A Cryptographic Ensemble for Secure Third Party Data Analysis: Collaborative Data Clustering Without Data Owner Participation

Nawal Almutairi[a,b,*], Frans Coenen[a,**], Keith Dures[a]

[a]*Department of Computer Science, University of Liverpool, Liverpool, UK*
[b]*Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*

Abstract

This paper introduces the twin concepts Cryptographic Ensembles and Global Encrypted Distance Matrices (GEDMs), designed to provide a solution to outsourced secure collaborative data clustering. The cryptographic ensemble comprises: Homomorphic Encryption (HE) to preserve raw data privacy, while supporting data analytics; and Multi-User Order Preserving Encryption (MUOPE) to preserve the privacy of the GEDM. Clustering can therefore be conducted over encrypted datasets without requiring decryption or the involvement of data owners once encryption has taken place, all with no loss of accuracy. The GEDM concept is applicable to large scale collaborative data mining applications that feature horizontal data partitioning. In the paper DBSCAN clustering is adopted for illustrative and evaluation purposes. The results demonstrate that the proposed solution is both efficient and accurate while maintaining data privacy.

Keywords: Data mining as a service, Privacy preserving data mining, Security, Data outsourcing.

1. Introduction

The resources facilitated through cloud computing has provided a variety of services including Data Mining as a Service (DMaaS). The emergence of third party data mining facilitated by DMaaS has effectively liberated data owners from managing their data and in-house data analysis. Furthermore, it has opened the door to collaborative data mining where a number of data owners pool their data (for analysis) so as to gain some mutual advantage. However, significant concerns related to data privacy and security have served to limit the adoption of DMaaS. The research domain of Privacy Preserving Data Mining (PPDM) came into being to address DMaaS privacy concerns [1, 2].

[*] Corresponding author
[**] Principal corresponding author
Email addresses: n.m.almutairi@liverpool.ac.uk, nawalmutairi@ksu.edu.sa (Nawal Almutairi), coenen@liverpool.ac.uk (Frans Coenen), dures@liverpool.ac.uk (Keith Dures)

Early PPDM solutions were directed at Secure Multi-Party Computation (SMPC) [3]. The fundamental idea was for individual data owners to locally process their data (in plaintext) to produce some statistical features that describe their private data. These statistics were then used as inputs to secure computation protocols that securely calculate global characteristics which were then used to develop a global data mining model. Despite the many possible SMPC applications, the significant computation and communication overhead associated with many SMPC protocols makes SMPC-based solutions infeasible for large scale collaborative data mining. Moreover, the requirement for data owner involvement has posed a security risk as it provides the potential for non-honest parties to launch "overlapping attacks" [4, 5].

The alternative solution to SMPC is to outsource the data analysis to a DMaaS third party. In this case, data privacy is maintained either by transforming or encrypting the data in such a way that the desired data analysis can still be conducted. The adopted transformation method can be applied either to selective sensitive data attributes, as in the case of data anonymisation [6], or to the entire dataset, as the case of data perturbations [7, 8]. In data anonymisation, the confidential attributes are removed; and then the produced dataset is generalised until some "syntactic" condition is achieved. Data perturbation operates by distorting or randomising the entire dataset by adding noise while maintaining the statistical makeup of the data. A criticism of the first is that data cannot be 100% anonymised [9], in many cases public attributes in anonymised data, dubbed *quasi-identifiers*, can be exploited through "linkage attacks" [10]. A criticism of data perturbation methods is that they cannot entirely assure data privacy since most of the methods used allow "reverse engineering" of the original data distribution [11]. Regardless of the technique applied, the nature of the proposed transformations has also been shown to adversely affect the quality of the data analysis [8, 12].

Data privacy can be substantially guaranteed using data encryption. The emergence of Homomorphic Encryption (HE) schemes, that support limited mathematical operations over cyphertexts (without decryption) [13, 14], goes some way to supporting DMaaS. However, HE schemes do not provide an entire solution; for example they do not support data comparison over cyphertexts. Proposed solutions either resort to an SMPC style protocol for data comparison [15, 16] or require recourse to data owners to conduct the required comparisons. Consequently, these solutions feature the same communication and computation overheads as in the case of the more general SMPC approach. There has been some work directed at reducing data owner participation. One example is the 2-D Encrypted Distance Matrix (EDM) proposed in [17] and the Super Secure Chain Distance Matrix (SSCDM) proposed in [4]. However, the usage of EDMs is limited to applications where a single data owner outsources data mining activity to a single third party data miner. SSCDMs entail inefficiencies, especially when applied to large dataset, because similarity is determined using a "chain feature" that increase in size with the number of data records in the dataset.

Given the above, this paper proposes a Cryptographic Ensemble approach to PPDM and introduces the concept of Global Encrypted Distance Matrices (GEDMs) that provide a solution to secure third party collaborative data clustering without involving data owner participation and without loss of accuracy. The Cryptographic Ensemble comprises Lui's HE scheme [13], to preserve data privacy, and Multi-Users Order Preserving Encryption (MUOPE) [4] to facilitate secure data comparison using a GEDM. The fundamental idea, influenced by the EDM concept presented in [17], is to use 2D Encrypted

Distance Matrices (EDMs) generated by individual data owners. The GEDM is then the union of two or more EDMs that is securely generated with limited data owner participation, using a proposed “pooling method” and the Cryptographic Ensemble scheme, with the support of a Semi-honest Third Party (STP). The GEDM, once generated, offers a secure and accurate mechanism to determine similarity between records distributed across data owners without involving data owner participation, and without resorting to SMPC protocols. The Cryptographic Ensemble and GEDM solution are fully described and evaluated in the paper. The evaluation is conducted in the context of DBSCAN, however, the proposed approach clearly has much wider application.

The rest of the paper is organised as follows. Section 2 presents a review of related work. Section 3 provides the fundamental background concerning the proposed Cryptographic Ensemble. The GEDM idea is then detailed in Section 4. Section 5 presents the proposed Secure DBSCAN (S-DBSCAN) algorithm, while Section 6 reports on the evaluation of the GEDM concept and S-DBSCAN. Finally, Section 7 concludes the paper with a summary of the main findings.

2. Related Work

Previous work directed at secure collaborative data clustering can be broadly categorised as being founded on either (i) SMPC or (ii) secure outsourcing to a third party. These solutions are considered in further detail below. The first category of solution makes use of secure computation protocols to provide secure solutions to the computation of any function [3]. In the context of collaborative data clustering these protocols have been adopted to allow participating parties to jointly compute functions concerning their data without revealing the data to other participants. The nature of the functions and protocols used depend on the nature of the clustering algorithm to be adopted. There have been a number of SMPC implementations for DBSCAN [5, 18, 19] directed at different numbers of participants (two-party or multiple-party) and various data partitionings. The required functions to be calculated with respect to DBSCAN are: (i) distances between records split across multiple data owners, typically calculated using scalar product protocols; and (ii) comparison of the calculated distances against threshold values, typically achieved using the well-established “Yao Millionaires Problem” [16] or Cachin’s scheme [15]. Whatever the case, SMPC has three major limitations: (i) SMPC protocols feature a significant computation and communication overhead and thus proposed solutions suffer from lack of scalability, (ii) they are inefficient due to the requirement of synchronising process across multiple parties and (iii) in terms of security, and in the specific context of DBSCAN, the nature of SMPC and the need for data owner involvement gives rise to the potential of “overlapping attacks” [4, 5].

Outsourcing of the data clustering to a third party data miner, using DMaaS, assumes that the third party data miner cannot be fully trusted, hence data has to be transformed or encrypted locally by individual data owners. The most straightforward transformation method is data anonymisation, as introduced in [6] and then further developed by considering the diverse levels of background knowledge of adversaries. However, the extensive experimentation reported in [20] demonstrated that the way the data is anonymised is very dependent on the nature of the data and the purpose for which it is to be used, thus limiting the type of data analytics that may be applied. The same study demonstrated that the “sanitisation feature” of anonymisation adversely effects the data utility and hence the clustering accuracy. More importantly, anonymised data can be "de-anonymised" using "linkage attacks" [10]. Data perturbation tends to operate by introducing "statistical noise". In the literature a number of different two-party and multi-party perturbation methods have been presented. There are a number of data mining algorithms that have been implemented using perturbation methods and these are surveyed in [21]. However, in many cases, perturbation has a security-accuracy tradeoff, since the higher the level of security provided by the perturbation method the worse the accuracy of the data mining. Moreover, the requirement for adopting the same perturbation method across all parties when conducting collaborative data mining makes the solution vulnerable to security breaches. Perturbation also provides potential for reconstructing the original data distribution.

Data encryption serves to preserve data confidentiality in the context of the outsourcing scenario, as in [4, 17, 22, 23]. In [22] datasets are encrypted using a threshold HE scheme and outsourced to a third party data miner who then utilises the HE properties to calculate distance between records. However, the calculated distance cyphertexts cannot be compared, hence data owners participation is required to compare values by decrypting the cyphertexts and comparing the plaintext values with thresholds. Data owner participation can be reduced through "secret sharing". The basic idea is to use a scheme that mathematically splits a secret key among multiple semi-honest and non-colluding parties that collaboratively manipulate data on behalf of data owners [23]. However, this approach tends to be inefficient for large datasets. Moreover, the requirement for a number of semi-honest and non-colluding parties is a security concern. The ideal solution is that recently introduced in [17, 4]. In [17] the concept of Encrypted Distance Matrices (EDMs) was used, however, this proposed solution is not applicable in the collaborative data mining context. The solution presented in [4] considered collaborative data clustering using the concept of Super Secure Chain Distance Matrices (SSCDMs). However, the way the third party derived similarities between data records required interaction with the SSCDM using a "chain feature". The larger the dataset the longer the SSCDM and thus the greater the time required to determine data similarity. More importantly, the reported results in [4] demonstrated that the SSCDM does not always match the data mining results produced using standard algorithms and unencrypted data (see also [24]).

3. The Cryptographic Ensemble

Before considering the proposed GEDM concept and Secure DBSCAN (S-DBSCAN) in detail, the Cryptographic Ensemble idea is presented here. The proposed approach uses two different encryption schemes: (i) Liu's HE scheme and (ii) the MUOPE scheme. Liu's HE scheme [13] is used to encrypt the raw data belonging to data owners. Therefore, the individual data owner is responsible for generating their own Liu's encryption key to encrypt their raw data (locally) before outsourcing their data. Liu's scheme supports addition, subtraction, multiplication of cyphertexts and multiplication and division of cyphertexts by real numbers. Although the proposed S-DBSCAN does not specifically utilise the HE properties of Liu's scheme, the proposed solution aims to provide a generic solution suited to many forms of secure data mining.

The MUOPE scheme, first introduced in [4], makes use of a Semi-honest Third Party (STP) that acts as a mediator between u data owners. In this paper, the STP's role is to: (i) derive the MUOPE encryption key and (ii) manage the pooling method used to

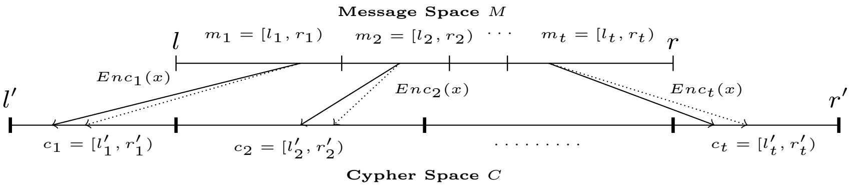

Figure 1: Message and expanded cypher space splitting

generate a GEDM. Two concepts are utilised to hide data distributions, "message space splitting" and "non-linear cypher space expansion", whilst a "one-to-many" encryption function is used to hide data frequency. The first step is for the STP to determine the message space "interval" $M = [l, r)$ and corresponding expanded cypher space "interval" $C = [l', r')$, where r represents the maximum interval boundary and l represents the minimum interval boundary. The intervals should be selected in such a way that $|C| \gg |M|$. The next step is to randomly split the message space M into t consecutive intervals, $M = \{m_1, m_2, \ldots, m_t\}$, where t is selected randomly by the STP. Each interval i has a minimum and maximum interval boundary, l_i and r_i (see Figure 1). The message space interval lengths are determined randomly by the STP. The cypher space C is then split into t intervals, however, to hide the data distribution the length for each cypher space interval c_i is determined using the "data density" of the corresponding message space m_i in such a way that message space intervals that have high data densities have large corresponding cypher space intervals. To accumulate the data density, in each message space interval, for data distributed across multiple data owners, the Paillier HE scheme [14] is utilised as in [4]. Once the data density is calculated by the STP and the data owners, the cypher space C is split into t intervals; $C = \{c_1, c_2, \ldots, c_t\}$. The generated interval boundaries are the MUOPE encryption keys.

Distance Matrices (DMs), generated by data owners and described further in Section 4, can be viewed as a large set of linear equations that might be used to reverse engineer aspects of original datasets. Therefore, DMs are encrypted by individual data owners, using MUOPE, to give Encrypted DMs (EDMs). The encryption is conducted as per Eq. 1, where: *dist* is a DM value to be encrypted; i is the interval ID within which a *dist* is contained; l_i, r_i, l'_i and r'_i are the ith interval boundaries; and δ_i is a random number sampled from the range 0 to $Sens \times Scale_i$ where Sense is the data sensitivity as defined in [25] and $Scale_i$ is as given in Eq. 1.

$$Scale_i = \frac{(l'_i - r'_i)}{(l_i - r_i)}, \quad Enc_i(dist) = l'_i + (Scale_i \times (dist - l_i)) + \delta_i \tag{1}$$

4. Global Encrypted Distance Matrices (GEDMs)

A GEDM is a global EDM, founded on the EDM concept first presented in [17], designed to support multi-party collaborative/distributed data clustering. A GEDM is a 2D matrix that holds distances between every record in the global dataset *GData* with every other record in *GData*; $GData = \cup_{i=1}^{i=u} D_i$, where D_i is the dataset belonging to data owner (participant, party) p_i, and u is the number of parties. In other words, *GData* is the union of the participating datasets and a GEDM is the combination of the associated EDMs. The GEDM is constructed by the STP. How the GEDM is constructed depends on how the data is partitioned across participants. In this paper, horizontal data partitioning is assumed, where each partition conforms to the same set of attributes A, but features different records. The GEDM generation process is detailed in this section.

As presented in [17], a Distance Matrix (DM) is a 2D matrix that holds distances (differences) between each record in a dataset D with every other record in D. Therefore, a DM's dimensions are defined by the number of records in D. The matrix is symmetric about the leading diagonal so only values for the lower (or upper) triangular of the matrix are required. An EDM is then an encrypted DM that has been encrypted using the MUOPE scheme described above. Each participant p_i generates an EDM, EDM_i, for their dataset D_i. The global set of EDMs, $\mathbf{EDM} = \{EDM_1, EDM_2, \ldots, EDM_u\}$, are combined to form a GEDM.

Algorithm 1 presents the pseudo code for the pooling method for GEDM generation. The inputs are the global set of EDMs, **EDM**, and the number of attributes in the data set a. The first step (lines 2 and 3) is for the STP to dimension the GEDM according to the number of records in $GData$ (the sum of the number of records in each EDM belonging to each participant). Next, the distance values in each participant's EDM are loaded into the GEDM (lines 5 to 10) thus populating the GEDM "leading diagonal band" with the existing encrypted distances. The next step (lines 11 to 25) is to populate the remainder of the GEDM with the distances between records held by pairs of data owners. The number of pairs will be $\frac{u(u-1)}{2}$. For each pair, a Pooled Matrix (PM) is constructed (line 13). A PM is a 3-D matrix designed to hold the encrypted difference between each attribute in each record held by a data owner (p_i) and each record held by another owner (p_j). The dimensions for a PM are thus: the number of records held by p_j, the number of records held by p_i and the number of attributes a in $GData$ (recall that horizontally partitioned data features the same set of attributes across the participants). The PM will then be populated with a set of random values. Next (line 14) the PM is encrypted to give PM', which is then sent to the data owners p_i and p_j who each create temporary EDMs holding the differences between their encrypted records and the contents of PM'. Data owner p_i will calculate the distance between each value in PM', using their MUOPE cypher, with the corresponding encrypted attribute value in their dataset D_i to produced PM'_1 (line 15); whilst data owner p_j will do the same with respect to their dataset D_j to give PM'_2 (line 16). Both are returned to the STP where they are used to create a pooled EDM by adding the PM'_1 and PM'_2 elements to give $PoolEDM$ (line 17) which is then used to populate the appropriate section of the GEDM. The relevant "start" indexes into the GEDM are first calculated (lines 18 and 19) and then the pooled EDM is processed and used to populate the appropriate section of the GEDM (lines 20 to 25). The process will be repeated until the GEDM has been fully populated.

5. Secure Collaborative DBSCAN (S-DBSCAN)

Using the GEDM and the presented Cryptographic Ensemble, collaborative data mining activities can be outsourced securely to a third party data miner. The standard data mining algorithms can be slightly modified to work with cryptographic ensemble using the GEDM. This is illustrated in this paper in the context of secure data clustering using a variation of DBSCAN [26], Secure DBSCAN (S-DBSCAN). The S-DBSCAN

Algorithm 1 Pooling Methods for GEDM generation

```
procedure PoolingMethod($\mathbf{EDM}$,$a$)
    $GDataSize = \sum_{p=1}^{p=u} |EDM_p|$ $(EDM_p \in \mathbf{EDM})$
    $GEDM$ = 2-D matrix measuring $GDataSize \times GDataSize$
    $currentRow = 0$, $currentCol = 0$
    for $p = 1$ to $p = u$ do
        for $r = 1$ to $r = |EDM_p|$ do
            for $c = 1$ to $c = |EDM_p|$ do
                $GEDM_{[currentRow+r,currentCol+c]} = EDM_{p[r,c]}$
        $currentRow = currentRow + |EDM_p|$
        $currentCol\ = currentCol + |EDM_p|$
    for $i = 1$ to $i = (u-1)$ do
        for $j = (i+1)$ to $j = u$ do
            $PM$ = 3-D matrix measuring $|EDM_j|\times|EDM_i|\times a$ and populated with random values
            $PM' = PM$ encrypted using MUOPE
            Participant $p_i$: $PM'_1$ = differences between $PM'$ and $D'_i$
            Participant $p_j$: $PM'_2$ = differences between $D'_j$ and $PM'$
            $PoolEDM = PM'_1$ and $PM'_2$ combined to form a pooled EDM
            $startRow = \sum_{n=1}^{n=(j-1)} |EDM_n|$
            $startCol\ = \sum_{n=1}^{n=(i-1)} |EDM_n|$
            for $gRow = 1$ to $gRow = |EDM_j|$ do
                for $gCol = 1$ to $gCol = |EDM_i|$ do
                    $v' = 0$
                    for $att = 1$ to $att = a$ do
                        $v' = v' + PoolEDM_{[gRow,gCol,att]}$
                    $GEDM_{[startRow+gRow,startCol+gCol]} = v'$
    Exit with GEDM
```

process is presented in Algorithm 2. The inputs are: (i) the Liu scheme encrypted global dataset $GData'$ collated with reference to the third party; (ii) the GEDM; and (iii) the DBSCAN minimum number of points and radius parameters, $MPts$ and ϵ', agreed by the participating parties. To allow secure comparison using GEDM, the ϵ value is encrypted using MUOPE to give ϵ', hence the third party data miner does not have access to the raw radius value.

The algorithm commences by creating an empty set C, and initialising the number of clusters sofar parameter k to 1 (line 2). For each encrypted record r'_i in $GData'$, which has not been previously assigned to a cluster (is "*unclustered*"), the *RegionQuery* procedure is called to determine the ϵ-neighbourhood set S of the record (line 5). The set S comprises the set of records in $GData'$ whose distance from r'_i is less than or equal to ϵ'. The GEDM is used to provide secure data comparison (line 25 in the *RegionQuery* procedure). In line 6, if the number of records in S is greater than or equal to $MPts$, record r'_i is marked as "*clustered*" and a new cluster C_k is created (lines 7 and 8). The cluster C_k is then expanded by considering the set of records in S using the *ExpandCluster* procedure (line 9). The inputs to the *ExpandCluster* procedure are: the cluster C_k sofar, the list S, the GEDM, the density parameters and the $GData'$. It is a recursive procedure, that adds the records in set S to cluster C_k if a record is "*unclustered*" (lines 15 to 17).

Algorithm 2 Secure DBSCAN clustering algorithm

```
1: procedure S-DBSCAN($GData'$,$GEDM$,$MPts$,$\epsilon'$)
2:     $C = \emptyset$, $k = 1$
3:     for $i = 1$ to $i = |GData'|$ do
4:         if $r'_i$ is Unclustered then
5:             $S$ = RegionQuery($r'_i$,$\epsilon'$,$GEDM$,$GData'$)
6:             if $|S| \geqslant MPts$ then
7:                 mark $r'_i$ as clustered
8:                 $C_k = r'_i$ (new cluster)
9:                 $C_k$ = ExpandCluster($C_k$,$S$,$GEDM$,$\epsilon'$,$MPts$,$GData'$)
10:                $C = C \cup C_k$
11:                $k = k + 1$
12:    Exit with C
13: procedure expandCluster($C$,$S$,$GEDM$,$\epsilon'$,$MPts$,$GData'$)
14:    for $\forall\ r'_i \in S$ do
15:        if $r'_i$ is Unclustered then
16:            mark $r'_i$ as clustered
17:            $C = C \cup r'_i$
18:            $S_2$ = RegionQuery($r'_i$, $\epsilon'$, $GEDM$, $GData'$)
19:            if $|S_2| \geqslant MPts$ then
20:                $C$ = ExpandCluster($C$,$S_2$, $GEDM$, $\epsilon'$, $MPts$, $GData'$)
21:    Exit with C
22: procedure RegionQuery($r'_{index}$, $\epsilon'$, $GEDM$,$GData'$)
23:    $N_\epsilon = \emptyset$
24:    for $\forall\ r'_j \in GData'$ do
25:        if $GEDM_{[index,j]} \leqslant \epsilon'$ then
26:            $N_\epsilon.add(r'_j)$
27:    Exit with $N_\epsilon$
```

The ϵ-neighbourhood for each record added to C_k is retrieved by another call to the *RegionQuery* procedure and returned in list S_2 (line 18). If the size of S_2 is greater than or equal to $MPts$ the *ExpandCluster* procedure is called again; and so on. The S-DBSCAN procedure continues in this way until all the records in $GData'$ have been processed. The algorithm exits with the cluster configuration C (line 12).

6. Experimental Evaluation

The evaluation of the Cryptographic Ensemble the GEDM concepts, together with the pooling method used to create GEDMs is presented in this section. For the evaluation two different kinds of dataset were used, synthetic datasets and UCI machine learning repository [27] datasets. The evaluation objectives were to consider the proposed mechanism in terms: (i) data owners participation, (ii) clustering efficiency, (iii) clustering accuracy, (iv) scalability and (v) security. The operation of the proposed mechanism was compared with the SSCDM mechanism presented in [4]. The proposed mechanism was implemented in Java and all experiments, including the comparison with the SSCDM mechanism, were run using an iMac (3.8 GHz Intel Core i5) running under the macOS High Sierra operating system with 8GB of RAM.

Table 1: The algorithm parameters used in the evaluation and cluster configuration comparison and execution time using DBSCAN, S-DBSCAN founded on GEDM and S-DBSCAN founded on SSCDM

UCI dataset	MPts	ϵ	DBSCAN (Standard)		S-DBSCAN (GEDM)		S-DBSCAN (SSCDM)	
			Sil. Coef.	Exc. Time (**ms**)	Sil. Coef.	Exc. Time (**ms**)	Sil. Coef.	Exc. Time (**Sec.**)
1. Arrhythmia	2	600	0.472	14.4	0.472	51.3	0.472	367.24
2. Banknote Auth.	2	3	0.922	44.2	0.922	264.9	0.922	254.25
3. Blood Trans.	2	10	0.971	28.5	0.971	65.6	**0.976**	4.73
4. Brest Cancer	2	5	0.678	30.8	0.678	44.8	**0.485**	9.60
5. Breast Tissue	2	100	0.628	2.7	0.628	5.5	0.628	0.27
6. Chronic Kid. Dis.	2	70	0.970	12.1	0.970	48.4	0.970	12.64
7. Dermatology	2	10	0.853	12.4	0.853	24.7	**0.881**	5.86
8. Ecoli	2	60	-1.000	33.0	-1.000	43.0	-1.000	4.58
9. Ind. Liver	3	40	0.789	12.0	0.789	56.6	0.789	25.53
10. Iris	5	2	0.722	5.5	0.722	13.7	0.722	0.33
11. Libras Mov.	5	2	0.715	12.0	0.715	39.5	0.715	120.62
12. Lung Cancer	2	20	0.053	1.8	0.053	2.4	0.053	0.01
13. Parkinsons	3	10	0.829	4.4	0.829	14.8	0.829	4.06
14. Pima Ind. Diab.	5	20	0.691	10.9	0.691	74.6	0.691	30.15
15. Seeds	5	1	0.852	5.1	0.852	13.7	0.852	1.43

6.1. Data Owner Participation

Data owners participation was measured in terms of runtime required for: (i) data encryption (*Data Enc.*), (ii) DM calculation (*DM Cal.*), (iii) DM encryption (*DM Enc.*), (iv) calculation of the data density required to dimension the MUOPE cypher space (*Dens. Cal.*), (v) time to generate Liu's scheme key (*HE Key Gen.*) and (vi) the amount of data owner involvement during the clustering process. Ten synthetic datasets, increasing in size from 1000 to $10,000$ records and increasing in steps of 1000, were used for the evaluation. The number of attributes was kept constant at 125. The results are shown in Figure 2. As expected, the time complexity increased in a linear manner as the number of records (r) increased. The reported time for *Data Enc.*, *DM Cal.*, *DM Enc.* and *Dens. Cal.* for r = 1K were 0.02Sec, 0.27Sec, 0.74Sec and 0.44Sec, respectively. The time required when the dataset size increases to r = 10K were 0.36Sec, 62.6Sec, 385.7Sec and 564.9Sec. However, regardless of number of records, the runtimes are not significant and do not introduce any overhead for the participating parties. The *HE Key Gen.* is a one time process that also does not add any overhead on behalf of the data owner, the experiments show that 20.95ms was required for *HE Key Gen.* Once the data owners have encrypted their data and jointly created the GEDM no further data owner participation is required and the entire clustering process is delegated to the third party.

6.2. Clustering Efficiency

The runtimes required to cluster the UCI datasets using standard (unencrypted) DBSCAN and S-DBSCAN were compared to evaluate any potential overhead of the proposed Cryptographic Ensemble and the usage of GEDM to run secure data clustering.

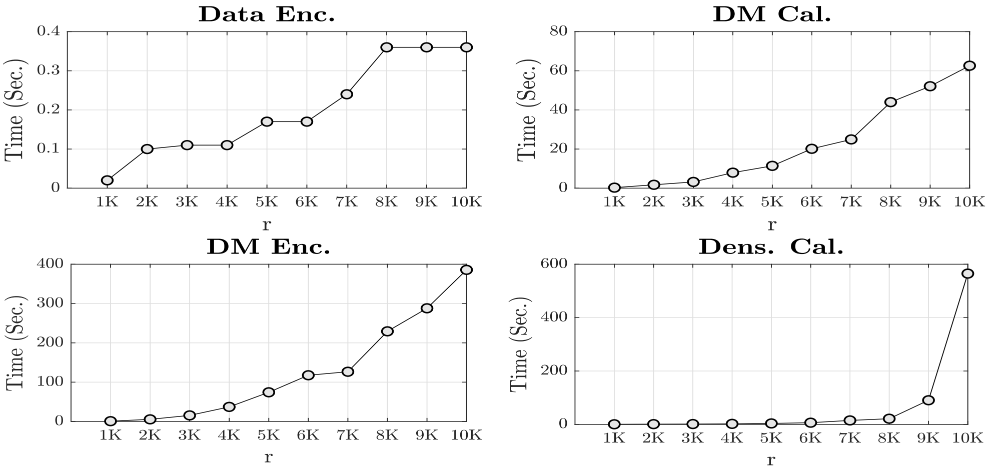

Figure 2: Time required (Sec.) for data owner participation in terms of number of records (r) in a data owner's local dataset

In the reported experiments, the DBSCAN parameters, *MPts* and ϵ, were again as given in columns 2 and 3 of Table 1. As expected, the runtime for S-DBSCAN were larger than for the standard algorithm; however, inspection of the table indicates that this did not present a significant overhead. For example, the largest dataset, Banknote Auth., required 44.2ms for DBSCAN compared to 264.9ms for S-DBSCAN.

6.3. Clustering Accuracy

The clustering configuration "correctness" was measured by comparing the final clustering configuration results obtained using standard DBSCAN with those obtained using the proposed S-DBSCAN mechanism. To confirm that the secure algorithm operated correctly, the intuition was that S-DBSCAN should produce comparable configurations to those obtained by standard DBSCAN. The Silhouette Coefficient (*Sil. Coef.*) [28] was used as the evaluation metric. The reported *Sil. Coef.* are presented in columns 4 and 6 of Table 1. From the table, it can be seen that the clustering configuration produced using S-DBSCAN were identical to those generated by the standard approach. In other words, the application of the various security mechanisms did not adversely affect the clustering effectiveness.

6.4. Scalability

The scalability of the proposed collaborative clustering was measured by considering the effect on time complexity as the number of participants increased. In the proposed approach, the collaboration between data owners occurs when: (i) generating the MUOPE encryption key (*Key Gen.*) and (ii) deriving the Pooled EDMs to arrive at a GEDM (*GEDM Gen.*). A sequence of experiments was conducted whereby the number of participating data owners was increased from 10 to 100 in steps of 10; for completeness the experiment also considers two and four data owners. Synthetic datasets were used,

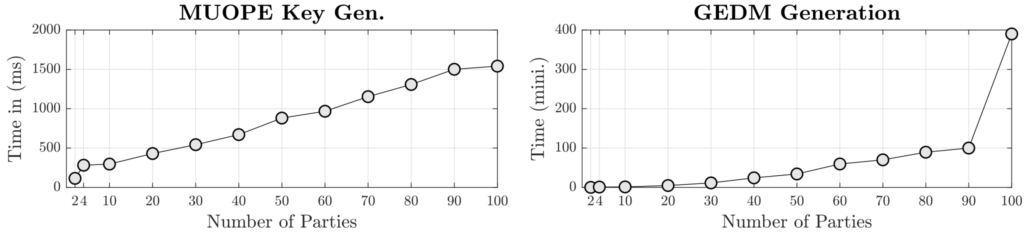

Figure 3: Runtime to generate MUOPE keys and the GEDM as the number of participating parties (data owners) increases

comprised of $10,000$ data records and 125 attributes equally distributed among the participation. The results are presented in Figure 3. From the figure it can be seen that the MUOPE *Key Gen.* time was negligible, even for 100 participants only 1541.39ms was required. As expected, the time required for *GEDM Gen.* increased with the number of participants. The reason was that the increases in the number of data owners increase the number of pairs and then the number of Pooled EDMs that need to be generated.

6.5. Security

The third party data miner is considered to be a "passive adversary" who follows the semi-honest model where the proposed S-DBSCAN process is honestly executed. This was considered to be a reasonable assumption since the main objective of the third party (who provides the DMaaS) is to deliver a high quality and accurate services to clients (data owners). In the proposed solution, the data and GEDM were encrypted and no decryption takes place at the third party side. Therefore, the security of the proposed solution relies on the security of the adopted Cryptographic Ensemble; thus Liu's scheme and the MUOPE scheme. Potential attacks that can be directed at the proposed solution: (i) Cyphertext Only Attacks (COAs) when the adversary somehow has access to the encrypted data and/or the GEDM and (ii) Overlapping Attacks (OAs) when the third party compares the distances with a DBSCAN radius parameter. In terms of COAs over encrypted datasets, Liu's scheme has been shown to be semantically secure [13]. This feature makes deriving any potential aspect of the plaintext from cyphertexts computationally expensive, rendering the likely success of COAs over Liu's scheme cyphers to be negligible. In terms of GEDMs, 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 plaintexts; but only if examples were available. As a countermeasure, MUOPE uses "message space splitting" and "non-linear cypher space expansion" to hide the data distribution, and a "one-to-many" encryption function to hide data frequency; which makes inferences using COAs difficult. OAs are precluded by encrypting the raw data used in the clustering and the DBSCAN radius parameter, which means that the third party compares the "order" of distance values and not the original distance values.

6.6. Comparison of GEDMs with Other Approaches

The operation of the GEDM concept was compared with the SSCDM concept presented in [4]; both provide the potential for collaborative data mining without entailing

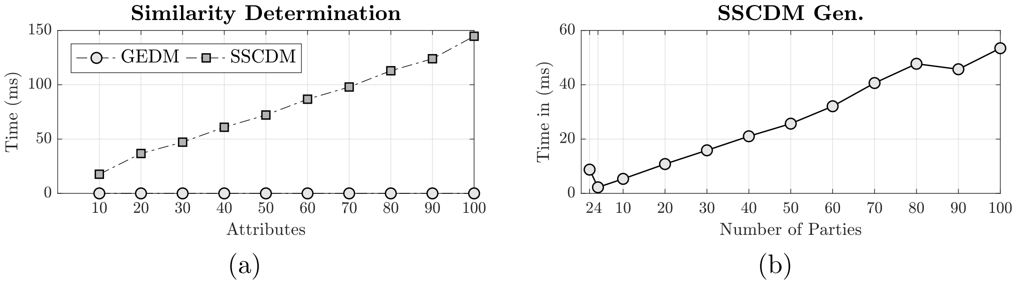

Figure 4: (a) Time (ms) for third party to determine similarity between two records using the GEDM and the SSCDM, (b) time (ms) to generate the SSCDM with respect to different data owners

any data owner participation. The comparison was conducted according to the following criteria; (i) accuracy of proposed solution in the context of DBSCAN, (ii) efficiency, (iii) data owner participation when preparing data for outsourcing and (iv) required memory resource; each discussed in further detail below.

Accuracy: The results obtained using S-DBSCAN founded on the Cryptographic Ensemble and GEDM were compared with those obtained using DBSCAN founded on the concept of SSCDMs (S-DBSCAN$_{sscdm}$). The correctness of the clusters produced using S-DBSCAN and S-DBSCAN$_{sscdm}$ was again compared with those using standard DBSCAN. The same DBSCAN parameters ($MPts$ and ϵ) as shown in columns 2 and 3 of Table 1 were used. It was found that S-DBSCAN$_{sscdm}$ produced clustering configuration comparable with those produced using S-DBSCAN and standard DBSCAN, but not the same (columns 4, 6 and 8). The reason for these difference, as reported in [4], was due to the "chain feature" used to determine similarity. Although S-DBSCAN$_{sscdm}$ produces comparable results, and in some cases slightly better configurations, the effects of chain features were not evaluated in the context of larger datasets. Therefore, the results produced by proposed Cryptographic Ensemble and GEDM are stable as the results produced was always the same as the standard algorithm.

Efficiency: The efficiency (in terms of execution runtimes) of proposed solution was also compared with the solution presented in [4]. Columns 7 and 9 of Table 1 show the runtimes for clustering the UCI datasets using S-DBSCAN and S-DBSCAN$_{sscdm}$ respectively. The runtimes required to cluster datasets using S-DBSCAN is much lower than the time required using S-DBSCAN$_{sscdm}$; note that the time for S-DBSCAN$_{sscdm}$ is given in *Sec.* whilst for S-DBSCAN in *ms*. The reason for this difference is due to the time required to utilise the SSCDM for determining similarity. The time required to use the GEDM and SSCDM for determining similarity between two data records were also compared by considering datasets with $10K$ records but with varying numbers of attributes from 10 to 100 increasing in steps of 10. Figure 4(a) shows the required runtime for determining the similarity between the first and the last record in each dataset. Regardless of number of attributes, the GEDM features an almost steady runtime to determine similarity, whilst the runtime required when using the SSCDM increased linearly with the number of attributes and was always higher than the time of using the GEDM.

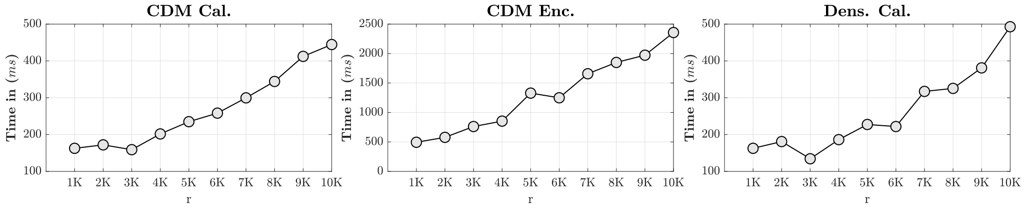

Figure 5: The complexity of data owner participation in the context of SSCDM

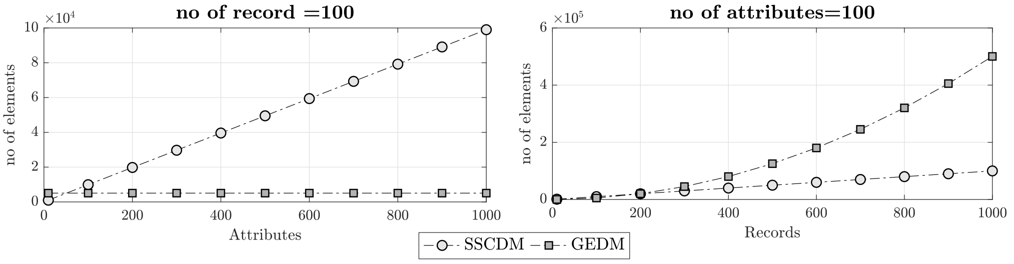

Figure 6: Number of elements in GEDM and SSCDM for different size of data

Data owner participation: The data owner participation using S-DBSCAN$_{sscdm}$ was compared with the S-DBSCAN data owner participation as presented in Sub-section 6.1. In the case of S-DBSCAN$_{sscdm}$ the data owner will participate in: calculating the CDM (*CDM Cal.*), CDM encryption (*CDM Enc.*) to arrive at Secure CDM (SCDM) and Density Calculation (*Dens Cal.*). Figure 5 shows the data owner participation in the context of SSCM and Figure 2 in the context of GEDM. The GEDM data owner participation is higher than that required by the SSCDM, but closer inspection of the figure indicates that this difference is not significant. The runtime required to generate the SSCDM is given in Figure 4(b) which is clearly less than the runtime required by generating the GEDM. However, the argument here is that this operation is one time process that is conducted before the data is outsourced to the third party and thus it will not introduce any significant overhead.

Memory resources: GEDMs and SSCDMs are both 2D matrices. The GEDM's two dimensions are correlated with the number of records in the dataset. However, due to the similarity around their leading diagonal only the upper or lower triangle of the GEDM is required. The SSCDM's first dimension is correlated with the number of data records, whilst the second dimension is correlated with the number of attributes. The number of elements for a range of GEDMs and SSCDMs, associated datasets featuring different numbers of records and attributes, are shown in Figure 6. The GEDM is more suited to dataset where the number of attributes is larger than the number of records, whilst the SSCDM is suited to datasets where the number of records is larger than the number of attributes (the more usual case).

7. Conclusion

This paper has proposed the Cryptographic Ensemble and the GEDM concepts for secure collaborative data clustering in a manner which, unlike existing solutions, does not require data owner participation when the clustering is undertaken by a third party data miner without loss of accuracy. The approach was illustrated using DBSCAN. The Cryptographic Ensemble encompassed two encryption schemes, Liu's HE scheme and the Multi-User Order Preserving Encryption (MUOPE) scheme. The approach offers three advantages: (i) use of the Cryptographic Ensemble and the GEDM obviates the need for the involvement of data owners in the clustering process; (ii) the accuracy of the clustering is not adversely affected by the Cryptographic Ensemble and (iii) the approach provides a general solution suited to many forms of secure data mining not limited to data clustering as presented in this paper. Moreover, use of the Cryptographic Ensemble and the GEDM was found to be comparatively efficient and scalable.

References

[1] R. Agrawal, R. Srikant, Privacy-preserving data mining., in: Proceedings of the 2000 SIGMOD International Conference on Management of Data, ACM, 2000, pp. 439–450.

[2] Y. Lindell, B. Pinkas, Privacy preserving data mining., Journal of Cryptology 15 (3) (2002) 177–206.

[3] O. Goldreich, Secure multi-party computation., Manuscript. Preliminary version 78.

[4] N. Almutairi, F. Coenen, K. Dures, Secure third party data clustering using ϕ data: Multi-user order preserving encryption and super secure chain distance matrices., in: Artificial Intelligence XXXV, Springer, Cham, 2018, pp. 3–17.

[5] J. Liu, L. Xiong, J. Luo, J. Z. Huang, Privacy preserving distributed DBSCAN clustering., Transactions on Data Privacy 6 (1) (2013) 69–85.

[6] P. Samarati, Protecting respondents identities in microdata release., IEEE Transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.

[7] S. Xu, X. Cheng, S. Su, K. Xiao, L. Xiong, Differentially private frequent sequence mining., IEEE Transactions on Knowledge and Data Engineering 28 (11) (2016) 2910–2926.

[8] L. Liu, M. Kantarcioglu, B. Thuraisingham, The applicability of the perturbation based privacy preserving data mining for real-world data., Data and Knowledge Engineering 65 (1) (2008) 5–21.

[9] S. Berinato, There's no such thing as anonymous data., Harvard Business Review February (2015) 2 – 4.

[10] A. Narayanan, V. Shmatikov, Robust De-anonymization of large sparse datasets., in: Proceedings of the 2008 IEEE Symposium on Security and Privacy, IEEE, 2008, pp. 111–125.

[11] Z. Huang, W. Du, B. Chen, Deriving private information from randomized data., in: Proceedings of the 2005 SIGMOD International Conference on Management of Data, ACM, 2005, pp. 37–48.

[12] X. Sun, H. Wang, J. Li, J. Pei, Publishing anonymous survey rating data., Data Mining and Knowledge Discovery 23 (3) (2011) 379–406.

[13] D. Liu, Homomorphic encryption for database querying, Patent 27 (PCT/AU2013/000674), iPC_class = H04L 9/00 (2006.01), H04L 9/28 (2006.01), H04L 9/30 (2006.01).

[14] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes., in: Proceedings of EuroCrypt, Springer, 1999, pp. 223–238.

[15] C. Cachin, Efficient private bidding and auctions with an oblivious third party., in: 6th ACM Conference on Computer and Communications Security, ACM, 1999, pp. 120–127.

[16] A. C. Yao, Protocols for secure computations., in: 23rd Annual Symposium on Foundations of Computer Science, IEEE, 1982, pp. 160–164.

[17] N. Almutairi, F. Coenen, K. Dures, Third party data clustering over encrypted data without data owner participation: Introducing the encrypted distance matrix., in: International Conference on Big Data Analytics and Knowledge Discovery, Springer, 2018, pp. 163–173.

[18] D. Jiang, A. Xue, S. Ju, W. Chen, H. Ma, Privacy-preserving DBSCAN on horizontally partitioned data., in: International Symposium on IT in Medicine and Education, IEEE, 2008, pp. 1067–1072.

[19] K. A. Kumar, C. P. Rangan, Privacy preserving DBSCAN algorithm for clustering., in: International Conference on Advanced Data Mining and Applications, Springer, 2007, pp. 57–68.

[20] M. E. Nergiz, C. Clifton, Thoughts on k-anonymization., Data and Knowledge Engineering 63 (3) (2007) 622 – 645.

[21] T. Zhu, G. Li, W. Zhou, S. Y. Philip, Differentially private data publishing and analysis: A survey., IEEE Transactions on Knowledge and Data Engineering 29 (8) (2017) 1619–1638.

[22] M. S. Rahman, A. Basu, S. Kiyomoto, Towards outsourced privacy-preserving multiparty DBSCAN., in: 22nd Pacific Rim International Symposium on Dependable Computing, IEEE, 2017, pp. 225–226.

[23] B. K. Samanthula, Y. Elmehdwi, W. Jiang, k-Nearest Neighbor classification over semantically secure encrypted relational data., IEEE Transactions on Knowledge and Data Engineering 27 (5) (2015) 1261–1273.

[24] N. Almutairi, F. Coenen, K. Dures, Data clustering using homomorphic encryption and secure chain distance matrices., SciTePress, 2018.

[25] D. Liu, S. Wang, Nonlinear order preserving index for encrypted database query in service cloud environments., Concurrency Computation: Practice and Experience 25 (13) (2013) 1967–1984.

[26] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 226–231.

[27] M. Lichman, UCI machine learning repository (2013).

[28] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis., Journal of Computational and Applied Mathematics 20 (1987) 53–65.

Nawal Almutairi is a lecturer within the information technology Department at King Saud University and registered for a PhD with the Department of Computer Science University of Liverpool. She has a general background in data mining, security and AI. Her research interest include the Privacy Preserving Data Mining (PPDM), Homomorphic Encryption (HE), Property Preserving Encryption (PPE), Data Mining as a Service (DMaaS) using Cloud facilities and collaborative data mining.

Frans Coenen has a general background in AI, and has been working in the field of data mining and Knowledge Discovery in Data (KDD) for the last fifteen years. He is interested in the application of the techniques of data mining and Knowledge Discovery in Data to unusual data sets, such as: (i) graphs and social networks, (ii) time series, (iii) free text of all kinds, (iv) 2D and 3D images, particularly medical images, and (v) video data. He is also interested in data mining over encrypted data. He currently leads a small research group working on many aspect of data mining and KDD. He has some 390 refereed publications on KDD and AI related research, and has been on the programme committees for many KDD conferences and related events. Frans Coenen is currently professor within the Department of Computer Science at the University of Liverpool where he is the director for the University of Liverpool Doctoral Network in AI for Future Digital Health.

Keith Dures is a lecturer in Computer Science within the Department of Computer Science at the University of Liverpool; Assistant Director of Studies for online MSc programmes. He is chair of the IT Sub-Group (University-wide), Admissions tutor, Internal examiner, Module co-ordinator and CPD lead for the department. He is a member of the Teaching And Scholarship in Computing (TASC) group with interests in research and development with respect to teaching and learning. Research interests in knowledge discovery in databases, data mining, software engineering and cyber security. Professional exposure (includes British Computer Society and Higher Education Academy), includes responsibility for academic and commercial course development, teaching, supervision and examination at all levels.