receive analysis results as and when required, without the need for any further communication whilst the analysis is taking place. In this paper a mechanism is proposed whereby secure third-party data mining (DMaaS) can be provided that does not entail any of the disadvantages of existing mechanisms. The fundamental idea, influenced in part by the UDM concept presented in [2], is to use a 2-D Encrypted Distance Matrix (EDM). The idea is illustrated in the context of Nearest Neighbour Clustering [3], we refer to this as the Secure Nearest Neighbour Clustering (SNNC) approach; however, the EDM idea clearly has wider application.

2 Previous Work

The main challenge of HE in the context of DMaaS in general, and data clustering in particular, is that HE schemes support only a limited number of arithmetic operations. As noted above, several theoretical and practical solutions to address this challenge have been proposed, these can be broadly categorised as either: (i) recourse to data owner when unsupported operations are required or (ii) utilising secret sharing techniques. Both have limitations in terms of communication complexity and security.

The key feature of the first category of solution is the realisation of data confidentiality by only permitting third party access to the HE data (without knowledge of the keys that have been used for the encryption). This means that data owner participation is required with respect to unsupported operations. The degree of participation depends on the nature of the mining to be undertaken. The worst case is when using what is known as Secure Multi-Party Computation (SMPC) [4, 11, 15], where the majority of data processing is conducted in-house by data owners. In the context of SMPC and k-Means clustering, as described in [11, 15], the third party data miner acts as a mediator and, on each iteration, calculates global cluster centroids; similarity measurement, assigning records to clusters and calculating local centroids are delegated to data owners. Data owners therefore do much of the work. In [4] k-Means clustering is also considered, but in this case, using an appropriate HE scheme; the third party data miner calculates distances between records and cluster centroids, and global centroids, whilst delegating similarity determination to data owners.

There has also been work directed at reducing the data owner's participation where the data owner provides static and dynamic *trapdoor* values to guide the third party data miner when comparing cypher-texts. One example is presented in [8] where K-Means clustering is used to illustrate the approach. However, data owner participation is still a significant requirement because the dynamic trapdoors need to be recalculated on each iteration of the K-Means clustering. The UDM concept proposed in [2] has the lowest data owner participation, illustrated in the context of k-Means clustering, where only very limited data owner participation is required on each iteration. However, the UDM (unlike the proposed EDM) is unencrypted; given that a UDM is essentially a set of linear equations this still presents a security threat. The idea of using user generated matrices holding data to support DMaaS has featured in other contexts. For example in [17] the data owner provides two matrices, of size $(2|A| + 2)$ and $(2|A|+2)\times|A|$ (where $|A|$ is number of attributes), the two matrices are computed using a private matrix; the process is directed at supporting data classification. However, data owner participation is still mandatory. In [16], an improvement of the scheme given in [17] was introduced that avoided data owner participation, however to do this part of the secret key is disclosed which in turn threatens data privacy.

The second category encompasses more recent work and features usage of some form of *secret sharing* scheme where at least two semi-honest, non-colluding, data miners perform computations by collaboratively decrypting private data on behalf of data owners, neither has access to the entire data set in unencrypted form. For example in [12, 14] a secret key is generated, by the collaborating parties, using a Threshold Paillier encryption scheme [5]. Secure computation protocols are used to allow the two parties to execute operations without requiring data owner participation. However, the collaborative nature of the computational protocols used induce communication overheads that make secret sharing very inefficient and thus not practical for large data sets. The requirement for at least two semi-honest and not-colluding data miners is also of concern.

The work presented in this paper does not fit neatly in either category, it does not require data owner participation and does not require secret sharing. In this context the proposed EDM mechanism is unique.

3 Preliminaries

Before considering the Encrypted Distance Matrix (EDM) concept and the Secure Nearest Neighbour Clustering (SNNC) approach in detail, the utilised encryption schemes are presented in this section.

3.1 Homomorphic Encryption: Liu's Scheme

Using the proposed SNNC approach, the raw data to be outsourced is encrypted using Liu's homomorphic encryption scheme as defined in [7]. The homomorphic properties of the scheme support the addition and subtraction of cypher-text, and the multiplication and division of cypher-text with real numbers. Although the proposed SNNC does not specifically utilise the homomorphic properties of Liu's scheme, the proposed solution is directed at providing a generic solution suited to many forms of secure data mining.

3.2 Order Preserving Encryption

A Distance Matrix holds the distances (differences) between each record in D with every other record in D. Using SNNC these distances are encrypted using an Order-Preserving Encryption (OPE) to give an Encrypted DM (EDM). Thus, the generated EDM holds the order of distance instead of real distance values.

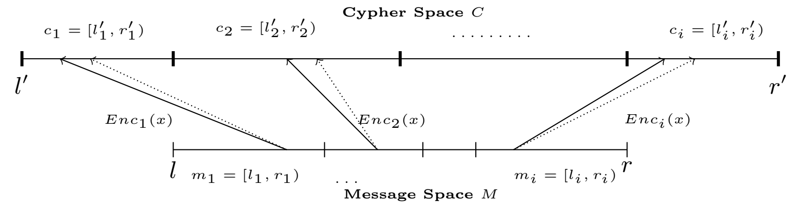


Fig. 1. Message and expanded cypher space splitting

The proposed OPE scheme is an amalgamation of two existing OPE schemes; [9] and [10]. The key feature of the OPE is to obscure any data distribution that

Algorithm 1 Order Preserving Encryption algorithm

```
1: procedure ENC($x, Sens$)
2:     $i$ = Interval($x$)
3:     $[l_i, r_i] \leftarrow$ Range($i$)
4:     $[l'_i, r'_i] \leftarrow$ Range$'$($i$)
5:     $Scale_i = \frac{(l'_i - r'_i)}{(l_i - r_i)}$
6:     $\delta_i$ = Random($0, Sens \times Scale_i$)
7:     $x' = l'_i + Scale_i \times (x - l_i) + \delta_i$
8:     Exit with $x'$
```

might be included in the generated cypher-texts using the concept of "message space splitting" and "non-linear cypher space expansion". The first step is to determine the "interval" of the message space $M = [l, r)$ and the "interval" of the cypher space $C = [l', r')$, where r is the maximum interval boundary and l is the minimum interval boundary, in such a way that $|C| \gg |M|$, as shown in Figure 1. The next step is to randomly split the message space into successive intervals, as also shown in Figure 1. The cypher space C is then split into the same number of intervals. However, the length of the cypher space intervals is determined by the density of the data in the corresponding message space intervals in such a way that message space intervals that have high data densities have large corresponding cypher space intervals. A "one-to-many" encryption function is then adopted. With the respect to the work presented in this paper the adopted function is shown in Algorithm 1. The encryption function commences by retrieving the interval ID number of value x by calling the *interval* function (line 2). The maximum and minimum interval boundary for the message space interval, holding x, and the corresponding cypher space interval are retrieved in lines 3 and 4. These values are used in lines 5 to 7 to generate the cypher x'. The δ_i variable in line 6 is a random number mapped from $[0, Sens \times Scale)$ where *Sens* is the minimum distance between the plain-text values in the data, as presented in [9], and scale value as calculated in line 5. This value guarantees that different cypher-texts, for the same plain-text value, will be generated on different occasions thus obscuring the data frequency.

4 Encrypted Distance Matrix (EDM) Generation

Regardless of whether standard or HE encryption is used, the encryption randomly translates the plain-text values in a given data set D to cypher-texts in such a way that any value ordering that existed in the original plain-text values is not preserved. Therefore, comparison operations cannot be directly applied and thus even the most trivial forms of data analysis cannot be performed. The idea presented in this paper is thus to use an Encrypted Distance Matrix (EDM), that holds encrypted distances between data values, so that comparisons can be conducted. An EDM is a 2D matrix where the first and second dimensions represent the records in D. The matrix is symmetric about the leading diagonal, thus only the leading triangle needs to be considered. EDM generation is done in two steps: (i) distance calculation and (ii) encryption. Given a data set $D = \{r_1, r_2, \ldots, r_n\}$, where each record r_x is a feature vector comprised of a set of values $\{v_{x_1}, v_{x_2}, \ldots, v_{x_a}\}$, a distance matrix, $DM(x, y)$, is calculated using:

$$DM(x,y) = \sum_{i=1}^{i=a} (v_{x_i} \sim v_{y_i}) \tag{1}$$

A DM calculated in this manner, as in the case of the UDM proposed in [2], essentially comprises a set of linear equations which might present a security threat. Therefore, the second step is to encrypt the data, but so that ordering is preserved. To this end the OPE scheme given in Sub-section 3.2 above was used.

5 Secure Nearest Neighbour Clustering

This section presents the proposed Secure Nearest Neighbour Clustering (SNNC) approach designed to operate over encrypted data and without any further user participation once the data has been outsourced. The clustering process, like the EDM generation process, has two steps: (i) data preparation conducted by the data owner and (ii) consequent clustering conducted by the third party. The first is discussed in Sub-section 5.1 below, and the second in Sub-section 5.2.

5.1 Data Owner Data Preparation

During the initial data preparation step the data owner pre-processes the data to be outsourced by replacing the categorical (or labelled) data with discrete integers values before any further processing. The processed data is then used to generate the required EDM after which the data is encrypted to give D'. Once the data has been successfully outsourced no further data owner participation will be required (other than receiving the final clustering result).

5.2 Third Party Clustering: SNNC Algorithm

The SNNC is conducted by the third party data miner following a process similar to that used for standard NNC [3]. The pseudo code presented in Algorithm 2 summarises this process. The input is the encrypted data set D', the EDM (previously submitted to the third party) and the desired threshold σ'. The algorithm commences by assigning the first record r'_1 to the first cluster K_1 (lines 2 and 3). Next, the number of generated clusters so far is set to be 1 (line 4). A loop is then entered (lines 5 to 11) that iteratively clusters the remaining records in D'. A feature of the SNNC algorithm is that the threshold value σ is also encrypted, to give σ', using the proposed OPE scheme so that the third party data miner processes the order of distances not real distance values. The record r'_i will be assigned to a cluster if there exists a record r'_m whose order of distance from r'_i is less than or equal to σ' using the EDM concept (lines 6 to 8). If no such record is found, a new cluster is created for r'_i (lines 10 and 11). The algorithm will continue until all records in D' are assigned to clusters and exits with a cluster configuration K.

Algorithm 2 Secure Nearest Neighbour Clustering

```
procedure SecureNearestNeighbourClustering($D', EDM, \sigma'$)
    $K_1 = \{r'_1\}$
    $K = \{K_1\}$
    $k = 1$
    for $i = 2$ to $i = |D'|$ do
        Find the $r'_m$ in some cluster in $K$ where the $EDM[r'_i, r'_m]$ is the smallest
        if $EDM[r'_i, r'_m] \leq \sigma'$ then
            $K_m = K_m \cup r'_i$
        else
            $k++$
            $K_k = \{r'_i\}$
    Exit with $K$
```

6 Evaluation

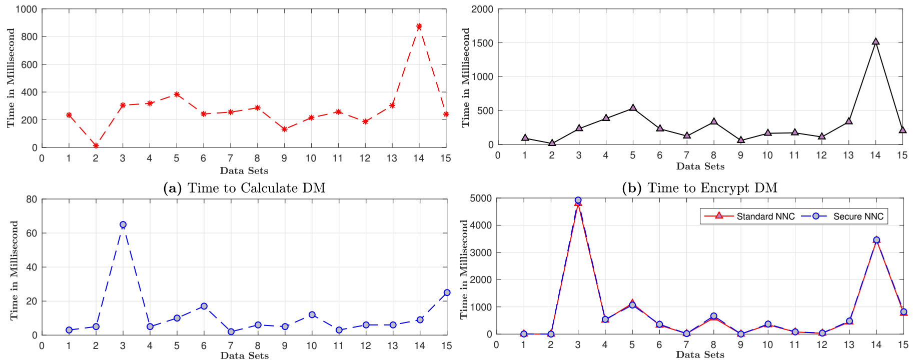
The evaluation of the proposed SNNC approach is presented in this section. Extensive experiments were conducted to evaluate both the SNNC approach and the EDM concept. Fifteen data sets from the UCI data repository were used [6], these were selected so that data sets of a variety of sizes and different numbers of classes could be considered. The data sets are listed in Table 1. The implementation was done using the Java programming language. The evaluation criteria considered were: (i) data owner participation, (ii) clustering efficiency, (iii) comparative clustering accuracy and (iv) security. Each is considered in further detail in Sub-sections 6.1 to 6.4.

6.1 Data Owner Data Preparation Run Time Complexity

Figure 2 (a to c) shows the runtime complexity recorded to encrypt the data, and calculate and encrypt the Distance Matrix (DM). The time to encrypt the data is correlated to the number of data records times the number of attributes in each data set. From the figure it can be seen that negligible time is required to encrypt the data, the recorded time to encrypt the largest data set (Arrhythmia) was 65ms. In the case of calculating and encrypting the DM to produce the desired EDM, the runtimes were longer compared to data encryption although inspection of the figure shows that it is not significantly so. The reported time to calculate and encrypt the EDM for (Banknote authent.) was the highest; 876ms to calculate the DM and 1509ms to encrypt it. Additional records can be added, as and when they arrive, without necessitating re-encryption of the existing data. Once encrypted, the data and EDM are sent to the third party, no further data owner participation is required.

6.2 Clustering Efficiency

A comparison of the runtime required to cluster the data using standard NNC and SNNC is presented in Figure 2 (d). From the figure it can be seen that



(a) Time to Calculate DM

(b) Time to Encrypt DM

(c) Time to Encrypt Datasets

(d) Execution Time for SNNC and NNC

Fig. 2. Runtimes for data owner data preparation and NNC/SNNC execution

Table 1. Cluster Configuration Comparison using Standard NNC and SNNC.

Data Set	R × C	Num. Labels	σ	Standard NNC Num. Cluster	Standard NNC Sil. Coef.	Secure NNC Num. Cluster	Secure NNC Sil. Coef.
1. Iris	150× 4	4	3.00	2	0.722	2	0.722
2. Lung cancer	32×56	3	0.10	32	1.000	32	1.000
3. Arrhythmia	452×279	16	1980.00	16	0.889	16	0.889
4. Blood transfusion	748×4	2	1046.00	4	0.895	4	0.895
5. Pima Ind. Diabetes	768×8	2	498.00	2	0.741	2	0.741
6. Chronic Kidney Dis.	400×24	2	952.00	16	0.981	16	0.981
7. Seeds	210×7	3	4.00	2	0.579	2	0.579
8. Brest Cancer	699×9	2	20.00	6	0.470	6	0.470
9. Breast Tissue	106×9	6	990.00	38	0.999	38	0.999
10. Dermatology	366×34	6	26.00	8	0.745	8	0.745
11. Ecoli	336×7	8	0.76	7	0.881	7	0.881
12. Parkinsons	195×22	2	91.00	8	0.930	8	0.930
13. Ind. Liver Patient	583×10	2	100.00	98	0.997	98	0.997
14. Banknote authent.	1372×4	2	11.00	16	0.752	16	0.752
15. Libras Movement	360×90	15	10.00	19	0.753	19	0.753

the difference in execution time is minimal. It can be concluded, at least in the context of NNC, that secure DMaaS using the proposed SNNC mechanism does not introduce a significant efficiency overhead (once the data has been encrypted).

6.3 Clustering Accuracy

In context of accuracy, cluster configuration "correctness" was measured by comparing the final clustering results obtained using standard NNC with those generated using the proposed SNNC approach; the SNNC approach should produce cluster configurations equivalent to those produced using standard NNC to prove that the proposed solution is operating correctly. This was measured by the Silhouette Coefficients (Sil. Coef.) [13]. Table 1 gives the results obtained in the context of both standard NNC and SNNC; columns 6 and 8. From the table it can be seen that the clustering configurations produced using SNNC were identical to those produced using standard NNC as evidenced by the Silhouette Coefficients obtained using the same σ threshold (shown in column 4).

6.4 Security Analysis

Security was evaluated in terms of the potential attacks that could be directed at the proposed secure clustering algorithm. The security of the proposed clustering approach relies on the security of: (i) Liu's scheme for encrypting the raw data and (ii) the OPE used for encrypting the DM. Liu's scheme is semantically secure as proven in [8], which means that adversaries cannot determine any information regarding the data from the cypher equivalents. In cryptography, when a scheme is said to be semantically secure this implies that the scheme is secure against Cypher-text Only Attacks (COAs). Therefore, with respect to the proposed secure clustering, adversaries that have access to the encrypted data set cannot readily threaten the system. In terms of the EDM, a COA could be used to extract statistical measures describing the frequency of distribution patterns which could be used to identify frequently occurring distributions which in turn could be used to identify the nature of plain-texts (if examples were available). However the nature of the OPE scheme is such that the distribution is obscured using the concept of message space splitting and non-linear cypher space expansion. The one-to-many encryption function produces different cypher-texts for the same plain-text values thus obscuring the data frequency, especially when the scale intervals are large. Recall that data owner participation is avoided, thus Chosen Cypher-text attacks or Chosen Plain-texts attacks cannot be instigated with respect to the proposed secure clustering algorithm.

7 Conclusion

In this paper a mechanism for DMaaS has been proposed founded on the idea of an Encrypted Data Matrix (EDM). The approach was illustrated using a clustering scenario, the Secure Nearest Neighbour Cluster (SNNC) approach. The proposed method utilised Order Preserving Encryption (OPE) and Homomorphic Encryption (HE) to maintain data confidentiality. Unlike other proposed solutions to third party data clustering, the proposed approach does not require any data owner participation once the data (and the EDM) have been sent to the third party. The reported evaluation clearly demonstrates that the encryption schemes do not adversely affect the quality of data clustering. These are the same as when standard NNC is applied to the same data. For future work, the authors intend to investigate the utility of the EDM concept with respect to alternative clustering and classification algorithms.

References

1. Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.
2. Nawal Almutairi, Frans Coenen, and Keith Dures. *K-Means Clustering Using Homomorphic Encryption and an Updatable Distance Matrix: Secure Third Party Data Clustering with Limited Data Owner Interaction*, pages 274–285. Springer International Publishing, Cham, 2017.
3. T.M. Cover and P.E. Hart. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
4. Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. Privacy-preserving user clustering in a social network. In *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 96–100. IEEE, 2009.
5. Carmit Hazay, Gert Læssøe Mikkelsen, Tal Rabin, and Tomas Toft. Efficient RSA key generation and Threshold Paillier in the two-party setting. In *CT-RSA*, pages 313–331. Springer, 2012.
6. M. Lichman. UCI machine learning repository, 2013.
7. Dongxi Liu. Homomorphic encryption for database querying, 12 2013.
8. Dongxi Liu, Elisa Bertino, and Xun Yi. Privacy of outsourced k-means clustering. In *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pages 123–134. ACM, 2014.
9. Dongxi Liu and Shenlu Wang. Nonlinear order preserving index for encrypted database query in service cloud environments. *Concurrency and Computation: Practice and Experience*, 25(13):1967–1984, 2013.
10. Zheli Liu, Xiaofeng Chen, Jun Yang, Chunfu Jia, and Ilsun You. New order preserving encryption model for outsourced databases in cloud environments. *Journal of Network and Computer Applications*, 59:198–207, 2016.
11. Deepti Mittal, Damandeep Kaur, and Ashish Aggarwal. Secure data mining in cloud using homomorphic encryption. In *Cloud Computing in Emerging Markets (CCEM), 2014 IEEE International Conference on*, pages 1–7. IEEE, 2014.
12. Fang-Yu Rao, Bharath K Samanthula, Elisa Bertino, Xun Yi, and Dongxi Liu. Privacy-preserving and outsourced multi-user k-means clustering. In *Collaboration and Internet Computing (CIC), 2015 IEEE Conference on*, pages 80–89. IEEE, 2015.
13. Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
14. Bharath K Samanthula, Yousef Elmehdwi, and Wei Jiang. K-nearest neighbor classification over semantically secure encrypted relational data. *IEEE transactions on Knowledge and data engineering*, 27(5):1261–1273, 2015.
15. Y. Shen, J. Han, and H. Shan. The research of privacy-preserving clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 324–327, 2010.
16. Youwen Zhu, Zhikuan Wang, and Yue Zhang. Secure k-nn query on encrypted cloud data with limited key-disclosure and offline data owner. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 401–414. Springer, 2016.
17. Youwen Zhu, Rui Xu, and Tsuyoshi Takagi. Secure k-NN query on encrypted cloud database without key-sharing. *International Journal of Electronic Security and Digital Forensics*, 5(3-4):201–217, 2013.