NTU Journal of Engineering and Technology (2024) 3 (1) : 35-44
DOI: https://doi.org/10.56286/ntujet.v3i1.861

P-ISSN: 2788-9971 E-ISSN: 2788-998X

NTU Journal of Engineering and Technology

Available online at: https://journals.ntu.edu.iq/index.php/NTU-JET/index

Preserving Big Data Privacy in Cloud Environments Based on Homomorphic Encryption and Distributed Clustering

Shatha A. Baker
Electronic Technologies Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq

Article Informations

Received: 20-01- 2024,
Revised: 22-02-2024,
Accepted: 02-03-2024,
Published online: 11-03-2024

Corresponding author:
Name: Shatha A. Baker
Affiliation Northern Technical University
Email: shathaab@ntu.edu.iq

Key Words:
Big data,
Cloud computing,
Homomorphic encryption,
Privacy,
Extremely Distributed Clustering.

ABSTRACT

Cloud computing has grown in popularity in recent years because to its efficiency, flexibility, scalability, and the services it provides for data storage and processing. Still, big businesses and organizations have severe concerns about protecting privacy and data security while processing these massive volumes of data.

This paper proposes approach that intends to enhance efficiency in delivering advanced data protection, hence filling security holes, by enhancing data protection from various big data sources. A partial homomorphic encryption system is used to encrypt data created by many sources or users and processed in the cloud without decrypting it, hence protecting data from attackers. Extremely Distributed Clustering (EDC) has also been applied to partition large datasets into many cloud computing node subsets. This method can ensure privacy and protect data while also enhancing the effectiveness and performance of big data analytics. According to the results, the proposed technique was faster and gave improved encryption performance by around 23-28%.

Introduction

One of the main obstacles for big data enabled apps is finding relevant information, and cloud computing helps businesses handle huge data more effectively. In order to enhance the quality assurance of application services, the cloud provided limitless resources for the storage, management, and analysis of large amounts of heterogeneous data [1]. However, there are several internal and external privacy breaches as well as leakage concerns that affect cloud computing. One of the main obstacles to moving data analytic services to the cloud is the privacy of sensitive data, which has raised serious worries about trust issues in cloud computing. As a result, it is critical to create secure data mining models that are able to utilize cloud resources is essential [2-3].

Homomorphic Encryption (HE) is a kind of current encryption method that allows us to do operations on data without having to first decrypt it[4]. After conducting calculations, the result is encrypted. There is no difference in the output if the calculating process is performed on decrypted data. The principal application of homomorphic encryption is to secure and maintain the privacy of data and information that is outsourced for storage and computing. The ability to process data without decryption enables commercially available cloud services to preserve a better level of data privacy[5].

The paper presents an approach for privacy preservation that encrypts cloud data using the HE technique The enhanced Partial Homomorphic Encryption is used as a basis for the suggested technique. In addition, Large-scale data is split up into subsets in various nodes using the EDC technique. The proposed approach protects big data's security and privacy, and the experimental evaluation demonstrates the efficiency of this approach and its ability to build a secure cloud application.

The paper is structured as follows: Sec. 2 offers a succinct overview of the related work. Sec. 3 offers a detailed discussion of the cloud computing. Sec.4 outlines homomorphic encryption's fundamental characteristics and literature. Sec. 5 describes the proposed approach. Sec.6 and 7 present the experimental results and conclusions, respectively.

Related works

This section examines the existing literature on big data privacy preservation, which is one of the primary areas connected to the approach this paper proposes.

In [4], A fully HE technique depended on ideal lattices that satisfies both multiplicative and additive homomorphisms was initially suggested by Craig Gentry in 2009. fully HE has been widely used because of its exceptionally strong security. Homomorphic encryption has greatly improved privacy protection, particularly in cloud computing [5]. A method to protect smart meter data privacy in a smart grid was presented in the work in [6]. Before storing the smart data in the cloud, they encrypted it using homomorphic encryption.

A comparable method has also been suggested in[7] for cloud-based medical image security utilizing HE. According to the authors in [8], third-party services are thought to be flawed and have a significant risk of security lapses while keeping data on the cloud. When data is transferred outside of administrative control, there may be a higher risk of attack. In [9], they explores privacy-preserving techniques in Big Data analytics using HE algorithms, evaluates their security, and compares performance and features of new toolkits. In [10], they successfully used a completely homomorphic encryption technique in mobile device contexts and created and built a Raspberry-based framework for comparable video recognition.

The authors in [11] proposes a privacy-preserving deep learning model (PPDLM) using HE and least-squares method for data protection. PPDLM significantly enhances data privacy protection and has higher computational efficiency compared to NPPDLM. While Mustafa et al.[12] developed a blockchain-based model for healthcare security, evaluating its privacy preservation, security, and complexity based on traditional technologies and safety precautions in healthcare systems.

Cloud Computing

It is a substantial collection of scalable, virtualized computing resources that can serve a broad range of applications and provide different services that clients need. According to its "pay only for use" philosophy, users are responsible for covering the resources' costs. Computing clouds are just a group of linked networks that offer high-quality, reasonably priced, scalable services that are available to anybody, anywhere, at any time, and for both individual and group use. Servers, services, computer networks, storage databases, and applications make up the architecture of cloud computing in general. To access the necessary services, such as using the processors' available computing power, keeping any amount of information and data, or exchange data with specific recipients, any user outside the cloud can connect to the computer cloud. It goes without saying that since cloud computing is used as needed, its cost will vary[14].

- ## Computer Cloud deployments

There are four deployable computer clouds models that are available and that can be used, namely[15]:

1. **Private**: this cloud is only connected to other private branches over the internet, but it is only set up, monitored, and maintained for a certain geographic area.

2. **Public:** Users can access such clouds through services like Google Drive and Microsoft One Drive. It benefits the general public because it costs less than building one's own facilities would.

3. **Hybrid:** It is possible to connect private and public clouds, and can access them, to exchange and provide services to the two connected parties.

4. **Community**: In this cloud, a sizable infrastructure is utilized. Government agencies that provide computer and data storage services to the computing community are included but not exclusively.

As seen in table 1 and described below, cloud computing implementation can be broadly divided into three levels or services[14] :

Table1: Model for deploying cloud services.

Cloud	Administration of infrastructure	Infrastructure Location	Infrastructure ownership	Consumption and access
Private/ community	Third-party service provider or organization	On-premises or off-premises	Third-party service provider or organization	Trusted
Public	Third-party service provider	Off-premises	Third-party service provider	Untrusted
Hybrid	Both Third-party service provider and organization	Both On-premises and off-premises	Both Third-party service provider and organization	Trusted and untrusted

- ## Cloud Computing Security

Security and privacy issues are the most significant impediments to mainstream adoption of cloud concepts, as employing a commercial public cloud reduces direct supervision of systems which maintain dependable applications and data. There are numerous cryptographic processes used to maintain information privacy in order to safeguard large data examinations in the cloud, like HE. Table 2 describes the primary features of data security in cloud computing, along with potential risks and defense solutions[15].

Table 2: The primary aspects of cloud computing data security

Security aspect	Threats	Strategies for defense
Confidentiality	− Cross VM assault	− Preventing Placement
	− a malicious system administrator	− Detection of Co-Residency
		− NoHype
		− Platform for dependable cloud computing
		− giving the customer back ownership of their data
Privacy	-Cloud confidentiality threats are the same	− Information-centered security
		− Computing that can be trusted
		− Homomorphic Encryption
Integrity	− Loss of data / manipulation	− Possession of Provable Data
	− Untrustworthy computation on remote servers	− Auditor from a Third Party
		− Preventing fraudulent computing
Availability	− Use low bandwidth for flooding attacks	− defending the new DOS attack
	− attack of Fraudulent Resource Consumption (FRC)	− detection of FRC attack

Homomorphic Encryption (HE)

One of the asymmetric encryption methods that enables calculations to be performed on encrypted data without exposing the underlying plaintext is HE. In order to protect user data and privacy, HE are currently being widely used with a variety of applications, particularly in the medical, financial, and industrial industries. Indeed, one of the most efficient ways to secure and handle data across

remote servers, including the cloud, is HE [16].
Figure 1 show the HE function.

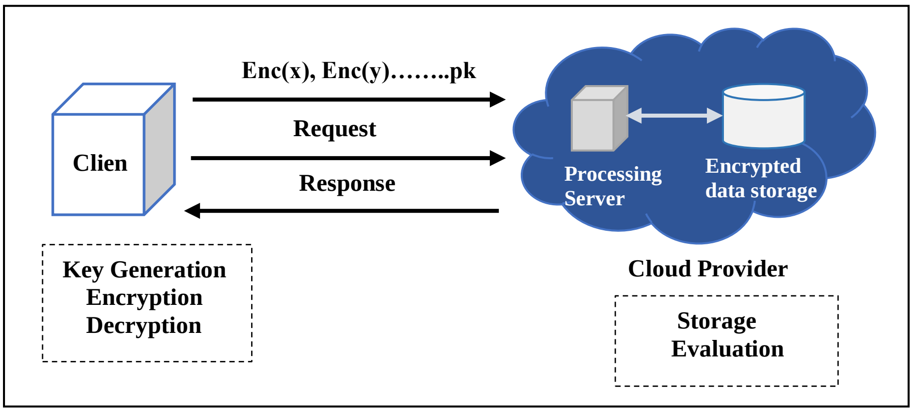

Figure 1: Functions of HE

• Homomorphic Encryption Classification

1. Multiplicative HE: It is considered to be multiplicatively HE when the only allowed operation on the encrypted data is multiplication[17], such as RSA. A HE is a multiplicative, if:

$Enc(x \otimes y) = Enc(x) \otimes Enc(y)$ ..(1)

$Enc\ (\prod_{i=1}^{1} m_i) = \prod_{i=1}^{1} Enc\ (m_i)$ …(2)

2.Additive HE: It is considered to be additive HE when the only allowed operation on the encrypted data is addition, such as Paillier [17]. HE is classified as an additive if:

$Enc(x \oplus y) = Enc(x) \otimes Enc(y)$ …(3)

$Enc(\sum_{i=1}^{1} m_i) = \prod_{i=1}^{1} \mathrm{Enc}(\mathrm{m}_i)$ ..(4)

• Types of Homomorphic Encryption

HE is divided into two categories according on the number of mathematical operations that may be done [18-19].

1. Partial Homomorphic Encryption (PHE)

A PHE can apply additive or multiplicative homomorphism but not both simultaneously. Examples include RSA, ElGamal, and Paillier. PHE is more efficient than fully and somewhat homomorphic ones, and have been used in healthcare, applications of deep learning and systems of intelligent transportation. Figure 2 depicts the progression of PHE [9].

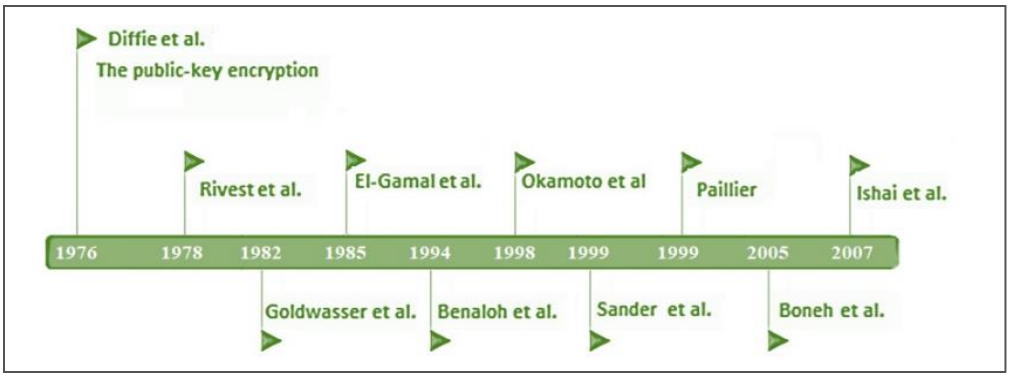

Figure 2. PHE Timeline

2. Fully Homomorphic Encryption (FHE)

FHE is a class of encryption algorithms that includes additive and multiplicative homomorphism operations. Based on an intricate network of perfect lattices for keys and ciphertext, Craig Gentry created it for the first time in 2009.

Despite initial challenges, researchers and industrial companies have continued to work on developing effective and efficient FH algorithms to overcome the problems of lattice-based encryption. Figure 3 depicts the progression of FHE [20].

The distinction between FHE and PHE is seen in Table 3[21].

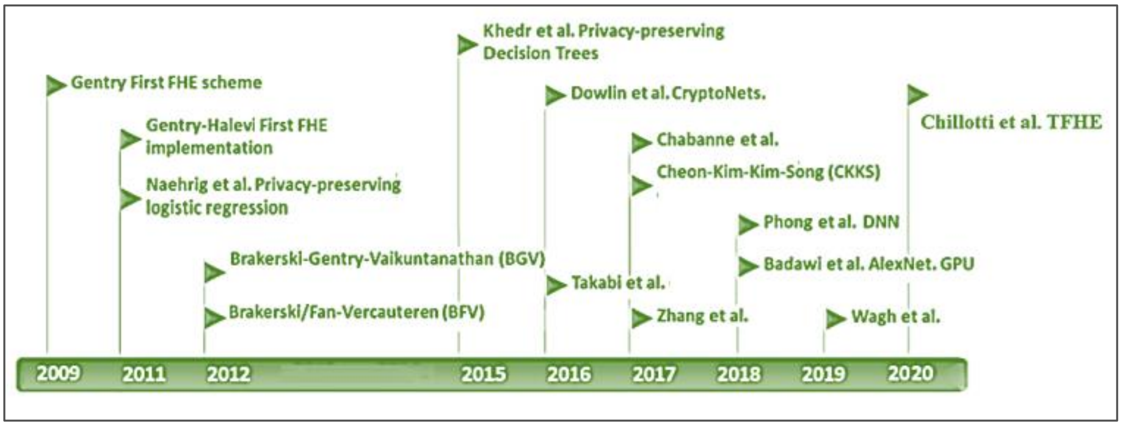
Gentry First FHE scheme
Gentry-Halevi First FHE implementation
Naehrig et al. Privacy-preserving logistic regression
Brakerski-Gentry-Vaikuntanathan (BGV)
Brakerski/Fan-Vercauteren (BFV)
Khedr et al. Privacy-preserving Decision Trees
Dowlin et al. CryptoNets.
Takabi et al.
Chabanne et al.
Cheon-Kim-Kim-Song (CKKS)
Zhang et al.
Phong et al. DNN
Badawi et al. AlexNet. GPU
Wagh et al.
Chillotti et al. TFHE
2009
2011
2012
2015
2016
2017
2018
2019
2020

Figure 3: FHE Timeline

Table 3: The difference between FHE and PHE

Parameter	**FHE**	**PHE**
Type of operation	Both	Either multiplication or addition
Ciphertext size	Large	Small
Computation	Unlimited	computations number is Limited
Versatility	High	Low
Computational efforts	Requires more effort	lesser effort is required
Performance	Slower	Faster and more compact
Examples	Gentry Scheme	ElGamal, RSA

- ## Homomorphic Encryption Security

A HE technique will give high levels of data security to applications that use it, and standardization will improve its popularity and use. A consensus on the level of security settings in different implementations and systems will be a critical component of this standardization process. Many literature state that a HE scheme has three security features [9]:

1. No opponent can tell whether both distinct messages were encrypted using the same ciphertext. Here, the encryption is randomized to ensure that the same message cannot have the same encryption, which assures semantic security..
2. Compactness: HE acts on ciphertext without extending its length.
3. Ciphertexts are decrypted and operated on efficiently. The running time of retrieving the plaintext should be unaffected by the functions produced from the ciphertext. As a consequence, the ciphertext's functions are unaffected by the decryption process.

Proposed Approach

The primary procedures of the suggested framework have been depicted in Figure 4. The improved HE algorithm processes the client data first. Subsequently, the encrypted data is transferred to the cloud and grouped using the EDC methodology. The EDC method divides data into various clusters so that they can be analyzed concurrently and independently. Any user can currently compute data on the clusters, including the cloud provider. Requests for the encrypted data to be decrypted can only be fulfilled by the verified client. The data cannot be decrypted unless the client submits a request to the cloud server. The HE is used to transfer the data through for decryption. The improved HE algorithm processes the encrypted data at this stage. The processes are explained in the ensuing subsections.

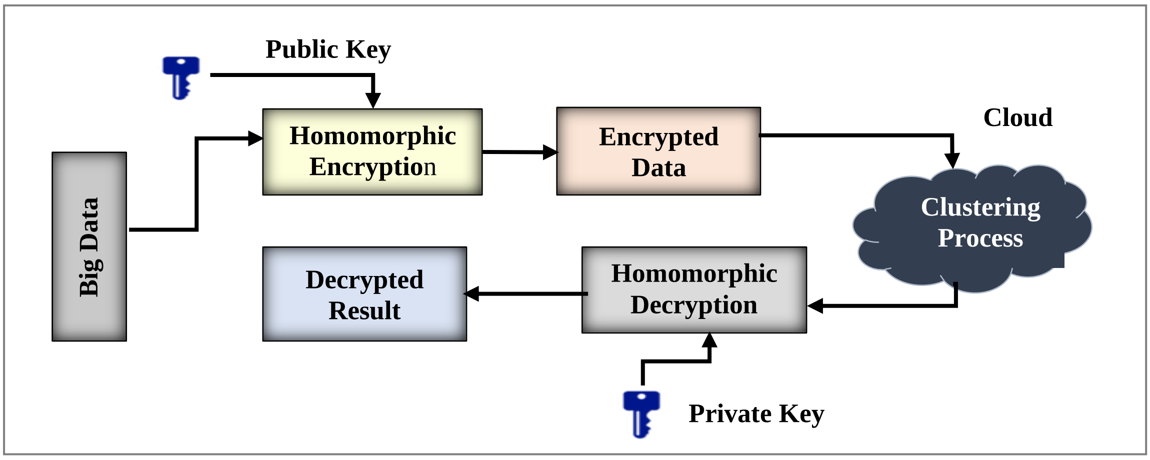

Figure 4. The proposed approach

• Encryption Method

Paillier encryption is a PHE applying additive homomorphism based on asymmetric and public-key cryptography. It consists of three sections:

Key generation:

1. Create two large prime numbers p and q, where gcd (p q (p - 1) (q- 1)) =1

2. λ = lcm (p - 1 , q - 1), and modulus n = p q.

3. $Z_n = \{ x \mid x \in Z , 0 \leq x < n \}$ is defined, $Z_n^* = \{ x \mid x \in Z , 0 \leq x < n \}$, gcd (x , n) =1)

4. Choose an integer $g \in Z^*_{n}{}^2$ such that it fulfills gcd (L (g^λ mod n^2), n) =1 ,

$L(x) = (x-1)/n$, $x \in \{ x < n^2 \mid x \equiv 1 \bmod n \}$

Private key is (λ) and public key is (n , g)

Encryption: Create a random number $r \in (0, n)$, then calculate

$$C = g^m .\ r^n \bmod n^2 \qquad ..(5)$$

Decryption: calculate

$$m = \frac{L(c^\lambda \bmod n^2)}{L(c^g \bmod n^2)} \bmod n \qquad ..(6)$$

However, the Paillier algorithm is difficult when it comes to encryption and decryption. Consequently, the paper propose a modified version of Paillier.

Key generation: if α is used as a divisor, it substitutes the location of λ in the private key and changes the order of g in the public key to make sure it is αn.

Encryption: Assume that m represents the plaintext, c represents the ciphertext, and r is represents the random positive integer, and that r is smaller than. The enhanced encryption procedure can be represented as follows:

$$c = g^m .(g^n)^r \bmod n^2 \qquad ...(7)$$

Decryption: The following represents the decryption process:

$$m = \frac{L(c^\alpha \bmod n^2)}{L(g^\alpha \bmod n^2)} \bmod n \qquad ...(8)$$

The decryption offers the main benefit of α using instead of using λ, as is observe from the aforementioned algorithm. From 2λ times to 2α times, there have been two power operations. As a result of being a α is a divisor of λ, the time overhead has been greatly decreased. The improved Paillier has a computational complexity $O(|n|^2 |\alpha|)$ compared to Paillier's $O(|n|^3)$.

• Clustering Process

Big data can be split up into various categories or clusters using clustering techniques. Smaller data requires less analysis time. As a result, This increases the effectiveness of processing big data. Similar data are grouped together based on how far from the centroid they are, and each cluster is represented by its centroid. With large-scale data processing, the EDC technique can be employed safely [16]. This technique uses many servers running as virtual machines to establish distinct data clusters. Through individually and simultaneous processing of each subset of data, the EDC approach facilitates a faster clustering process. When compared to the usual centralized clustering strategy, the EDC calculations can improve computation efficiency. The EDC method is one of the primary processes in the proposed method for partitioning huge data into smaller data clusters. It is divided into three stages, as shown in figure 5.

1.The initial clusters will be split up across many virtual machines.

2.Each virtual machine is processed independently.

3.The final clustering output is produced by applying renormalization (each data point's membership probability for a set of clusters should be adjusted) and reconciliation (Put every data point in the appropriate cluster) throughout the whole data set during the merging process.

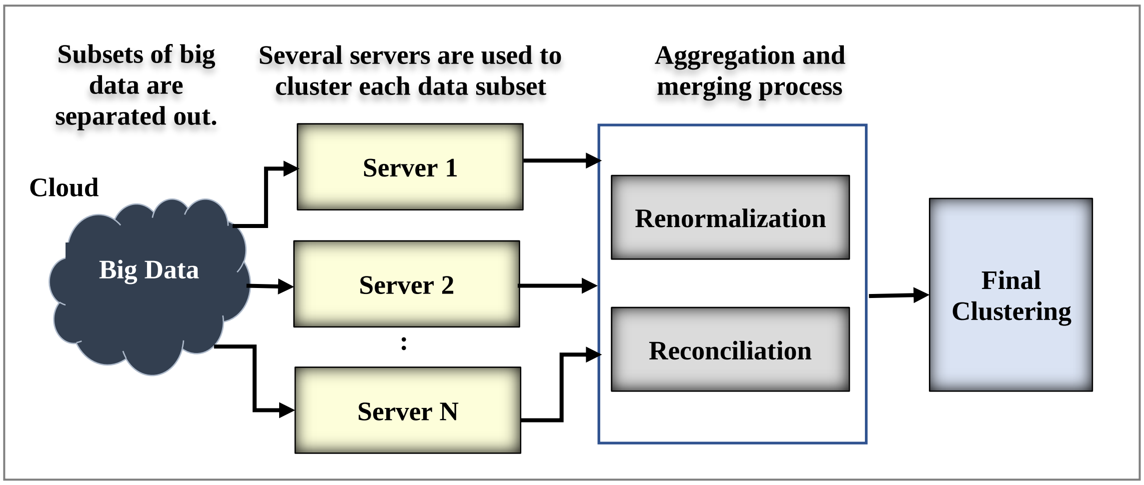
Subsets of big data are separated out.
Several servers are used to cluster each data subset
Aggregation and merging process
Cloud
Big Data
Server 1
Server 2
:
Server N
Renormalization
Reconciliation
Final Clustering

Figure 5: The EDC technique's general structure

Experiments and Results

Numerous practical experiments were conducted on HE methods, yielding the following outcomes, which are detailed in tables 4,5. The RSA method has acceptable efficiency when taking into account its features; yet, it has the disadvantage that its semantic security contradicts the homomorphism property. The RSA technique is therefore unable to offer both properties simultaneously. Semantic security is provided via the HE property (multiplicative property) of the ElGamal method. Its primary disadvantages, however, are that it requires randomness, moves more slowly, and encrypts messages that are twice as long as the plaintext due to the message's twofold expansion during encryption. The Paillier method is classified as probabilistic cryptography even though it has a high level of security due to its semantic security. It possesses the essential (additive) quality of symmetry. Its most noticeable key, though, has a lengthy encryption and decryption period. Compared to others, its ciphertext is longer. The results showed that the proposed technique was faster and provided better encryption performance by about 23-28% compared to the original method.

Tables 4 display the time taken to generate the key and its final size based on the mathematical problem used to generate the key, the random number selected throughout the procedure, and the necessary key length. The length of time it took to generate the key variable for each of the methods was caused by a mathematical issue with the encryption keys generated by each of the approaches. The key file for the Paillier algorithm is nearly the most important. But because more mathematical operations are needed to generate the keys for any given method, the higher the key size, the longer it takes to generate the keys, as shown in figure 6.

Table 4: Comparison between Proposed method and other methods that use constant file sizes and variable key sizes

Methods / Key Size	RSA	ElGamal	Paillier	Proposed method
256	0.0468	0.1249	0.0937	0.07287
512	0.7968	1.828	0.609	0.4705
1024	2.626	1.1405	3.5948	2.6961
2048	25.2807	8.3588	39.7197	29.2671

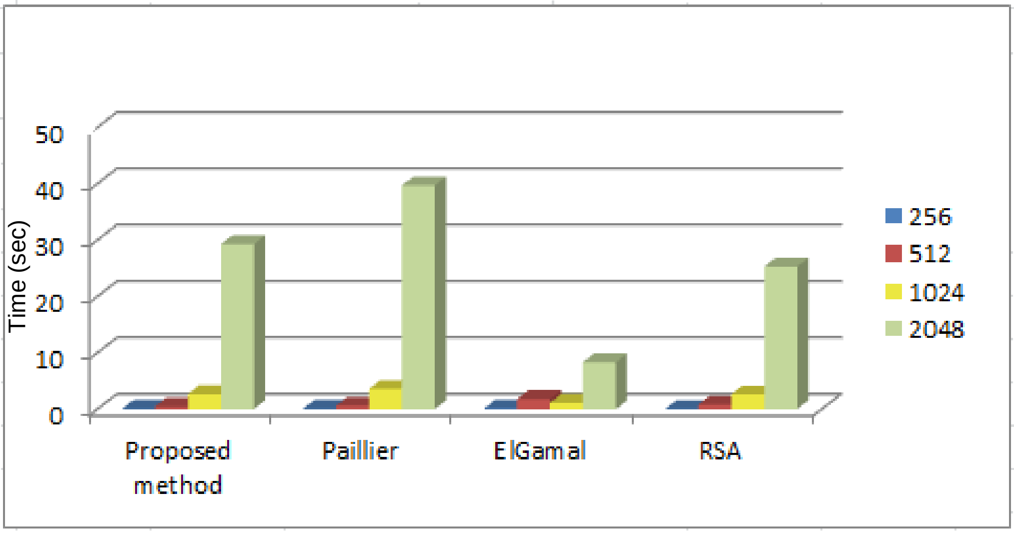

Figure 6. The time costs of encryption and decryption with constant file sizes and variable key sizes

Table 5 shows how the long execution time of the Paillier technique results from the calculations required during the procedure. While the proposed method was faster and provided better encryption performance compared to the original method, as shown in figure 7.

After the encrypted data is transferred to the cloud, EDC uses four servers, data sets containing 1,000 data points, and twelve central cluster cores to break down the huge data into smaller data. Hence the data size becomes The data is smaller, resulting in reduced processing time for huge data. In light of this, the proposed approach is a useful and effective way to process huge amounts of data while maintaining its security and privacy.

Table 5: Comparison between Proposed method and other methods that use constant key sizes and variable file sizes

Methods **Data**	**RSA**	**ElGamal**	**Paillier**	**Proposed methods**
Data 1	2.62347	1.124927	3.62234	2.817378
Data 2	2.66115	1.156168	3.69299	2.853674
Data 3	2.65559	1.140542	3.70042	2.775321
Data 4	2.623395	1.131446	3.76969	2.77766

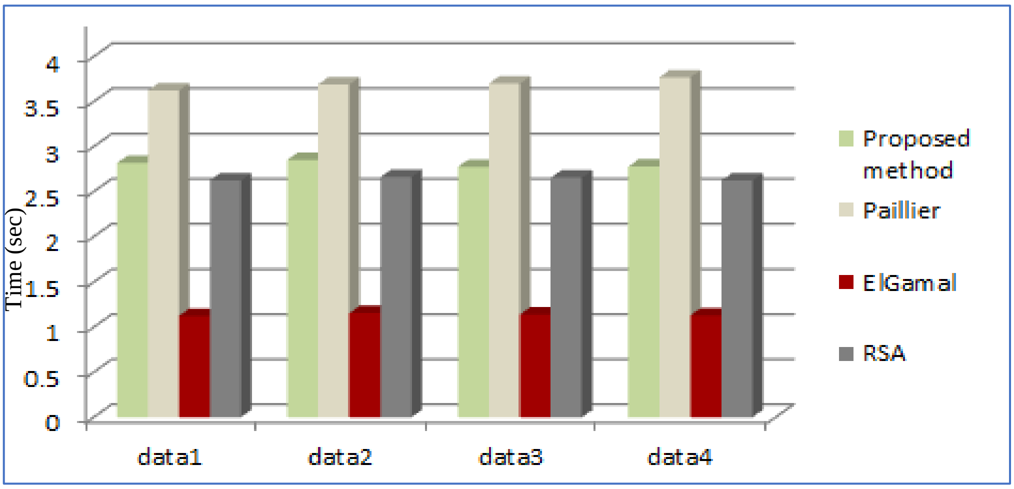

Figure 7. The time costs of encryption and decryption with constant key sizes and variable file sizes

Conclusion

The suggested approach presented in this paper addresses privacy and data security issues that arise while managing massive volumes of data in cloud computing using a powerful HE scheme based on improved PHE method which is more efficient and successful as the results showed that the proposed technique was faster and provided approximately 23-28% better encryption performance compared to the original method. Then, using a process known as EDC, the big data is broken down into smaller units after being encrypted and transferred to the cloud. Because of this phase, the data became smaller, which reduced the amount of time needed to calculate big data while maintaining the security and privacy of the data.

References

[1] Naeem, N., Khan, F., Yaqoob, T., & Tahir, S. (2023). Privacy-Preserving Computing via Homomorphic Encryption: Performance, Security, and Application Analysis. Handbook of Research on Cybersecurity Issues and Challenges for Business and FinTech Applications, 288-313.

[2] Bouleghlimat, I., Boudouda, S., & Hacini, S. (2023). PPSecS: Privacy-Preserving Secure Big Data Storage in a Cloud Environment. Arabian Journal for Science and Engineering, 1-15.

[3] Baker, S. A., Nori, A. S. "Internet of things security: a survey"(2021). Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8-9, 2020, Revised Selected Papers 2. Springer Singapore.

[4] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

[5] Çatak, F. Ö., & Mustacoglu, A. F. (2018). CPP-ELM: cryptographically privacy-preserving extreme learning machine for cloud systems. International Journal of Computational Intelligence Systems, 11(1), 33.

[6] Mai, V., & Khalil, I. (2017). Design and implementation of a secure cloud-based billing model for smart meters as an Internet of things using homomorphic cryptography. Future Generation Computer Systems, 72, 327-338.

[7] Vengadapurvaja, A. M., Nisha, G., Aarthy, R., & Sasikaladevi, N. (2017). An efficient homomorphic medical image encryption algorithm for cloud storage security. Procedia computer science, 115, 643-650.

[8] Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. Information sciences, 305, 357-383.

[9] Hamza, R., Hassan, A., Ali, A., Bashir, M. B., Alqhtani, S. M., Tawfeeg, T. M., & Yousif, A. (2022). Towards secure big data analysis via fully homomorphic encryption algorithms. Entropy, 24(4), 519.

[10] Lagesse, B., Nguyen, G., Goswami, U., & Wu, K. (2021, March). You Had to Be There: Private Video Sharing for Mobile Phones using Fully Homomorphic Encryption. In 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (pp. 730-735). IEEE.

[11] Fan, Y., Zhang, W., Bai, J., Lei, X., & Li, K. (2023). Privacy-preserving deep learning on big data in cloud. China Communications.

[12] Ali, A., Al-Rimy, B. A. S., Alsubaei, F. S., Almazroi, A. A., & Almazroi, A. A. (2023). HealthLock: Blockchain-Based Privacy Preservation Using Homomorphic Encryption in Internet of Things Healthcare Applications. Sensors, 23(15), 6762.

[13] Mahmood, A. I., Alsaif, O., & Saleh, I. A. (2022). Routing flying Ad Hoc network using salp swarm algorithm. Indonesian Journal of

Electrical Engineering and Computer Science, 28(2), 946-953.

[14] Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 34(1), 1-11.

[15] Mohammed, S. J., & Taha, D. B. (2021). From cloud computing security towards homomorphic encryption: A comprehensive review. TELKOMNIKA (Telecommunication Computing Electronics and Control), 19(4), 1152-1161.

[16] El Makkaoui, K., Ezzati, A., & Hssane, A. B. (2015, June). Challenges of using homomorphic encryption to secure cloud computing. In 2015 International Conference on Cloud Technologies and Applications (CloudTech) (pp. 1-7). IEEE.

[17] EZZATI, A., El Makkaoui, K., & Hssane, A. B. (2015). Homomorphic Encryption as a Solution of Trust Issues in Cloud. In International Conference on Big Data, Cloud and Applications.

[18]Kartit, Ali. "New Approach Based on Homomorphic Encryption to Secure Medical Images in Cloud Computing." Trends in Sciences 19.9 (2022): 3970-3970.

[19] Mohamed, S. A., Alsaif, O. I., & Saleh, I. A. (2022). Intrusion Detection Network Attacks Based on Whale Optimization Algorithm. Ingénierie des Systèmes d'Information, 27(3).

[20] Pulido-Gaytan, B., Tchernykh, A., Cortés-Mendoza, J. M., Babenko, M., Radchenko, G., Avetisyan, A., & Drozdov, A. Y. (2021). Privacy-preserving neural networks with homomorphic encryption: C hallenges and opportunities. Peer-to-Peer Networking and Applications, 14(3), 1666-1691.

[21] Seth, B., Dalal, S., & Kumar, R. (2019). Hybrid homomorphic encryption scheme for secure cloud data storage. Recent Advances in Computational Intelligence, 71-92.