International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Jan-24)

http://dx.doi.org/10.12785/ijcds/150101

A Secure Cloud Framework for Big Data Analytics Using a Distributed Model

Zainab Salman[1], Alauddin Alomary[1] and Mustafa Hammad[2]

[1]*College of Information Technology, University of Bahrain, Sakhir, Bahrain*
[2]*Mutah University, Al-Karak, Jordan*

Received 3 Jan. 2023, Revised 24 Oct. 2023, Accepted 21 Dec. 2023, Published 1 Jan. 2024

Abstract: As technology is improving and changing rapidly, cloud security has become a challenging task. Consequently, there is a need for more powerful and robust techniques to secure the cloud. Meanwhile, due to the huge size of the provided big data on the cloud, other techniques and methods should be utilized to improve big data analytics and processing. The paper aims to provide a framework for secure and efficient processing and analysis of big data using a double layer of security that is based on Elliptical Curve Cryptography (ECC) and Fully Homomorphic Encryption (FHE). Additionally, a distributed model has been defined to partition big data into smaller data sizes processed by different numbers of virtual CPUs. In the defined distributed model, many virtual machines process different partitions of data parallelly and simultaneously to speed up the processing time of data. KMeans clustering algorithm is used in three datasets as an instance of data analytics to test the suggested framework. Furthermore, the produced results are compared with a centralized-based model to assess the productivity and efficiency of the distributed model. Besides, the *principal component analysis* (PCA) is applied to the used clustering algorithm to diminish the required clustering time by the distributed model. The results indicate that the clustering time can be reduced by up to 91%, and even with 18% more reduction in the execution time using the distributed model. The recommended solution can improve the effectiveness of big data analytics while guaranteeing the security of such data.
Keywords: Cloud security, big data analytics, hybrid encryption, KMeans clustering, principal component analysis, distributed model

1. INTRODUCTION

Security is essential and crucial for many communities especially sensitive organizations and institutions, such as the banking and governmental sectors. Any small gap or limitation in security may cost millions of dollars and disasters for the companies and their customers as well. Furthermore, any doubt in the level of security will affect the adoption of cloud computing and its underlying services.

To extract information and make decisions, data analytics has been used and applied. Big data is growing exponentially with the usage of IoT devices and social media. Therefore, critical issues come out such as the traditional databases cannot deal with a huge amount of data and store it. Also, traditional analytical tools cannot store and process data. Cloud computing comes as a solution to big data that can provide the required facilities and computing services by providing unlimited and on-demand resources. Furthermore, big data owners are worried about their data with an increment of data breaches, hackers, and the presence of third parties in the cloud [1].

One of the public key encryption methods that is used widely in data transition and networks is Elliptic Curve Cryptography (ECC). It is a lightweight encryption method that needs a small key size to generate an equivalent level of security in contrast with other encryption algorithms. As it needs fewer key sizes, therefore fewer resources and storage sizes are required. ECC can be used in many applications such as image encryption, Internet of Things (IoT), OpenSSL protocols, and Bitcoin digital signatures. Additionally, ECC has been used wildly in network transitions, digital signatures, and even the standard encryption process. It can be used in devices with limited storage capacity and resources [2].

One of the powerful ways to secure the cloud could be by employing a cloud-based encryption method such as Homomorphic Encryption (HE). FHE can be utilized as a capable public-key encryption method that permits users to do computations on encrypted data without any need to decrypt it. It can perform all kinds of operations and has a wide range of applications [3]. while using HE, the data owner is the only user who can access the plaintext data. Therefore, HE can preserve the privacy of on-cloud data. Moreover, HE can support data mining operations that are required for big data.

To work with big data with thousands or even millions of records, other techniques and methods should be used

E-mail address: zsalman@uob.edu.bh, aalomary@uob.edu.bh, mhammad@mutah.edu.jo

to reduce the processing time of big data. For instance, we need clustering techniques in a distributed environment to split up big data into smaller data parts and then process all the parts simultaneously. This can improve the analysis performance of big data [4]. One of the well-known methods in clustering is using the KMeans algorithm. Many different KMeans algorithms have been defined as researchers trying to propose more efficient and accurate algorithms. The standard KMeans clustering algorithm is easy to use, and it is specialized for big data clustering [5]. At the same time, it is low in cost and compatible with cloud and unstructured data.

This paper ensures strong data security for big data analytics in the cloud. The data is encrypted using ECC and then FHE. A distributed model is developed and tested using different sizes of datasets. To evaluate and compare the performance of the proposed solution in the distributed model, a centralized-based model has been developed. Moreover, the KMeans clustering algorithm and PCA method are integrated to reduce the execution time of big data analytics in the developed framework.

The following sections are structured as follows: Section 2 points out the objectives of the research. Section 3 investigates the literature review of related works. Besides, Section 4 illustrates the proposed framework by explaining the main processes. Moreover, Section 5 describes the used datasets, and Section 6 shows the designed platform and setups. Besides, Section 7 explains the experimental results and findings. A general discussion is provided in Section 8 and in Section 9, the conclusion and the future works are provided.

2. Research Objectives and Methodology

This paper proposes a secure structure for cloud data analytics computations by employing a hybrid encryption approach. The prime objective of the framework is to preserve the privacy of big data that resides in the cloud and speed up its analytics. The proposed framework has the following objectives:

1) To provide a solution for big data processing and privacy-preserving problems and use a hybrid encryption system based on ECC and FHE.
2) To provide more privacy using multi-cloud architecture to save sensitive information in a private cloud.
3) To provide a choice of decrypting the computed results. Only the data owner can access the results and plaintext data.
4) To speed up the big data analysis by clustering the encrypted data in a distributed computing environment.
5) To use the Principal Component Analysis (PCA) method as a feature selection technique that can be used to reduce the size of attributes in the used datasets and therefore, reduce the clustering time.
6) To test the processing time of the used clustering algorithm and compare it with a similar one in a centralized model to approve its efficiency over the traditional centralized model.

3. Related works

Securing big data is a key challenge for many organizations and systems. The work in [6] focused on presenting fundamental issues in securing big data in medical and healthcare systems. The authors discussed some data production rules in different countries aiming to address the legal responsibilities and available risks in such countries. As another example, the study in [7] highlighted different privacy-preserving techniques by explaining different cryptographic algorithms and protocols and comparing them in terms of cost and time of their performance. Moreover, securing big data schemes for cloud tenants and Map-Reduced clouds has been examined. Another work in [8] investigated different techniques for data confidentiality and privacy, such as anonymization, data encryption, and access control methods. Furthermore, IoT security and privacy were explored from different aspects, such as cryptographic protocols, network security, and application security. Besides, the work in [9] presented the most important issues in cloud security. Different encryption and security methods were reviewed with defining a taxonomy for reviewed studies and research in securing the cloud.

Traditional privacy-preserving methods no longer protect big data. Consequently, many researchers tried to come up with new techniques and methods. Wei Fang et al. [10] surveyed some new challenges of big data privacy by defining suitable countermeasures and legal measures. Furthermore, the work in [11] introduced a novel privacy-preserving approach as a differential privacy technique that is a noise-based method. In this approach, privacy can be achieved by adding a suitable amount of noise to data. Moreover, [12] presented a secure setting for big data analytics using the clustering method to decrease the execution time of data.

Elliptic curve cryptography was used in many studies as a booster for homomorphic encryption as it uses fewer key sizes and execution times compared to other schemes used for homomorphic encryption. For instance, the work in [13] proposed homomorphic encryption that is based on ECC to improve the communication and energy costs along with securing cloud computing. In another research [14], a study was conducted to show that fully homomorphic encryption is impractical for large datasets, and to improve the execution time of homomorphic encryption a new protocol was proposed that is based on ECC.

Homomorphic Encryption (HE) has been used for securing big data analytics in the cloud. For instance, the work in [15] proposed a cloud-based framework for securing big data, which employs Fully Homomorphic Encryption (FHE). The suggested solution is to partition into smaller data so that each part can work independently. Similarly, the work in [16] suggested a privacy-preserving clustering

approach using HE along with different clustering methods. In this study, the results for different clustering algorithms have been compared according to cluster evaluation metrics. Besides, the work in [3] presented a review paper for different types of HE algorithms showing the most important properties of HE.

KMeans and KMeans++ clustering algorithms have been used in many applications due to their efficiency in parallelizing processes and their low computational cost in big data analytics. One of its uses is in medical applications. One example is [17], in which a hybrid clustering model for medical applications has been proposed to overcome the limitation of the KMeans algorithm in defining overlapping clusters. Similar work was conducted in [18] that KMeans used for medical image segmentation. Besides, other traditional KMeans usages are presented and compared with the proposed improved KMeans algorithm.

This study provides a sophisticated framework for big data analytics that uses a double layer of encryption. The proposed framework differs from other works in the literature by combining the encryption and the clustering methods proposed in [19], [20], [1], and [15]. For example, the work in [20] defined an encryption system that used the ECC and FHE methods, whereas the work in [15] used a clustering method based on the KMeans algorithm and Fuzzy C-means clustering (FCMC). In this paper, the proposed framework implements and evaluates a secure framework with a hybrid encryption system that can improve big data analytics by using a set of VMs in a distributed computing environment. This research intends to reduce the clustering time of data when it is compared with a centralized-based model.

4. Proposed Framework

In this section, the main processes of the suggested framework have been presented. Figure 1 depicts the important processes of the suggested framework. *ECC-FHE Encryption*, *Data Clustering*, and *FHE-ECC Decryption* are the main processes performed by the developed framework. In ECC-FHE Encryption, a hybrid encryption system is used to secure the framework in which data is encrypted using ECC and FHE. The original data can be accessed only by the data owner. First, the data owner encrypts the original data using ECC. Then, the data is re-encrypted using FHE to maintain an extra security layer. Encrypting data is performed by the data owner in a private cloud. After data encryption, the generated ciphertext is delivered to a public cloud. In the Data Clustering process, any authenticated data user or even the cloud itself can use the encrypted data to perform the clustering process in the public cloud. The data user never has access to plaintext as he only works on encrypted data. The data user can only use encrypted data to do some data analytics such as the clustering process and then reply with the clustering results. In the FHE-ECC Decryption process, the data decrypts the results and the data. The encrypted data first is decrypted with FHE and then the ECC decryption methods. Note that the encryption and decryption processes only can be executed by the data owner in a private cloud. The following sub-sections explain the main processes of the proposed framework.

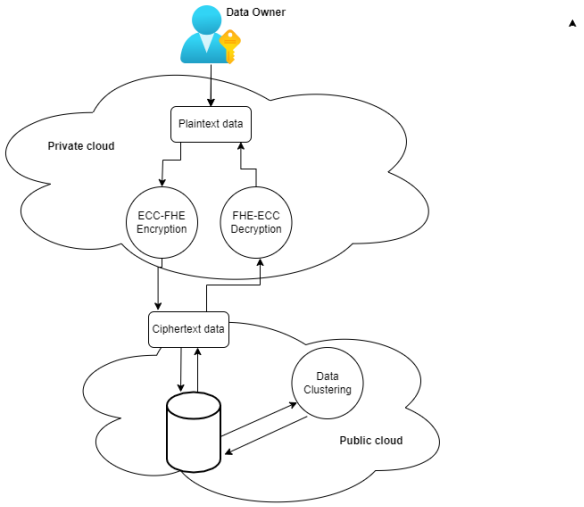

Figure 1. The major processes in the suggested secure big data analytics framework

A. *ECC-FHE Encryption*

The process of encryption is performed only by the data owner. A hybrid encryption method is executed on data using ECC and then FHE methods. ECC has been chosen as a light encryption system with a good security level and is based on the hardness of different problems. ECC can obtain an equivalent security degree with a minimum cost in contrast with other types of encryption methods. The way ECC encryption works is that some points are defined on the elliptic curve using plaintext as an input to the algorithm. First, the data is converted to some integers, then, these integers are calculated as some points on the specified curve. For ECC decryption, the points on the elliptic curve are converted to some integer values, then to plaintext or the original data in the end.

As the second layer of encryption, FHE is applied to the provided ciphertext from ECC encryption. This step enables the ciphertext to be used in the cloud for data analytics like clustering hence FHE can support arbitrary computations. Therefore, FHE can be considered a suitable solution for providing privacy on the cloud. Four major operations in FHE are defined. They are defined as private and public key generation, encryption, decryption, and evaluation processes. In the key generation process, the private and public keys are selected. In the encryption and decryption process, the plaintext data is encrypted and decrypted. Furthermore, in the evaluation process, the property of HE is defined [12]. Algorithm 1 demonstrates the pseudo-code for the defined encryption process.

As shown in Algorithm 1, the original data *Dp*, and the

private key *Keypr* are the inputs to the algorithm. The output is the encrypted data *Dee* after encrypting using both ECC and FHE. A generator function *G* is used to create the public key *Keypu*. As a result of this encryption, the ciphertext *Ci* is generated using a random value *Ra* and generated function *G*. The data is encrypted with *Ra*, the public key *Keypu*, and the created point on the elliptic curve *P*. It is notable, that ECC works on the prime field of values to generate the points on the elliptic curve. Given parameters to the ECC algorithm define the shape of the generated elliptic curve and the numbers that exist on the curve. The input data to the ECC is defined as some points on the calculated curve. The next step is to encrypt *Ci* using FHE which is generated from ECC encryption. After processing FHE, *Ci* is converted to *Dee* to add more level of security to the data. Then the generated Dee is safe to be uploaded to the public cloud for any data analytics.

Algorithm 1 ECC-FHE Encryption

Inputs: Original data (Dp), and Private Key (Keypr)
Output: Double encrypted data (Dee)

1) The following formula is used to generate the public key using the private key:
   $Keypu = Keypr * G$, where a generator function G is calculated from the elliptic curve calculation.
2) A random 4-bit value 'Ra' is used to create Ci.
   $Ci = Ra * G$.
3) Then, De is calculated using the following formula:
   $De = (Ra * Keypu) + (Dp, P)$,
   P is a calculated point presented on the defined elliptic curve.
4) Considering that De is the encrypted data, (FHE) is executed on De as:
   FHE(De)
5) Then, (Dee) is the output of the algorithm.
   $Dee = FHE(De)$
6) In the end, Dee is uploaded to the public cloud for data computation and can be used by any authenticated data user.

B. Data Clustering

A distributed model has been defined in this study to process big data with smaller portions of data. As more CPUs or VMs work on data, therefore, less analytic time is needed to accomplish the clustering process. Consequently, big data analytics turns into a more efficient process using the distributed model. In this model, many CPUs work together to accomplish the data analytics process. The distributed model can improve the clustering process by working on each division of data separately and simultaneously. The proposed distributed method can increase the efficiency of the clustering process compared with the traditional centralized model.

Figure 2 depicts the general idea of the distributed model using clusters of data and processors. In this approach, the big data is divided into smaller parts of data and then processed using a set of VMs that work on each cluster of data in parallel and independently. Firstly, the data parts are distributed among many VMs. Each VM processes the data using the clustering method. Finally, the final result is provided using renormalization and reconciliation processes [12].

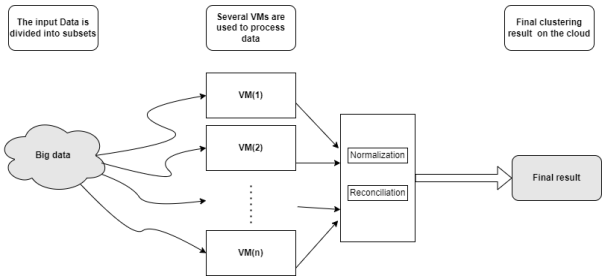

Figure 2. The defined distributed clustering process in general

C. FHE-ECC Decryption

The decryption process is executed by the data owner upon his/her request. Algorithm 2 shows the pseudo-code of the defined decryption process [20]. The output of the decryption process is the original text and the results of data processing. For the decryption process, first FHE is applied to the ciphertext. Then, the ECC decryption process is applied to calculate the output from FHE decryption to the original data. In this process, to decrypt the data, the private key is used. The Encrypted data *Dee* and the private key *Keypr* are the inputs to the decryption process. First, the FHE applied to *Dee*. As a result of this decryption, *De* is calculated. Then, the private key *Keypr* is used to calculate the cipher *Ci*. A randomly selected number *Ra* is calculated from the generator function *G*. It can be provided from the elliptical curve and the private key *Keypr*. Then, the data points on the elliptic curve are used to maintain the original data *Dp*.

Algorithm 2 FHE-ECC Decryption

Inputs: Encrypted data (Dee), Private Key (Keypr)
Output: Original data (Dp)

1) Calculate DE by applying FHE on Dee as:
   $FHE(Dee) = De$
2) Consider the Private key (Keypr)
3) Calculate the ciphertext Ci using De:
   $Ci = Keypr * De$
4) Use generator function G to provide the random value Ra using the elliptic curve equation:
   $Ci = Keypr * (Ra * G)$
5) Calculate De using the formula:
   $P(De) = Ci + De - Ci$, consider that the elliptical curve is used to calculate P.
6) The original data (Dp) is calculated using the defined points P on the curve.
   $Dp = P(De)$

5. Description of Used Datasets

The results of this study are based on three datasets Bank, Marital, and Flight dataset. They have taken from the Kaggle website [21]. Different sizes of data have been selected to explore the result of the distributed model on the processing time. For example, the Bank dataset is used as a smaller dataset, the Marital dataset is used as a medium dataset, and the Flight dataset is used as a bigger dataset. Table I shows each used dataset with its properties. All the selected columns have positive integer values to be compatible with the developed framework for encryption and decryption purposes. The datasets were accessed on 27 Aug. 2021.

6. Designed Platform and Setups

The explained framework in Section 4 has been implemented on the cloud. To this end, the AWS cloud platform has been used to conduct the experiments. According to the explained contribution of this research, a hybrid encryption system has been executed on the used datasets to encrypt the data. After data encryption, a clustering method has been used to cluster the data and count the clustering time for different data portions and different numbers of CPUs. For this purpose, the models have been designed using Python in Tensorflow. The Elastic Compute Cloud (EC2) [22] has been installed as a cloud-based virtual machine on AWS. Furthermore, for clustering purposes, different numbers of resources are used on AWS. Every single used CPU has 2 GiB Memory with up to 6.25 Gbps (Network Bandwidth) and a core turbo frequency of 3.5 GHz. They are powered by 3rd generation Intel Xeon Scalable processors and are used for compute-intensive workloads.

7. Experimental Results

The experiments have been conducted for the Bank, Marital, and Flight datasets to explore the effect of using different sizes of datasets on the clustering process and the distributed model. The distributed model has been designed to include 4, 8, 16, 32, and 64 virtual CPUs according to the selected resource type on AWS. Doubling the number of resources in each iteration is to find out the effect of increasing resources on the efficiency of the distributed approach. The experiments are performed for the distributed model using two different clustering algorithms, KMeans only and KMeans using the PCA method. In each experiment, the proposed distributed environment has been evaluated using the centralized model with only one CPU to show how the suggested distributed model outperforms the centralized model.

It is noteworthy that PCA is used as a dimension reduction technique [23] to optimize the performance and decrease the clustering time. PCA works on the features or columns of the selected dataset and then decreases them to the most relevant ones according to their effect on the whole dataset. The upcoming sub-sections show the results of conducted experiments on the selected datasets.

A. Clustering Results for the Bank Dataset

The defined framework has been tested with the Bank dataset using the AWS cloud platform. Varying portions of data have been used to contain 2000, 4000, 6000, 8000, and 10128 records from the Bank dataset, and different numbers of CPUs have been used to execute the secured proposed clustering framework.

Table II shows the clustering time for varying sizes of data and the numbers of used CPUs to accomplish data clustering. It shows how using cloud resources affects the execution time of the proposed solution. The provided results have approved that increasing the number of used CPUs can decrease the clustering time. Moreover, when data size increases, the clustering time increases as there is more data to be processed.

To assess the efficiency of the distributed model, the centralized model has been used. In this model, only one CPU executes the proposed framework using the same settings and platform in the distributed model. As it is clear in Table II, there is a huge difference in performance between our proposed distributed model using 4, 8, 16, 32, and 64 CPUs and the centralized model with only one CPU. The distributed model has higher efficiency in performance with very little clustering time. In the centralized model, only a single block of memory is used to support computational results whereas, in the proposed distributed model, each VM has its memory block which makes the process of clustering faster hence all used VMs run at the same time to execute the clustering process. As an example, the distributed model within 64 VMs can analyze 2000 data points in 1.44 seconds only, whereas, in the centralized model, a single CPU takes 344 seconds to accomplish the same job. According to the results in Table II, in the distributed model, the clustering time of the whole data can be decreased by up to 86% using 64 CPUs instead of 4 CPUs. Furthermore, as the data size is growing, the clustering time reduction is greater. It is obvious that the suggested distributed model outperforms the centralized model and makes it impractical for real-world applications hence millions, billions, or even trillions of data records need to be analyzed and processed.

As the next part of the experiment, the PCA method has been added to find out its effect on the suggested framework. Table III depicts the results of adding the PCA method on the Bank dataset with the same settings in the previous experiment. As it is clear, the clustering time results show 15% more reduction compared with the results in Table II using the PCA method in the distributed model.

B. Clustering Results for the Marital Dataset

To evaluate the proposed framework with a bigger dataset, the Marital dataset has been used with more than 40000 records. Different portions of data have been tested to contain 8500, 17000, 25500, 34000, and 42661 records from the Marital dataset. Table IV shows the clustering time for different data sizes and numbers of used CPUs to perform the clustering process in the centralized and distributed

TABLE I. The used datasets with their properties

Dataset Name	No. of Selected Records	No. of Selected Feilds
Bank Dataset	10128	11
Marital Dataset	42661	20
Flight Dataset	107316	17

TABLE II. Clustering time in seconds using the centralized-based and distributed model in the Bank dataset

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(1)	344	636	917	1211	1503
(4)	5.86	10.78	15.8	21.04	26.4
(8)	3.75	6.47	9.26	11.95	15.32
(16)	2.53	4.19	5.86	7.36	11.19
(32)	1.84	2.77	3.83	4.76	6.00
(64)	1.44	1.89	2.52	3.05	3.71

TABLE III. Clustering time in seconds using the distributed model with the PCA method in the Bank dataset

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(4)	4.98	9.16	13.43	17.88	22.44
(8)	3.18	5.50	7.87	10.15	13.02
(16)	2.15	3.56	4.98	6.26	9.51
(32)	1.56	2.35	3.26	4.04	5.10
(64)	1.23	1.61	2.14	2.60	3.16

model. The results in Table IV show that increasing the number of used CPUs can decrease the clustering time.

It is considered that there are significant differences in execution time between our proposed distributed method and the centralized model with only one CPU. For instance, the distributed model within 64 VMs can analyze 42661 data points in 10.97 seconds whereas, in the centralized model, a single CPU takes 6083 seconds to perform the clustering process. According to the results in Table IV, increasing the number of CPUs can decrease the clustering time up to 90% using 64 CPUs instead of 4 CPUs. Furthermore, as the data size is growing, the clustering time can be reduced more and more using the distributed model. As it is clear, the suggested distributed model outperforms the centralized model and provides a better solution for data clustering taking the benefits of cloud computing resources.

For the next part of the experiment, the PCA method has been added to the distributed model. Table V presents the results of using the PCA method on the distributed model in the Marital dataset. The clustering time can be reduced using the PCA method by up to 18% more reduction than the provided results in Table IV.

C. Clustering Results for the Flight Dataset

The Flight dataset is used as the third dataset to evaluate the defined framework. Different portions of data have been used to contain 21500, 43000, 64500, 86000, and 107316 data points from the Flight dataset with different numbers of CPUs to calculate the clustering time. Table VI shows the clustering time for varying sizes of data and the numbers of used CPUs to fulfill data clustering. It shows how changes in different numbers of cloud resources affect the execution time of the clustering process and explores the advantages of using the distributed model.

There are significant differences in performance between our proposed distributed model and the centralized model. For example, the distributed model within 64 CPUs or VMs can analyze 107316 data points in 25.51 seconds only whereas in the centralized model, a single CPU takes 15053 seconds to complete the clustering process. According to the results in Table VI, increasing the number of CPUs can decrease the clustering time up to 91%, hence more resources work on data simultaneously.

In the next part of the experiment, the PCA method has been added to the distributed model as a booster to find out its effect on the selected dataset. Table VII shows the results of using the PCA method on the Flight dataset with the same settings in the previous experiments. The results show that the clustering time can be reduced by up to 18% more compared with the results in Table VI.

8. Discussion

The proposed framework has been secured using a double encryption technique that uses ECC and FHE encryption methods. What distinguishes the framework is that any authenticated data user can execute data analytics compu-

TABLE IV. Clustering time in seconds using the centralized-based and distributed model in the Marital dataset

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(1)	1208	2385	3638	4822	6083
(4)	22.37	43.65	64.87	85.90	107.90
(8)	13.14	25.08	36.87	48.64	61.00
(16)	8.00	14.85	21.58	28.33	35.42
(32)	5.06	8.90	12.70	16.46	20.49
(64)	3.10	5.11	7.23	8.90	10.97

TABLE V. Clustering time in seconds using the distributed model with the PCA method in the Marital dataset

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(4)	19.01	36.67	53.84	70.87	88.48
(8)	11.17	21.07	30.60	40.13	50.02
(16)	6.80	12.47	17.91	23.37	29.05
(32)	4.30	7.48	10.54	13.58	16.80
(64)	2.63	4.29	6.10	7.34	9.20

TABLE VI. Clustering time in seconds using the centralized-based and distributed model in the Flight dataset

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(1)	3138	6105	9067	12018	15053
(4)	54.90	108.70	162.40	216.40	269.10
(8)	31.31	61.74	91.54	121.40	151.22
(16)	18.42	36.00	52.89	70.02	86.98
(32)	10.87	20.71	30.29	39.89	49.51
(64)	6.08	10.99	15.84	20.66	25.51

TABLE VII. Clustering time in seconds using the distributed model with the PCA method in the Flight dataset

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(4)	46.67	91.31	134.79	178.53	220.66
(8)	26.61	51.86	75.98	100.16	124.00
(16)	15.65	30.24	43.90	57.77	71.32
(32)	9.24	17.40	25.14	32.91	40.60
(64)	5.17	9.23	13.15	17.04	20.91

tations on the ciphertext without any decryptions. The data analytic computations such as machine learning algorithms can be easily executed on the proposed framework.

Hence a hybrid encryption system has been used in the proposed framework, it is difficult to generate encrypted data that can be understandable or processable by data analytic computations. One of the challenges comes out when using FHE as the implementation of FHE is applicable only with integer values or integer domains [1]. To avoid the issue, only integer values have been selected from the used datasets.

The experiments have shown that increasing the number of used CPUs can significantly decrease the clustering time of data. Moreover, more data to be processed, therefore more clustering time is needed as there is more data to be processed. Besides, when increasing the data size, there is more reduction in the clustering time using the distributed model. Figure 3 illustrates how the clustering time is affected by different numbers of CPUs for each dataset while processing whole data. To check the effect of using cloud resources on the clustering time of data, different numbers of CPUs have been used. As it is clear in Figure 3, adding more resources can decrease the processing time of data. Therefore, scalable cloud resources should be used to reduce the overhead of using big data.

Figure 4 shows the percentage of decrement for the clustering time in the distributed model. As it is shown in Figure 4, the clustering time for the Bank dataset was decreased by up to 86%, in the Marital dataset, by up to 90%, and in the Flight dataset by up to 91% using 64 CPUs instead of 4 CPUs. Therefore, in the case of big data, as

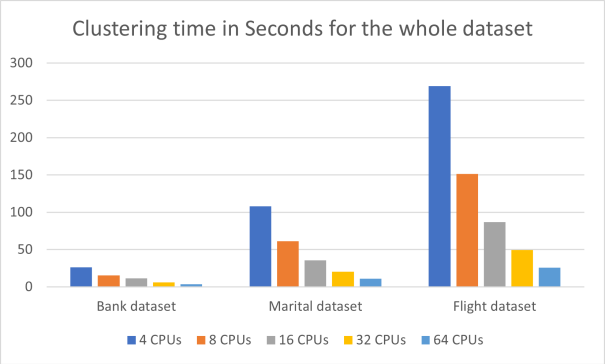

Figure 3. The effect of using the distributed model in each dataset

much as there is more data, there might be more percentage of decrement in clustering time using the distributed model. Furthermore, the PCA method was added to the proposed distributed model to check its effect on the selected datasets. As PCA works on the most relevant features or columns, if there are more features to be processed, then there might be more reduction in the clustering time using the PCA method. In the conducted experiments, the PCA method showed up 18% more reduction in the clustering time using the distributed model.

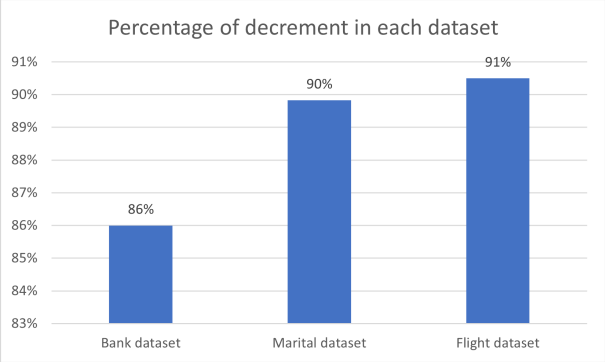

Figure 4. The percentage of decrement in the clustering time of each dataset using 64 CPUs instead of 4 CPUs

The scalability of the proposed solution can be discussed in three points. They are clustering time, the number of resources, and the cost of money. The proposed framework is scalable for any size of data even with millions of points and hundreds of resources. Obviously, in a distributed environment, a greater number of used CPUs provides less clustering time. However, in real-world applications, the number of used resources is related to a physical cost which in turn, may limit the number of resources that can be used and therefore limit minimum clustering time. It is notable that in some cases when more resources are added, there is no further reduction in clustering time. So, it is better to stop adding more resources to avoid adding more costs using the cloud. On the other hand, for the centralized model, only one CPU is needed to accomplish the processing but there is a huge gap in clustering time which can be significantly considered. The centralized model needs more time to process data and then more cost will be added by the cloud. The difference is very big in clustering time between the distributed model and the centralized model. For instance, in the Flight dataset, the centralized model executed the clustering process in more than 4 hours while with the same configurations, the distributed model could accomplish it in only 25 seconds which is a huge difference in results. As per cloud policy, more time to use the cloud more cost will be added.

As explained before, there is a significant gap between the results provided by the centralized model and the distributed model. Table VIII shows the clustering time in seconds provided by comparing both models for the three selected datasets. Additionally, the reduced clustering time using the PCA method has been shown in the table. In the Bank dataset, to process the whole dataset, 1503 seconds were required whereas using the distributed model, only 3.71 seconds were required to accomplish the clustering process. Furthermore, using the PCA method, the clustering time was reduced to 3.16 seconds as the minimum clustering timing using 64 CPUs to process the whole dataset. Likewise, the centralized model required 6083 seconds to process the whole Marital dataset using 64 CPUs whereas in the distributed model only 10.97 seconds were required to do the clustering job. Also, adding the PCA method could reduce the clustering time by up to 9.2 seconds. As well, in the Flight dataset, the elapsed clustering time using the centralized model was 15053 seconds compared with the distributed model which took only 25.51 seconds to finish the clustering process on the whole dataset. Additionally, the PCA method could reduce the timing up to 20.91 seconds which is the minimum clustering time using 64 CPUs. The developed distributed model has powerfully reduced the clustering time needed to accomplish the data clustering on the whole dataset.

There is no discussion that time is an important factor in any data analytics or computations, especially nowadays. However, other factors should be considered, such as cost and acceptable reduction in accuracy as all of them can be counted as the most important factors to define performance. Cost can be a limiting factor in defining the clustering time as it can be defined by the number of used resources in the cloud. Additionally, there is more possibility of reducing the clustering accuracy when using more clusters of nodes to accomplish the clustering job. As all clusters are working in parallel and at the same time, there is a merge process at the end to merge all the results provided by the nodes. With more nodes to be merged, there is more possibility of a reduction in the accuracy of clustering results. Therefore, there is a trade-off in choosing the best combination of cost, number of resources, time, and acceptable errors or desired clustering accuracy.

When the performance of the proposed framework is

TABLE VIII. Clustering time in seconds needed by different models on the whole dataset using 64 CPUs

Name of the Dataset	Centralized Model	Distributed Model	Distributed with Adding PCA
Bank	1503	3.71	3.16
Marital	6083	10.97	9.20
Flight	15053	25.51	20.91

compared with other researchers, it can be found that the work in [15] also used a secure framework with clustering but with one layer of security. In the mentioned study, the framework decreased the clustering time by up to 81% using a similar dataset and settings, while our proposed framework succeeded in further reduction and decreased the clustering time by up to 86% in the Bank dataset with two layers of security. The defined framework can be a novel approach toward efficient and secure data analytics in the cloud. The suggested solution integrates a hybrid encryption method with a distributed computation environment to enhance the data clustering process, taking benefit of cloud resources.

9. Conclusion and Future works

This study focused on securing big data analytics in the cloud by employing a hybrid encryption method using ECC and FHE. The data always is protected in any case while it is on rest, storage, or even in processing. Besides, FHE makes data to be compatible with any data analytics on the cloud. Additionally, a distributed model was used to reduce the clustering time. The encrypted data is distributed over many VMs or CPUs to accomplish the clustering job simultaneously. To approve the outperformance of the suggested distributed model, the provided results in the distributed model were compared with the results in a centralized model. Moreover, the PCA method was used as an extra proposed solution to decrease the clustering time more and more. The results showed that data analytics improved significantly by up to 91% using 64 CPUs instead of 4 CPUs, and the clustering time was reduced by up to 18% more reduction by adding the PCA method to the distributed model. The proposed framework can be a novel solution for data security in cloud-based big data analytics.

For future works, huge and numerous datasets will be considered to explore their effect on the defined framework. Additionally, different clustering algorithms can be applied to the developed framework to compare the clustering time and accuracy for the used clustering methods. Different data analytics methods rather than the used ones can be used to find out the efficiency of the suggested framework using different approaches.

References

[1] A. Alabdulatif, H. Kumarage, I. Khalil, and X. Yi, "Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption," *Journal of Computer and System Sciences*, vol. 90, pp. 28–45, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000017300284

[2] A. V. Lucca, G. M. Sborz, V. Leithardt, M. Beko, C. A. Zeferino, and W. Parreira, "A review of techniques for implementing elliptic curve point multiplication on hardware," *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, p. 3, Dec. 2020. [Online]. Available: https://doi.org/10.3390/jsan10010003

[3] Z. Salman and W. M. Elmedany, "A trustworthy cloud environment using homomorphic encryption: a review," pp. 31–36, 2021.

[4] N. Almutairi, F. Coenen, and K. Dures, "K-means clustering using homomorphic encryption and an updatable distance matrix: Secure third party data clustering with limited data owner interaction," pp. 274–285, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-64283-3_20

[5] R. S. M. L. Patibandla and N. Veeranjaneyulu, "Survey on clustering algorithms for unstructured data," pp. 421–429, 2018. [Online]. Available: https://doi.org/10.1007/978-981-10-7566-7_41

[6] K. Abouelmehdi, A. Beni-Hssane, H. Khaloufi, and M. Saadi, "Big data security and privacy in healthcare: A review," *Procedia Computer Science*, vol. 113, pp. 73–80, 2017. [Online]. Available: https://doi.org/10.1016/j.procs.2017.08.292

[7] N. K. Anuar, A. A. Bakar, and A. A. Bakar, "A review on privacy-preserving techniques in data analytics," Oct. 2021. [Online]. Available: https://doi.org/10.1109/icsip52628.2021.9688624

[8] L. E. Haourani, A. A. E. Kalam, and A. A. Ouahman, "Big data security and privacy techniques," Mar. 2020. [Online]. Available: https://doi.org/10.1145/3386723.3387841

[9] Z. Salman and M. Hammad, "Securing cloud computing: A review," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 545–554, Apr. 2021. [Online]. Available: https://doi.org/10.12785/ijcds/100152

[10] W. Fang, X. Z. Wen, Y. Zheng, and M. Zhou, "A survey of big data security and privacy preserving," *IETE Technical Review*, vol. 34, no. 5, pp. 544–560, Sep. 2016. [Online]. Available: https://doi.org/10.1080/02564602.2016.1215269

[11] S. H. Begum and F. Nausheen, "A comparative analysis of differential privacy vs other privacy mechanisms for big data," Jan. 2018. [Online]. Available: https://doi.org/10.1109/icisc.2018.8399125

[12] Z. Salman, M. Hammad, and A. Y. Al-Omary, "A homomorphic cloud framework for big data analytics based on elliptic curve cryptography," Sep. 2021. [Online]. Available: https://doi.org/10.1109/3ict53449.2021.9582001

[13] M.-Q. Hong, P.-Y. Wang, and W.-B. Zhao, "Homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud computing," Apr. 2016. [Online]. Available: https://doi.org/10.1109/bigdatasecurity-hpsc-ids.2016.51

[14] C. Aguilar-Melchor, J.-C. Deneuville, P. Gaborit, T. Lepoint,

and T. Ricosset, "Delegating elliptic-curve operations with homomorphic encryption," May 2018. [Online]. Available: https://doi.org/10.1109/cns.2018.8433140

[15] A. Alabdulatif, I. Khalil, and X. Yi, "Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 192–204, Mar. 2020. [Online]. Available: https://doi.org/10.1016/j.jpdc.2019.10.008

[16] F. O. Catak, I. Aydin, O. Elezaj, and S. Yildirim-Yayilgan, "Practical implementation of privacy preserving clustering methods using a partially homomorphic encryption algorithm," *Electronics*, vol. 9, no. 2, p. 229, Jan. 2020. [Online]. Available: https://doi.org/10.3390/electronics9020229

[17] S. Khanmohammadi, N. Adibeig, and S. Shanehbandy, "An improved overlapping k-means clustering method for medical applications," *Expert Systems with Applications*, vol. 67, pp. 12–18, Jan. 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2016.09.025

[18] R. Alzu'bi, A. Anushya, E. Hamed, B. A. Vincy, and A. AlSha'ar, "Medical image segmentation via optimized k-means," Sep. 2017. [Online]. Available: https://doi.org/10.1109/ctceec.2017.8455030

[19] S. PRABAKERAN, K. R. HEMANTH, T. ARVIND, N. BHARATH, , , and and, "HYBRID CRYPTOSYSTEM USING HOMOMORPHIC ENCRYPTION AND ELLIPTIC CURVE CRYPTOGRAPHY ALGORITHM," *i-manager's Journal on Computer Science*, vol. 7, no. 1, p. 1, 2019. [Online]. Available: https://doi.org/10.26634/jcom.7.1.15667

[20] G. P. Kanna and V. Vasudevan, "A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data," *Cluster Computing*, vol. 22, no. S4, pp. 9561–9569, Apr. 2018. [Online]. Available: https://doi.org/10.1007/s10586-018-2723-9

[21] "Kaggle: Your machine learning and data science community." [Online]. Available: https://www.kaggle.com/

[22] Amazon, "Amazon ec2." [Online]. Available: https://aws.amazon.com/ec2/?nc2=type_a

[23] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.2980942

Zainab Salman is currently a Ph.D. candidate at the University of Bahrain and studying in Computing and Information Sciences program. She received her Master's degree in Computer Science from Al Ahlia University, Bahrain in 2010 and her B.Sc. in Computer Science from the University of Bahrain in 2005. Her research interests include cloud computing, security, and big data analytics.

Alauddin Alomary is an Associate Professor of Computer Engineering, College of Information technology, University of Bahrainresearch Interest: Hardware/Software co-designTelematic system, Machine-to-Machine Communication, Mobile Network performance, ASIC and embedded system design using VHDL and FPGA.

Mustafa Hammad is an Associate Professor in the Mutah University, Jordan. He received his Ph.D. in Computer Science from New Mexico State University, USA in 2010. He received his Master's degree in Computer Science from Al-Balqa Applied University, Jordan in 2005, and his B.Sc. in Computer Science from The Hashemite University, Jordan in 2002. His research interests include machine learning, and software engineering with a focus on software analysis and evolution.