International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X; pISSN:2321-2241 Volume: 15; Issue: 1; March -2026

Secure Data Sharing Using Homomorphic Encryption

Raghavendra N, Prof. Gunashekaran K

Student, Department of MCA, AMC Engineering College, Bengaluru India

Assistant Professor, Department of MCA, AMC Engineering College, Bengaluru India

ABSTRACT:

With the rise of the digital era, cloud computing and outsourcing of data has developed such a worrying concern regarding the safety and privacy of data. Companies tend to store and manipulate sensitive information such as financial data, medical data and personal data in third-party cloud storage systems. Though conventional methods of encryption guard data both in storage and transmission, they need decryption prior to processing, exposing sensitive data to the risk of security attacks. This vulnerability offers the need to seek more advanced cryptography solutions. Homomorphic Encryption (HE) is a recent technique that allows performing operations on encrypted data, without decryption. This will ensure that sensitive data is secure in every stage of its lifecycle even when accessed by untrusted cloud server. It was initially suggested by Craig Gentry (2009), and subsequently enhanced by cryptosystems such as the Pascal Paillier cryptosystem, and more modern ones such as the CKKS scheme to approximate computations.

This paper introduces a homomorphic encryption cloud computing safe data sharing design. The system enables users to encrypt their data and post it on the cloud where it is processed on encrypted data with no disclosure of the content. The results that have been processed are sent back encrypted and only authorized users can decode the results. The approach ensures data confidentiality, integrity and privacy and allows to work safely in collaboration and analyze data. The suggested system shows that homomorphic encryption can be a viable privacy-saving solution to computation and sharing of data in the current cloud applications, especially in areas where the sensitivity of data is paramount.

.

Keywords — Homomorphic Encryption, Secure Data Sharing, Cloud Computing Security, Privacy-Preserving Computation, Data Confidentiality, Cryptography, Encrypted Data Processing, Fully Homomorphic Encryption (FHE), Cloud Data Protection.

I. INTRODUCTION

The rapid growth of cloud computing and other digital technologies has transformed the process of data storage, processing, and sharing. Cloud platforms are gaining popularity among organizations and individuals because it is cost-effective, expandable and can be used to handle data in bulk. Nonetheless, this change has brought about grave issues of data security and privacy since sensitive data is usually stored and processed in third party servers which may not be entirely trusted. Conventional encryption methods are popular in securing information when transmitting and storing data. Even though these methods offer confidentiality, they require the data to be decrypted and only after this, one will be able to execute any computation or analysis. Such a decryption process creates a loophole in the process where sensitive data can be exposed to attackers or other unauthorized individuals. Due to this, more sophisticated security systems are increasingly required that are able to safeguard data even in the processing stage.

Homomorphic Encryption (HE) has come up as a groundbreaking solution to this issue. It also enables calculations to be done on encrypted data directly and without having to decrypt the data hence maintaining the privacy of data all the way through. Only recently in 2009 did the concept of fully homomorphic encryption by Craig Gentry

come into being which was a significant breakthrough in the cryptography domain. Also, some previous work by Pascal Paillier led to the creation of partially homomorphic encryption schemes which allow certain operations on encrypted data. As the need to share data in applications like healthcare, finance, and cloud analytics goes up, homomorphic encryption is a promising solution to the need to assure confidentiality and privacy. The study aims at applying homomorphic encryption methods to facilitate secure data sharing in the cloud to enable users to safely outsource data processing services without exposing sensitive data.

II. LITERATURE SURVEY

Homomorphic Encryption (HE) has recently been the focus of much attention as a potential remedy to safe data sharing and privacy-friendly computation. These studies have led to a number of researches that have helped in developing and enhancing homomorphic encryption schemes. Homomorphic encryption was based on the work of Pascal Paillier (1999), who proposed the Paillier cryptosystem, which is an additive homomorphic encryption scheme. This technique enables certain arithmetic operations to be carried out on encrypted data, which is convenient in securing calculations in the cloud settings.

Craig Gentry (2009) was also one of the first to make a significant breakthrough in this field, and in his publication Fully Homomorphic Encryption Using Ideal Lattices, he proposed the first Fully Homomorphic Encryption (FHE) scheme. This scheme provided arbitrary computation with encrypted data without ecryption and a new opportunity of processing the data safely.

Further refinements were made with Zvika Brakerski and Vinod Vaikuntanathan (2011) who developed more efficient FHE schemes to the Learning With Errors (LWE) problem. Their algorithm made computational complexity much simpler and the homomorphic encryption more viable. In 2010, Marten van Dijk and those before him suggested the scheme Fully Homomorphic Encryption over the Integers, which made FHE easier to implement and more available in the practical applications.

More recently Jung Hee Cheon et al. (2017) suggested the CKKS scheme that helps in approximate arithmetic operations with encrypted data. This progress can be particularly relevant to machine learning and real number computation of encrypted information. With these developments issues like high computational cost, large size of ciphertext and performance limits persist. Therefore, the recent studies are geared towards improving efficiency and scalability to enable homomorphic encryption to be applied to secure data sharing systems at a large scale.

1. Existing System

The most common approach to secure data sharing in the existing system is by using the traditional encryption algorithms such as symmetric and asymmetric encryption. The information is encrypted and stored or transferred to cloud servers to ensure privacy in storage and communication. However, to do any calculation or processing, the data is to be first decrypted, processed in its original form and then re-encrypted after which it is stored again. This process is very insecure, in that, valuable information is exposed in the process of decryption. The attackers get access to the original data in case of breach to the cloud server or processing environment.

The existing systems also fail in terms of privacy of data working in collaborative conditions when the number of users, who should access and process common data, is high. It is also more vulnerable to unauthorized access, data leakage and insider attacks since data are calculated in the plaintext form. Besides, these systems lack mechanisms of conducting secure computations on encrypted data. Although there are partially homomorphic encryption schemes (such as one by Pascal Paillier) where some operations can be performed on encrypted data, they can only perform specific mathematical operations and not complex computations. Therefore, the existing system lacks a fully secure system of processing sensitive data in an encrypted state and therefore it is prone to security risks in the cloud-based systems.

2. Proposed System

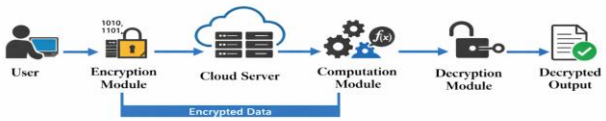
To address the shortcomings of the conventional encryption methods, the proposed system presents a safe model of data

sharing with the help of Homomorphic Encryption (HE). In this model, the user encrypts the data and uploads to the cloud server and therefore, data is not presented in the original form. The proposed model, in contrast to current systems, enables computations to be done on the encrypted data without the need to decrypt it. The system will be composed of a number of important modules, such as the user interface, encryption module, cloud server, computation module, and decryption module.

Firstly, data are entered in by the user and encrypted with a homomorphic encryption algorithm. The encrypted information is safely sent and stored in the cloud platform. When a computation request is made, the cloud server utilizes the homomorphic operations to work on the encrypted data and generates an encrypted result. system proposed guarantees the confidentiality, integrity and privacy of data but also enables the collaboration among the multiple users in a secure manner. It finds its use in particular areas, like healthcare, finance, and cloud-based analytics, when sensitive data should be processed without the loss of security. The system thus provides an efficient and reliable method of sharing data in cloud systems that are privacy conscious.

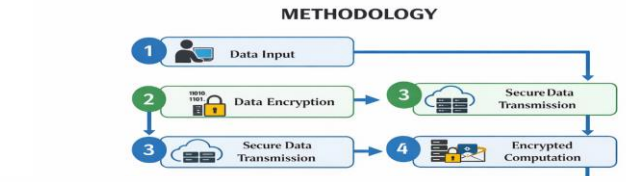
III. SYSTEM ARCHITECTURE

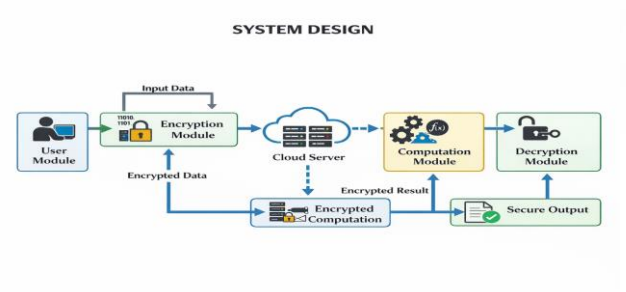
Secure data sharing architecture on Homomorphic Encryption (HE) is an architecture of system that aims at ensuring that sensitive information is not decrypted when storing and computing the information. The architecture will consist of a number of components that will interact with each other to provide safe processing of privacy-sensitive data in a cloud setup. The architecture begins with the User Module where the input of the user is sensitive data. This is then sent to the Encryption Module where it is encrypted using a homomorphic encryption algorithm. It can be computed on encrypted data without exposing the original data with the concept of fully homomorphic encryption which was invented by Craig Gentry. The information is coded and becomes safe and is transmitted to the Cloud Server which is encrypted. The cloud server does not have access to the decryption key, and therefore cannot access the information and abuse it. When a request is to be processed, the Computation Module directly works with the encrypted data, by the homomorphic properties.



IV. RELATED WORK

In the fields of cryptography and cloud security secure data sharing and computation that protects privacy have been extensively researched. Numerous researchers have put forth strategies to safeguard private information while allowing computation in environments that are outsourced. Craig Gentry (2009) introduced a Fully Homomorphic Encryption (FHE) scheme that allows arbitrary computations on encrypted data which was the first practical implementation of the homomorphic encryption concept. The groundwork for safe data processing in untrusted settings like cloud computing was established by this work. A partially homomorphic encryption scheme that permits additive operations on encrypted data was previously presented by Pascal Paillier (1999). Despite its efficiency it can only perform certain tasks and cannot handle the intricate calculations needed in contemporary applications. Later studies by Zvika Brakerski and Vinod Vaikuntanathan (2011) used the Learning With Errors (LWE) problem and lattice-based cryptography to increase the effectiveness of homomorphic encryption. Their efforts helped to improve the usefulness of FHE in practical applications. Marten van Dijk and colleagues made additional progress. (2010) who reduced implementation complexity by putting forth a more straightforward FHE scheme over integers. More recently Cheon Jung Hee et al. (2017) presented the CKKS scheme which is commonly utilized in encrypted machine learning applications and supports approximate arithmetic operations. High computational overhead big key sizes and performance inefficiencies are still problems despite these advancements. In order to provide quicker scalable and useful solutions for safe data sharing in cloud environments current research focuses on improving homomorphic encryption techniques.





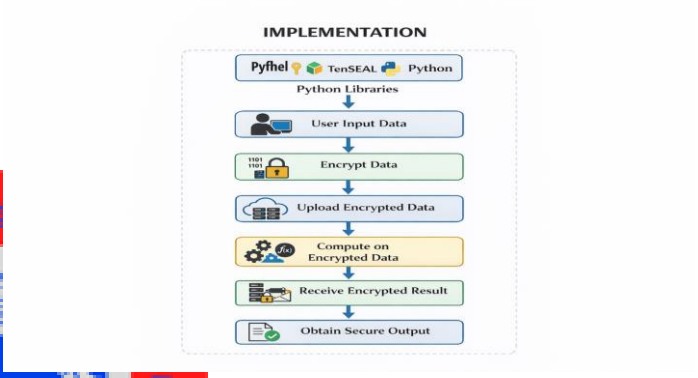
V. METHODOLOGY

Secure data sharing and privacy-preserving computation have been thoroughly studied in the domains of cloud security and cryptography. Several researchers have proposed methods to protect personal data while enabling computation in outsourced environments. The first practical application of the homomorphic encryption concept was Craig Gentrys (2009) Fully Homomorphic Encryption (FHE) scheme which permits arbitrary computations on encrypted data. This work laid the foundation for safe data processing in untrusted environments such as cloud computing. Pascal Paillier previously introduced a partially homomorphic encryption scheme that allows additive operations on encrypted data (1999). Despite its effectiveness it is limited to specific tasks and is unable to manage the complex computations required in modern applications. Later research by Zvika Brakerski and Vinod Vaikuntanathan (2011) increased the efficacy of homomorphic encryption by utilizing lattice-based cryptography and the Learning With Errors (LWE) problem. Their work improved FHEs usefulness in real-world applications. Further advancements were made by Marten van Dijk and associates. (2010) proposed a simpler FHE scheme over integers thereby lowering implementation complexity.

Cheon Jung Hee et al. more recently. (2017) introduced the CKKS scheme which supports approximate arithmetic operations and is frequently used in encrypted machine learning applications. Despite these improvements there are still issues with large key sizes high computational overhead and performance inefficiencies. Current research focuses on enhancing homomorphic encryption techniques to provide faster scalable and practical solutions for secure data sharing in cloud environments

VI. DESIGN AND IMPLEMENTATION

1. System Design



The User Module Encryption Module Cloud Server Computation Module and Decryption Module are the systems five primary modules. The User Module is in charge of receiving the final output and supplying the input data. Using homomorphic encryption the Encryption Module transforms plaintext data into ciphertext. Craig Gentrys idea guarantees that calculations on encrypted data can be carried out without disclosing the original data. After that the encrypted data is sent to the Cloud Server for safe storage. Homomorphic operations like addition and multiplication are used by the Computation Module to process the encrypted data. The cloud server cannot access the actual content because the data is still encrypted guaranteeing privacy. Lastly the Decryption Module enables the user to obtain the final result by decrypting the processed data using a private key.

2.Implementation

Python which has libraries for homomorphic encryption is one programming language that can be used to implement the system. Encryption decryption and encrypted computations

can be carried out using libraries such as Pyfhel or TenSEAL. The following are the fundamental steps of implementation. accepting data entered by the user. utilizing a homomorphic encryption algorithm to encrypt data. transferring data to the cloud server in encrypted form. working with encrypted data. giving the user results that are encrypted. obtaining the output by decrypting the result.

To confirm accuracy the system can be tested using straightforward arithmetic operations on encrypted data. The implementation is appropriate for practical applications in secure cloud computing environments since it shows that secure data sharing can be accomplished without disclosing sensitive information at any point.

VII. OUTCOME OF THE RESEARCH

The results of this study show that Homomorphic Encryption (HE) is an efficient way to accomplish secure data sharing in cloud environments. The suggested method effectively makes it possible to perform calculations directly on encrypted data without disclosing private information at any point during processing. Even when data is processed and stored on unreliable third-party servers this guarantees a high degree of data privacy and confidentiality.

Inspired by Craig Gentrys work the application of homomorphic encryption techniques demonstrates that secure computation is possible without the need for decryption. The system lowers the risk of data breaches and cyberattacks by preserving data integrity and preventing unauthorized access. The experimental findings show that the suggested method can perform simple mathematical operations on encrypted data including addition and multiplication with precise results following decryption. The increased security and privacy advantages outweigh the performance constraints even though there is some computational overhead when compared to conventional techniques.

Additionally the system facilitates safe cooperation by enabling multiple users to exchange and handle encrypted data without disclosing the original content. This makes it highly suitable for applications in sensitive domains such as healthcare finance and cloud-based analytics. Overall the study demonstrates that homomorphic encryption is a dependable and promising solution for secure data sharing and privacy-preserving data processing opening the door for more sophisticated and secure cloud computing systems in the future.

VIII. FUTURE UPDATES

Even though Homomorphic Encryption (HE) offers a solid basis for safe data exchange there are a number of areas in which the system could be strengthened. Future advancements seek to increase its real-world applicability while overcoming present constraints like high computational cost and performance overhead. Optimizing homomorphic encryption algorithms to increase speed and efficiency is one of the main upcoming updates.

Current HE schemes are slower than conventional approaches because they demand a large amount of computational power. The system will be more useful for large-scale applications if performance is improved through hardware acceleration (such as GPUs and specialized processors) and optimized algorithms. Integrating homomorphic encryption with cutting-edge technologies like machine learning and artificial intelligence is another crucial area. This will make it possible to train and analyze encrypted data securely without sacrificing privacy particularly in delicate industries like healthcare and finance. In order to balance security and performance future systems may also concentrate on creating hybrid encryption models that incorporate homomorphic encryption with additional cryptographic methods. Enhancing scalability will also enable the system to effectively handle big datasets and real-time data processing in cloud environments. To increase the systems accessibility for non-technical users user-friendly interfaces and implementations can be created.

Additionally current challenges in homomorphic encryption systems include reducing ciphertext expansion and key sizes which are being investigated. Future developments will primarily concentrate on improving homomorphic encryptions speed effectiveness and widespread adoption in order to provide safe and useful data sharing options for cloud computing systems of the future.

IX. CONCLUSION

In summary this study offers a safe and effective method for sharing data in cloud computing environments using Homomorphic Encryption (HE). By allowing computations to be carried out directly on encrypted data without the need for

decryption the suggested system overcomes the main drawback of conventional encryption methods. This guarantees that sensitive data is safeguarded at every stage of the data lifecycle including processing transmission and storage. The study emphasizes the value of homomorphic encryption as a potent instrument for protecting data confidentiality and privacy particularly in untrusted settings like cloud platforms. Craig Gentrys idea has been successfully applied to show secure computation without disclosing original data. Secure data processing can be accomplished while preserving data integrity and preventing unwanted access according to the systems design and implementation.

The advantages of improved security and privacy outweigh any potential computational overhead in the suggested system when compared to traditional techniques. Additionally the study demonstrates that homomorphic encryption is ideal for use in crucial fields where data security is crucial like big data analytics healthcare and finance. All things considered the suggested method offers a dependable and future-ready way to share data securely. Homomorphic encryption is anticipated to be crucial in creating safe privacy-preserving systems in contemporary cloud computing environments with ongoing developments and improvemet.

REFERENCES

1. Craig Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *ACM Symposium on Theory of Computing (STOC)*, 2009.
2. Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *EUROCRYPT*, 1999.
3. Marten van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," *IACR Cryptology ePrint Archive*, 2010.
4. Zvika Brakerski and Vinod Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE," *IEEE FOCS*, 2011.
5. Craig Gentry and Shai Halevi, "Implementing Gentry's Fully Homomorphic Encryption Scheme," *EUROCRYPT*, 2011.
6. Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan, "Fully Homomorphic Encryption without Bootstrapping," *ITCS*, 2012.
7. Jung Hee Cheon, Andrey Kim, Miran Kim, Yongsoo Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," *ASIACRYPT*, 2017.
8. Dan Boneh, Eu-Jin Goh, Kobbi Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," *Theory of Cryptography Conference (TCC)*, 2005.
9. Nigel Smart and Frederik Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," *Public Key Cryptography (PKC)*, 2010.
10. Craig Gentry, Shai Halevi, Nigel Smart, "Homomorphic Evaluation of the AES Circuit," *CRYPTO*, 2012.