International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

# A System for Implementing Homomorphic Encryption in Secure Computation for Cloud Computing Environment

[1] Dr. A. Velliangiri, [2]D. Sivakumar, [3]T. Sivaprakasam, [4]V. Nandini

**Submitted**: 05/02/2024 **Revised**: 13/03/2024 **Accepted**: 19/03/2024

**Abstract:** The system for implementing homomorphic encryption in a secure computation framework within a cloud computing environment enhances data security and privacy by enabling computations on encrypted data without necessitating decryption Through cloud services, people and organizations can use hardware and software resources that are remotely controlled by cloud providers. The client's data is physically located far from him, which poses a barrier because it could be accessed by a third party and have an impact on the confidentiality of client data. The most famous strategy for crossing over this security hole has been to scramble the distant information utilizing ordinary encryption calculations prior to moving it to the cloud supplier. To decode the information and execute the vital computations, the client should give the server the confidential key. Computations on scrambled information can be finished without first unscrambling it on account of homomorphic encryption. This paper explains how to encrypt client data on a cloud server using homomorphic encryption, which also makes it possible to do necessary calculations on the encrypted data. This innovative approach to secure computation offers a significant advancement in protecting sensitive data in the cloud, fostering greater trust and adoption of cloud-based solutions across diverse industries.

***Keywords:*** *Homomorphic, Encryption, Secure, Computation, Cloud Computing, Environment*

## 1. Introduction

The quick uptake of cloud computing, which offers major benefits in terms of scalability, affordability, and accessibility, has completely changed how businesses handle and process data. But in addition to these advantages, there are serious privacy and data security issues. Strong security measures are essential since the danger of unauthorized access and breaches rises when data is outsourced to outside cloud providers. Conventional encryption techniques perform poorly when it comes to processing encrypted data without exposing it to potential dangers, even though they are good at safeguarding data while it is in transit and at rest. This disparity highlights the need for sophisticated cryptographic techniques that can protect data while it is being computed.

Homomorphic encryption shows promise as a way to deal with these cloud computing security issues. Homomorphic encryption, in contrast to traditional encryption methods, enables calculations to be done directly on encrypted data, producing encrypted outputs that only the owner of the data can decipher. This feature reduces the possibility of data exposure even in an untrusted environment by guaranteeing that sensitive information is safeguarded during the whole computational process. Integrating a homomorphic encryption system into a cloud computing framework can greatly improve data processing operations' security and privacy. An all-encompassing solution to these security issues is the goal of the suggested method for integrating homomorphic encryption in a safe computation framework for cloud computing environments. The system allows for the execution of diverse computing activities, including complicated queries, machine learning, and data analysis, on encrypted data without jeopardizing secrecy through the integration of homomorphic encryption algorithms. This guarantees adherence to strict data privacy laws while still maintaining data integrity. In addition, the system's scalability and efficiency are guaranteed, minimizing performance overhead and guaranteeing interoperability with current cloud infrastructure. Consequently, enterprises may use the processing capacity of cloud services while upholding the strictest guidelines for data security and privacy, encouraging more confidence and wider acceptance of cloud computing.

## 2. Review of Literature

**Acar et al. (2018)** provides an extensive overview of homomorphic encryption (HE) schemes, focusing on both theoretical aspects and practical implementations. This paper is notable for its depth in covering the evolution of HE, from partially homomorphic encryption (PHE) to

[1]*Assistant Professor, Department of ECE, K.S.R.College of Engineering, Tiruchengode.*

[2]*Professor, Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Ramohalli Cross, Mysore Road, Bangalore 560074.*

[3]*Assistant professor/ECE Karpagam Academy of Higher education(Faculty of Engineering) Pollachi Main Road, Eachanari Post, Coimbatore - 641 021, Tamil Nadu, India.*

[4]*Associate Professor Dept of CSE Sona College of Technology*

fully homomorphic encryption (FHE). The authors systematically categorize HE schemes, discussing various constructions such as Gentry's breakthrough lattice-based FHE scheme and subsequent improvements aimed at enhancing efficiency and reducing computational overhead.

**Alkady, Farouk, and Rizk (2019)** explore the integration of fully homomorphic encryption (FHE) with the Advanced Encryption Standard (AES) to enhance cloud computing security. Presented at the International Conference on Advanced Intelligent Systems and Informatics, this work aims to combine the strengths of AES's efficiency and FHE's robust security guarantees. The authors propose a hybrid encryption scheme where AES is used for data encryption and FHE for secure computation, thereby addressing the performance limitations typically associated with FHE.

**Awadallah and Samsudin (2020)** present a focused discussion on the application of homomorphic encryption in cloud computing, emphasizing the inherent challenges. Their paper, presented at the IEEE International Conference on Engineering Technologies and Applied Sciences, highlights the growing importance of HE in ensuring data privacy in cloud environments, where data is often processed by third-party services.

## 3. Security of Cloud Computing

In numerous organizations, the utilization of cloud computing has developed fundamentally. All the while, the issues of securely reevaluating computing and outsider information security gain conspicuousness. Sending individual information to a cloud administration conveys the risk that it very well may be held for all time or utilized for different reasons by somebody with terrible goals, who might see it as significant. Furthermore, such information might be open to both homegrown and worldwide government elements, which could think twice about security.

Cloud computing raises some security concerns, including protection, outsider control, and information security. The three issues referenced above would be settled on the off chance that all information kept on cloud capacity stages were scrambled utilizing regular cryptosystems. The mystery key should be shared by the client and the cloud supplier to execute an essential computation on scrambled information put away in the cloud. The cloud supplier sends the client the result after first unscrambling the information to complete the necessary activities. The information should be scrambled involving a cryptosystem in light of homomorphic encryption to determine this issue. Taking into account that different cryptosystems empower computation on encoded information.

## 4. Homomorphic Encryption

A kind of encryption known as homomorphic encryption empowers explicit computations to be made on ciphertext and yield an encoded outcome, the result of which, when decoded, is equivalent to the result of playing out the procedure on the plaintext.

Since it ensures the protection of handled information, the homomorphic include assists with making a secure democratic framework with an elevated degree of protection in information recovery. It additionally use cloud computing. For example, to exhibit its numerical soundness, suppose there are two numbers, 10 and 20, which are both encoded to become 56 and 69, separately. Utilizing the expansion administrator, we get a worth of 125, which can be decoded to become 30.

### 4.1. History of Homomorphic Encryption

Ron Rivest and Leonard Adleman proposed the idea of homomorphic encryption in 1978. But the improvement has been extremely slow for thirty years. In 1982, their encryption scheme, which uses additive homomorphic encryption and can encrypt a single bit.

proposed an additional additive homomorphic encryption scheme. developed an encryption security method in 2005 that performs several adds but only one multiplication. Craig Gentry built a completely homomorphic encryption-based system in 2009 that can do addition and multiplication simultaneously.

### 4.2 Categories of Homomorphic Encryption

To some degree Homomorphic Encryption (PHE) and Completely Homomorphic Encryption (FHE) are the two essential kinds of homomorphic encryption calculations. PHE frameworks, as ElGamal, Paillier, RSA, and others, let expansion or increase to be finished on encoded information.

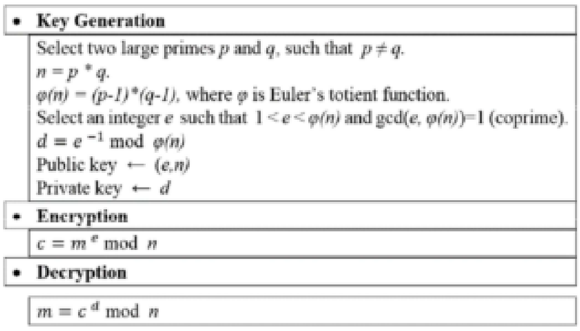
It was hard to develop a plan that could oblige the two exercises simultaneously. It was only after 2009 that the three-decade-old issue was tackled in original work by Nobility, where he exhibited that performing both expansion and duplication all the while is conceivable in completely homomorphic encryption, regardless of coming the nearest, permitting limitless increments and a solitary augmentation.

#### 4.2.1 Partially Homomorphic Encryption

#### A. Homomorphic Schemes with Multiplication

On the off chance that there is a calculation that can decide Enc(x × y) from Enc (x) and Enc (y) without requiring information on x and y, then, at that point, a homomorphic encryption is multiplicative [17]. like the ElGamal and RSA calculations. The RSA calculation is displayed in

Figure 2 as a delineation of a multiplicative homomorphic plot.



- **Key Generation**

Select two large primes $p$ and $q$, such that $p \neq q$.
$n = p * q$.
$\varphi(n) = (p-1)*(q-1)$, where $\varphi$ is Euler's totient function.
Select an integer $e$ such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n))=1$ (coprime).
$d = e^{-1} \bmod \varphi(n)$
Public key $\leftarrow (e,n)$
Private key $\leftarrow d$

- **Encryption**

$c = m^{e} \bmod n$

- **Decryption**

$m = c^{d} \bmod n$

**Fig 1:** RSA algorithm.

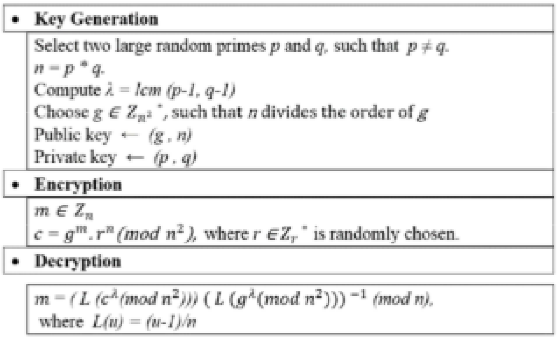
The RSA plan's multiplicative homomorphic trademark is as per the following.

$$\text{Given } c_1 = m_1{}^{e} \bmod \text{n},\ c_2 = m_2{}^{e} \bmod \text{n}$$

$$\begin{aligned} c_1 . c_2 &= E_{pk}(m_1) . E_{pk}(m_2) \\ &= m_1{}^{e} . m_2{}^{e} \ (\text{mod n}) = (m_1 . m_2)^{e} \ (\text{mod n}) \\ &= E_{pk}(m_1 . m_2) \qquad (1) \end{aligned}$$

**B. Additive Homomorphic Schemes**

If an algorithm exists that can compute Enc(x + y) from Enc(x) and Enc (y) without requiring knowledge of x and y, then a homomorphic encryption is additive [17]. like the Goldwasser-Micali and Paillier algorithms. The Pathier algorithm is shown in Figure 3 as an example of an addtive homomorphic scheme.



- **Key Generation**

Select two large random primes $p$ and $q$, such that $p \neq q$.
$n = p * q$.
Compute $\lambda = lcm\ (p-1, q-1)$
Choose $g \in Z_{n^2}{}^{*}$, such that $n$ divides the order of $g$
Public key $\leftarrow (g\ ,\ n)$
Private key $\leftarrow (p\ ,\ q)$

- **Encryption**

$m \in Z_n$
$c = g^{m} . r^{n}\ (mod\ n^2)$, where $r \in Z_r{}^{*}$ is randomly chosen.

- **Decryption**

$m = (\ L\ (c^{\lambda}(mod\ n^2)))\ (\ L\ (g^{\lambda}(mod\ n^2)))^{-1}\ (mod\ n)$,
where $L(u) = (u-1)/n$

**Fig 2:** Paillier algorithm.

The following illustrates the Paillier scheme's homomorphic characteristic.

$$\begin{aligned} \text{E}(m_1) . \text{E}(m_2) &= (g^{m_1} .\ r_1{}^{n})\ (g^{m_2} .\ r_2{}^{n}) \\ &= g^{m_1 + m_2}\ (r_1 + r_2)^{n} \\ &= \text{E}(m_1 + m_2\ (\text{mod n})) \end{aligned}$$

**4.2.2Fully Homomorphic Encryption**

Except for the Boneh-Goh-Nissim plot, which grants leading an endless number of expansion activities however just a single increase, all PHE strategies license homomorphic computation of only one activity, either expansion or duplication, on scrambled dates. It was accepted to be inconceivable until 2009, when Craig Upper class proposed the main viable development of a completely homomorphic plot [3]. The development of a plan that licenses one to register erratic computation (a plan ought to permit a limitless number of both expansion and duplication tasks) over scrambled information has stayed a focal open issue in cryptography for over 30 years.

Gentry's approach supports simultaneous addition and multiplication, which in Boolean algebra correspond to AND (∧) and XOR (⊕). The ability to transform any calculation into a function that just contains (∧) and (⊕) is a significant benefit of providing these two Boolean functions. There are various methods in algebra that can be applied to simplify a function. This method allows a function to be changed to exclusively employ certain Boolean operations (such ∧ or ⊕). One way to write ¬A is as A ⊕ 1. Another example is A ∨ B, which can be transformed into (¬A) ∧ (¬B), which can then be further transformed into (A ⊕ 1) ^ (B ⊕ 1). These methods can be used to transform any function into a sequence of (∧) and (⊕) operations. Gentry's work is based on this [19]. The distinction between completely homomorphic systems and standard encryption techniques—which do not use PHE—is seen in Figure 3.

Gentry employs cryptography based on lattices. His suggested completely homomorphic encryption is broken down into multiple steps: begin with what was described as an ideal lattice-based, slightly homomorphic encryption technique that is restricted to assessing low-degree polynomials over encrypted data. It is constrained by the fact that every ciphertext has some level of noise, which increases with the number of ciphertexts added and multiplied until the noise eventually renders the resultant ciphertext unintelligible. It then breaks down the decryption process into a form that can be stated as a low-degree polynomial that the scheme can handle. Ultimately, it employs a bootstrapping transformation to produce a fully homomorphic scheme via a recursive self-embedding.

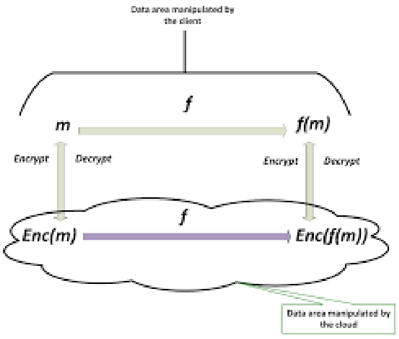

Data area manipulated by the client
f
m
f(m)
Encrypt
Decrypt
Encrypt
Decrypt
f
Enc(m)
Enc(f(m))
Data area manipulated by the cloud

**Fig 4:** distinction between completely homomorphic algorithms and standard encryption schemes

## 5. Conclusion

homomorphic encryption is a revolutionary method to data security and privacy when used in secure computation for cloud computing environments. Homomorphic encryption guarantees that sensitive data is safeguarded throughout the processing lifespan by allowing calculations on encrypted data without the need to decode it. This method preserves confidentiality even in possibly untrusted contexts, hence addressing important security concerns related to cloud computing, such as data leaks and unauthorised access. Although complicated key management and large computational overhead remain obstacles, new developments and hybrid encryption models present viable ways to improve effectiveness and usability. The incorporation of homomorphic encryption techniques will be essential to maintaining data security, building trust, and providing safe, private cloud services as cloud computing grows.

## References

[1] Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (Csur), 51(4), 1-35.

[2] Alkady, Y., Farouk, F., & Rizk, R. (2019). Fully homomorphic encryption with AES in cloud computing security. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2018 4 (pp. 370-382). Springer International Publishing.

[3] Awadallah, R., & Samsudin, A. (2020, December). Homomorphic encryption for cloud computing and its challenges. In 2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-6). IEEE.

[4] Awasthi, P., Mittal, S., Mukherjee, S., & Limbasiya, T. (2019). A protected cloud computation algorithm using homomorphic encryption for preserving data integrity. In Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017, Volume 1 (pp. 509-517). Springer Singapore.

[5] Das, D. (2018, January). Secure cloud computing algorithm using homomorphic encryption and multi-party computation. In 2018 International Conference on Information Networking (ICOIN) (pp. 391-396). IEEE.

[6] Geng, Y. (2019). Homomorphic encryption technology for cloud computing. Procedia Computer Science, 154, 73-83.

[7] Hallman, R. A., Diallo, M. H., August, M. A., & Graves, C. T. (2018, March). Homomorphic Encryption for Secure Computation on Big Data. In IoTBDS (pp. 340-347).

[8] Ibtihal, M., & Hassan, N. (2020). Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. In Cryptography: breakthroughs in research and practice (pp. 316-330). IGI Global.

[9] Jin, B. W., Park, J. O., & Mun, H. J. (2019). A design of secure communication protocol using RLWE-based homomorphic encryption in IoT convergence cloud environment. Wireless Personal Communications, 105, 599-618.

[10] 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.

[11] Park, J., Kim, D. S., & Lim, H. (2020). Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. IEEE Access, 8, 203564-203579.

[12] Thabit, F., Can, O., Alhomdy, S., Al-Gaphari, G. H., & Jagtap, S. (2022). A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing. International Journal of intelligent networks, 3, 16-30.