Educational Administration: Theory and Practice
2024, 30(4), 7347-7353
ISSN: 2148-2403
https://kuey.net/

Research Article

Enhancing Data Security In Multi-Cloud Settings With Homomorphic Encryption: Concepts, Uses, And Obstacles

Dr.P.Thangavel[1*], Dr.P.Shyamala Anto Mary [2,] Mr.M.RameshKannan[3], Dr.K.Deiwakumari[4]

[1*]Assistant Professor, Department of Computer Science, SRM Trichy Arts and Science College, Trichy-621 105 , Tamilnadu, India,velmcaccet@gmail.com
[2]Assistant Professor ,Department of Mathematics, SRM TRP Engineering College, Trichy-621 105, Tamilnadu, India,shyamkarthi12@gmail.com
[3]Assistant Professor, Department of Computer Science, SRM Trichy Arts and Science College, Trichy-621 105 , Tamilnadu, India, rameshkannanm@yahoo.com
[4]Assistant Professor Department of Mathematics, Sona College of Technology, Salem- 636 005, Tamilnadu, India

Citation: Dr.P.Thangavel, et al (2024), Enhancing Data Security In Multi-Cloud Settings With Homomorphic Encryption: Concepts, Uses, And Obstacles, Educational Administration: Theory and Practice, 3(4), 7347-7353
Doi: 10.53555/kuey.v30i4.1691

ARTICLE INFO **ABSTRACT**

Homomorphic encryption allows calculations on encrypted data without requiring decryption; it offers a revolutionary approach to data security in multi-cloud contexts. This paper explores the theory, applications, and difficulties of homomorphic encryption in relation to cloud computing, emphasising the technology's potential to enhance data security and privacy. The benefits and drawbacks of fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE) are covered. The practical uses of homomorphic encryption in multi-cloud scenarios—including secure data analysis, cooperative computing, and machine learning model training—are covered in this work. The research covers computational overhead, enhanced key management, and interoperability with existing cloud infrastructures in addition to these crucial topics. Further research directions are offered to solve these challenges and improve the capabilities of homomorphic encryption. These include of improving FHE algorithms, accelerating critical administration processes, and looking into new uses in multi-cloud environments. Homomorphic encryption has the potential to revolutionise data security in multi-cloud environments and enable secure and confidential computing on dispersed cloud infrastructures.

Keywords—Homomorphic Encryption, Cloud Computing, Data Security, Privacy, Encryption, Cryptography

Introduction

The advent of cloud computing has revolutionized the way that computing is done today. It allows businesses to delegate complex computational tasks to distant cloud servers, all while gaining centralized computational power, efficiency, and scalability. Cloud computing has some benefits, such as low costs and efficient use of resources, but it also poses new security issues, particularly in relation to the privacy, integrity, and confidentiality of customer data. Customers should encrypt their data before sending important information to the cloud to guard against misuse and unauthorized access. However, as standard encryption methods need decryption before processing, cloud servers have substantial hurdles when executing computations on encrypted data. As a solution to these problems, homomorphic encryption has become a popular cryptographic method that eliminates the need for decryption and enables computation to be done directly on encrypted data. With this method, customers can take advantage of cloud servers' processing capacity without compromising the privacy of their data. A major development in the realm of cryptography was brought about in 2010 when Craig Gentry proposed Fully Homomorphic Encryption (FHE), which allows users to execute both additive and multiplicative homomorphic operations on encrypted data [1]. Although FHE has great promise to improve cloud computing data security, there aren't many real-world applications of this approach yet. The concepts of homomorphic encryption are examined in this study along with how it might be used to secure data in cloud computing settings. We analyze the difficulties in executing calculations on encrypted data and how homomorphic encryption can help to overcome these difficulties. Furthermore, we go over the shortcomings of the encryption techniques that are currently in use as well as

Copyright © 2024 by Author/s and Licensed by Kuey. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the benefits that homomorphic encryption provides in terms of data privacy and confidentiality in cloud contexts. This study intends to add to the existing discussion on enhancing data security in cloud computing through creative cryptographic solutions by giving an overview of the state of the art in homomorphic encryption algorithms and suggesting future research directions.

1.1Motivation of Problem

The increasing use of third-party cloud services for computing is raising serious security concerns, including risks to data security, privacy, confidentiality, integrity, and authentication. It becomes more difficult to maintain good security standards when you depend on other parties to offer critical services since it becomes harder to protect sensitive information. In response to these concerns, many users decide to lower security risks by keeping their data in encrypted form on the cloud. On the other hand, decryption is necessary in order to operate on encrypted data in the cloud, which may jeopardize the privacy and confidentiality of the data that is stored. A potential answer to these issues seems to be homomorphic encryption, or HE, which permits calculations to be done on the encrypted data itself. Unlike traditional encryption techniques that require decryption before data processing, HE allows operations to be performed instantly on ciphertexts, producing encrypted results that are equivalent to those obtained from processing plaintext data. Users can preserve secrecy and privacy in cloud settings by using HE to do computations on their encrypted data without revealing it in its original format.

I. Introduction to Homomorphic Encryption

A revolutionary development in cryptography is homomorphic encryption, which enables calculations on encrypted data without the need for pre-decryption. This implies that data confidentiality and privacy can be maintained while processing it by keeping it encrypted. In algebra, the term "homomorphic" describes a characteristic of functions that maintains specific operations. Mathematical operations on encrypted data, including addition and multiplication, are preserved by homomorphic encryption systems. Partially homomorphic encryption (PHE) and fully homomorphic encryption (FHE) are the two main types of homomorphic encryption.

TABLE I. Homomorphic Encryption type

Homomorphic Encryption Type	**Description**	**Example Schemes**	**Key Features**
Partially Homomorphic Encryption (PHE)	Supports encryption-based addition and multiplication operations, but not both.	RSA, Paillier	1] Restricted use of encrypted data. 2] Utilized in the field of cryptography. 3] Does not allow for both multiplication and addition
Fully Homomorphic Encryption (FHE)	Enables complicated computations without the need for decryption by supporting addition and multiplication operations on encrypted data.	Craig Gentry's techniques (2009)	1] Supports methods involving addition and multiplication. 2] Allows for complex calculations to be performed on encrypted data. 3] Computationally more difficult than PHE

2.1 Types of Homomorphic Encryption

The innovative cryptographic method known as homomorphic encryption is available in several varieties according to the extent of operations that can be performed on encrypted data. There are two primary varieties of homomorphic encryption:

Partially Homomorphic Encryption (PHE):

Partially homomorphic encryption schemes support only one type of operation on encrypted data, either addition or multiplication, but not both simultaneously. There are two common types of PHE:

1] Additive Homomorphic Encryption: In additive homomorphic encryption schemes, computations involving addition on encrypted data are possible while maintaining the encryption. However, multiplication operations are not supported without additional techniques.

2] Multiplicative Homomorphic Encryption: Conversely, multiplicative homomorphic encryption schemes allow computations involving multiplication on encrypted data while preserving the encryption. Addition operations are typically not supported in these schemes without additional techniques.
Examples of partially homomorphic encryption schemes include the RSA cryptosystem (multiplicative homomorphism) and the Paillier cryptosystem (additive homomorphism) [4].

Fully Homomorphic Encryption (FHE):

Introduced by Craig Gentry in 2009 [1], fully homomorphic encryption (FHE) algorithms are incredibly effective yet computationally demanding because they allow both addition and multiplication operations on encrypted data. Numerous applications in secure computation and privacy-preserving data analysis are made possible by FHE, which enables complicated computations to be carried out directly on encrypted data without the need for decryption [2]. On the other hand, compared to PHE, FHE usually takes more computer power and is more difficult to execute. FHE methods offer a significant degree of freedom for calculations on encrypted data because they are made to permit arbitrary combinations of addition and multiplication operations. Even though FHE has computing difficulties, research is still being done to increase its effectiveness and usability so that it can be used in more practical ways. These kinds of homomorphic encryption are essential for maintaining data security and privacy in a number of areas, such as cloud computing, safe outsourcing, and processing private data.

2.2 Symmetric key cryptography

Symmetric key cryptography, sometimes referred to as private key cryptography, uses a shared secret key between the sender and the recipient to encrypt and decrypt messages. The fact that the encryption and decryption keys used in this technique are identical is what is meant by "symmetric". This method is not the same as asymmetric key cryptography, which uses different keys for data encryption and decryption. Block ciphers, sometimes referred to as stream ciphers, are the foundation of symmetric key cryptography. While stream ciphers encrypt a single character or piece of data at a time, block ciphers handle data in fixed-size blocks. Two popular cryptographic protocols that are commonly utilized as block cipher designs are the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES). A major disadvantage of symmetric key cryptography is the need for secure key management. Since the same key is used for both encryption and decryption, maintaining the confidentiality of communications depends on how these keys are distributed and managed. The encryption of the communication could be compromised by any intrusion into the key management process, which involves key creation, distribution, storage, and revocation. All things considered, symmetric key cryptography is suitable for many applications where two people may safely exchange a secret key since it offers a rapid and efficient means of encrypting and decrypting data.

2.3 Asymmetric key cryptography

Two keys are used in asymmetric key cryptography, also known as public key cryptography. These keys are a public key and a private key. Unlike symmetric key cryptography, which employs the same key for both operations, asymmetric key cryptography uses separate keys for encryption and decryption. In this arrangement, the public key is widely distributed and available to anyone wishing to securely communicate with the key owner. On the other hand, the private key is secret and accessible only to the key owner. Public keys can be used to encrypt messages, and private keys can be used to decrypt them. Compared to symmetric key encryption, asymmetric key cryptography has several advantages. Since the public key can be distributed freely without jeopardizing security, it eliminates the necessity for secure key distribution, which is one of its main benefits. Asymmetric key cryptography also makes safe digital signatures possible. This lets the sender to sign a communication with their private key, ensuring authentication and non-repudiation. The Elliptic Curve Cryptography (ECC) algorithm, which is based on the mathematical characteristics of elliptic curves, and the RSA technique, which is based on the difficulty of factoring big prime numbers, are two popular asymmetric key techniques. Despite having more capabilities and more security than symmetric key encryption, asymmetric key cryptography can occasionally be slower and need more computing resources. In hybrid cryptographic systems, asymmetric key cryptography is consequently often used in conjunction with symmetric key encryption to achieve both security and efficiency

II. RELATED WORK

Over the years, there have been notable breakthroughs and contributions in the field of homomorphic encryption. In order to solve security and privacy concerns in cloud computing and other applications, a number of researchers have investigated various elements of homomorphic encryption schemes. Here, we summarize some of the current approaches and findings from this field of study. In cloud computing contexts, the core application scenarios of various homomorphic encryption cryptosystems, including RSA, Paillier, El Gamal, and Gentry, were analyzed by Maha Tebba et al. [5]. They contrasted various cryptosystems according to important characteristics such as the kind of homomorphic encryption, data privacy, security uses, and key kinds. In order to improve security in cloud computing, Reem Alattas et al. [6] introduced the use of algebraic homomorphic encryption techniques based on Fermat's Little Theorem. Their method sought to improve data privacy and confidentiality in cloud environments by utilizing mathematical concepts. The goal of parallelizing completely homomorphic encryption for cloud environments was the focus of Ryan Hayward and Chia-Chu Chiang [7]. In order to enhance the efficiency of completely homomorphic encryption systems and make them more useful for practical applications, their work investigated the use of parallel processing techniques. A thorough introduction to fully homomorphic encryption was given by Frederik Armknecht et al. [8], who also offered insights into the theoretical underpinnings and real-world applications of this encryption method. Their goal was to close the gap that exists between completely homomorphic encryption theory and real-world implementations. Furthermore, the construction of numerous homomorphic encryption algorithms has been made possible by the groundbreaking research of scholars like Craig Gentry [1], Yao [8], and Rivest et al. [9]. While Yao [8] suggested protocols for safe computations, Rivest et al. [10] established the idea of privacy homomorphisms. An important development in the field was Craig Gentry's seminal work on fully homomorphic encryption using ideal lattices [1], which created new avenues for safe computing on encrypted data. All things considered, a broad spectrum of research and works have been done in the field of homomorphic encryption with the goal of improving the security, privacy, and effectiveness of computations in the cloud and other areas.

III. Security Issues in Cloud Computing

Security is a primary concern in cloud computing due to its intrinsic distributed nature and reliance on external infrastructure. With this dispersed design, it is challenging to manage the security of essential services including data security, privacy, confidentiality, integrity, and authentication. Users usually decide to store their data in encrypted form in the cloud in order to minimize security risks. However, in order to perform operations on encrypted data stored in the cloud, the cloud provider must first decrypt the data, raising questions around data privacy and confidentiality. A potential encryption technique that makes it possible to perform operations directly on ciphertext and produce encrypted results that, when decoded, resemble the results of operations performed on plaintext data is homomorphic encryption (HE). The issues of data privacy and confidentiality in cloud processing and storage environments are addressed by this capability [1]. Without sacrificing data security, HE approaches let customers utilize cloud computing capabilities while keeping control over their sensitive data. Even with all of the potential advantages of HE, there are still issues with its effectiveness and practical application. Compatibility with current cloud infrastructures, performance overheads, and complexity in key management must all be addressed for HE to fully realize its benefits in practical settings. In addition, for HE algorithms to be practically deployed in cloud environments, they must balance computational efficiency with security requirements. In the end, even though homomorphic encryption has the potential to improve cloud computing security by enabling safe processing of encrypted data, its acceptance and application will necessitate resolving a number of practical and technological issues. Overcoming these obstacles will enable HE to make a substantial contribution in mitigating security issues and guaranteeing the privacy and confidentiality of data in cloud computing settings.

IV. CRYPTOGRAPHIC TECHNIQUES IN CLOUD COMPUTING

In cloud computing, various cryptographic techniques are employed to ensure data security and privacy. Three commonly used cryptographic techniques include:

RSA Cryptosystem: In cloud computing contexts, the RSA cryptosystem—named for its founders, Rivest, Shamir, and Adleman—is frequently utilized for secure communication and data encryption. The difficulty of factoring big composite numbers into their prime factors is the foundation for it [11]. Large prime numbers' mathematical characteristics are used by RSA encryption to safely encrypt and decrypt data.

Paillier Cryptosystem: Since the Paillier cryptosystem may execute computations on encrypted data without first decrypting it, it is a popular additive homomorphic encryption approach in cloud computing. It is especially helpful for calculations that protect privacy, like adding encrypted values, and enables the safe aggregation of encrypted data [12].

Elliptic Curve Cryptography (ECC): The method for public-key encryption called elliptic curve cryptography is based on the algebraic structure of elliptic curves over finite fields. ECC is especially well-suited for resource-constrained contexts, including cloud computing platforms, as it provides security comparable to RSA but with reduced key sizes [13].

These cryptographic algorithms enable secure communication, data storage, and computation while protecting the confidentiality and integrity of sensitive information, which is vital for safeguarding data and guaranteeing privacy in cloud computing environments.

V. HOMOMORPHIC ENCRYPTION IN CLOUD COMPUTING

In cloud computing, homomorphic encryption is a cryptography approach that enables calculations on encrypted data without requiring decryption. By allowing data to stay encrypted throughout processing, this protects security and privacy, particularly in cloud environments where private data is maintained and kept remotely. The two primary types of homomorphic encryption systems are fully homomorphic encryption (FHE) and partly homomorphic encryption (PHE). PHE restricts the type of operation (multiplication or addition) that can be carried out on encrypted data without compromising the accuracy of the outcome. On the other hand, because FHE allows addition and multiplication operations to be performed on encrypted data, it provides greater flexibility but often requires more computing power.

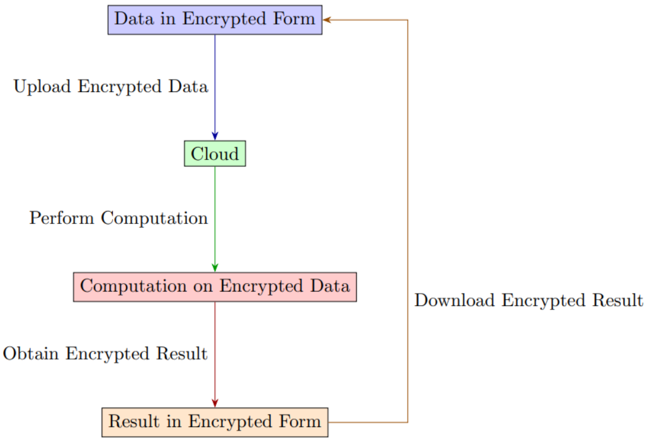

Fig: Cloud Computing Using Homomorphic Encryption

Before being delivered to the cloud for processing, the data in this diagram is encrypted. The cloud maintains the confidentiality of the encrypted data by doing the required calculations directly on it. Prior to being returned to the user, the computation's outcome is likewise encrypted, which they can decrypt with their private key. In cloud computing, homomorphic encryption provides a potent way to guarantee data security and privacy while permitting safe cloud-based computation task outsourcing. But it has computational overhead and complexity, particularly when it comes to FHE, which needs a lot of processing power. Therefore, the goal of current research and technological developments in homomorphic encryption is to overcome these obstacles and increase the usefulness of homomorphic encryption in practical cloud computing applications.

6.1 Partially Homomorphic Encryption

A cryptographic system that demonstrates either additive or multiplicative homomorphism, but not both at the same time, is said to be partially homomorphic. RSA (based on multiplicative homomorphism), Paillier [12] (based on additive homomorphism), and ElGama [14] (also based on multiplicative homomorphism) are a few examples of partially homomorphic encryption methods.

6.2 Fully Homomorphic Encryption

A cryptographic system that demonstrates both additive and multiplicative homomorphism capabilities is referred to as fully homomorphic encryption. In 2010, Craig Gentry suggested and constructed a lattice-based cryptosystem, which is the first and so far only fully homomorphic encryption system [1]. It is thought that fully homomorphic encryption is the most effective and potent method of protecting data that is outsourced. This approach enables homomorphic computation on low degree polynomials. Scenarios concerning the malleability quality of homomorphic encryption schemes are being actively investigated by researchers [15]. The encryptions of plaintexts M1 and M2 are designated as Encr (M1) and Encr(M2) in a fully homomorphic encryption method. Encr (M1+M2) and Encr (M1×M2) can be computed safely and effectively since fully homomorphic encryption provides both additive and multiplicative features. In order to overcome the difficulties associated with fully homomorphic encryption, researchers are also looking into workable scenarios and solutions, which includes examining current computing tools and capabilities. But there are still a lot of unanswered concerns and real-world problems that the scientific community needs to work out in light of Gentry's innovative idea.

VI. PRACTICAL APPLICATIONS OF HOMOMORPHIC ENCRYPTION IN MULTI-CLOUD ENVIRONMENTS

Beyond data security and privacy, homomorphic encryption has several real-world uses in multi-cloud scenarios. Secure data analysis across several cloud platforms is one such application. Organizations can do sophisticated data analytics on encrypted data from many cloud providers without compromising the safety of confidential data by utilizing homomorphic encryption. Furthermore, in a multi-cloud configuration, homomorphic encryption permits safe cooperative computing amongst numerous parties. For example, businesses are able to collaborate on calculations and exchange encrypted data between various cloud environments all while protecting the privacy of their data. When working with sensitive material, this feature is especially helpful for remote teams or companies collaborating. Moreover, in multi-cloud environments, homomorphic encryption can help with safe and private machine learning and artificial intelligence (AI) model training. Businesses can protect data privacy and confidentiality by using encrypted data from different cloud providers to train machine learning models without disclosing the underlying data.

VII. Overcoming Challenges and Future Research Directions

Enhancing data security in multi-cloud systems can be achieved through the use of homomorphic encryption. To achieve its full potential and enable its widespread adoption, a few issues must be resolved. Furthermore, there are other directions for further investigation in this area that may enhance the potential and suitability of homomorphic encryption. The computational expense of fully homomorphic encryption (FHE) is one of the main obstacles to the deployment of homomorphic encryption in multi-cloud scenarios. The encryption, decryption, and computation of encrypted data necessitate a large amount of processing power and time when using FHE algorithms. It is imperative to tackle these computational obstacles in order to render FHE feasible for actual use cases. To lessen these computational limitations, future research endeavors might concentrate on developing effective hardware implementations and refining FHE algorithms. The complexity of key management in homomorphic encryption systems is another crucial factor to take into account. The creation, sharing, and storing of public and private keys constitute asymmetric key management, which can be laborious and vulnerable to security flaws. Encrypted data integrity and confidentiality must be guaranteed by optimizing key security measures and streamlining key management procedures. Subsequent studies may examine new methods for key management in homomorphic encryption systems, such as the application of sophisticated cryptographic protocols and safe multi-party computation strategies. Furthermore, the smooth integration of homomorphic encryption into multi-cloud systems depends on its interoperability with current cloud infrastructures. Systems for homomorphic encryption must be built to function well with a range of cloud services and platforms, supporting a variety of data formats, security needs, and protocols. Furthermore, the smooth integration of homomorphic encryption into multi-cloud systems depends on its interoperability with current cloud infrastructures. Systems for homomorphic encryption must be built to function well with a range of cloud services and platforms, supporting a variety of data formats, security needs, and protocols. Subsequent investigations may concentrate on creating homomorphic encryption standards and protocols that are compatible with a variety of cloud environments.

Future research in homomorphic encryption should investigate novel applications and use cases in multi-cloud contexts, in addition to tackling technical issues. Compute that is secure and maintains privacy across distant cloud platforms is becoming more and more important as multi-cloud techniques proliferate in enterprise settings. In multi-cloud scenarios, homomorphic encryption can be extremely important for providing secure data processing and analysis while maintaining data privacy and confidentiality. Subsequent investigations may concentrate on pinpointing certain scenarios and uses where homomorphic encryption can yield noteworthy advantages concerning data protection, confidentiality, and adherence to regulations. Overall, interdisciplinary cooperation and creative research efforts will be needed to overcome the computational, key management, and interoperability issues related to homomorphic encryption. Homomorphic encryption has the ability to transform data security in multi-cloud systems and open the door for safe and private computation in the digital era by tackling these issues and investigating new uses.

VIII.Future Scope

In multi-cloud settings, homomorphic encryption has great promise for revolutionizing data security and privacy across a range of applications and industries. As more companies employ multi-cloud strategies to capitalize on distributed computing's benefits, robust security measures become increasingly important. Homomorphic encryption offers a transformational solution while preserving data privacy and confidentiality by enabling calculations to be performed directly on encrypted data without the need for decryption. Homomorphic encryption can be a vital component in multi-cloud scenarios, where confidential data is transferred across several cloud platforms and service providers, guaranteeing complete security and compliance. Organizations can conduct secure and privacy-preserving computations across diverse cloud environments while keeping control over their sensitive data by including homomorphic encryption into multi-cloud architectures. Moreover, the future application of homomorphic encryption in multi-cloud systems will cover cutting-edge technologies like edge computing and the Internet of Things (IoT) in addition to conventional data processing [16]. The requirement for safe and privacy-preserving compute at the network edge is growing as a result of the growth of edge devices and Internet of Things sensors that are producing enormous volumes of sensitive data. By enabling edge devices and Internet of Things endpoints to process encrypted data locally, homomorphic encryption can reduce the privacy issues associated with sending raw data to centralized cloud servers [17]. By minimizing the need for data transmission over the network, this decentralized method not only increases data privacy and security but also improves latency and bandwidth efficiency. prerequisites. The use of homomorphic encryption has great promise to provide safe and effective data processing at the network edge while maintaining strict privacy and security standards, especially as edge computing continues to gain traction in multi-cloud settings.

IX. Conclusion

Homomorphic encryption provides a workable solution for enhancing data security and privacy in multi-cloud environments. By enabling computations on encrypted data without the need for decryption, it resolves significant problems with data integrity and privacy in cloud computing. This paper provides an overview of homomorphic encryption, covering its theory, applications, and challenges. The benefits and drawbacks of fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE) have been discussed. The application of homomorphic encryption in real-world multi-cloud scenarios, such as secure data analysis, cooperative computing, and machine learning model training, has been studiedA few concerns need to be fixed before homomorphic encryption is commonly employed in multi-cloud systems. Among these challenges are complex key management, computational overhead, and compatibility with existing cloud infrastructures. Future research should concentrate on improving FHE algorithms, accelerating crucial administration processes, and looking into new applications in multi-cloud systems in order to overcome these problems.

Ultimately, homomorphic encryption has the ability to completely transform data security in multi-cloud environments by enabling private and safe computation across dispersed cloud platforms. Further research and development effort is required in order to fully realise the benefits of homomorphic encryption and address any outstanding difficulties. Homomorphic encryption has the potential to play a significant role in ensuring data security and privacy as multi-cloud computing gets more complex.

References

1. Craig Gentry, "Fully homomorphic encryption using ideal lattices," Proceedings of the forty-first annual ACM symposium on Theory of computing. ACM, 2009.
2. Jonathan Katz and Yehuda Lindell, "Introduction to Modern Cryptography," CRC Press, 2021.
3. Vinod Vaikuntanathan, "Homomorphic Encryption for Secure Computation," Communications of the ACM, 2010.
4. D. H. Lee and K. Lee. “Multi-Client Order-Revealing Encryption”. Jan. 2018.
5. Maha Tebba et al., "Secure Cloud Computing through Homomorphic Encryption," International Journal of Advancements in Computing Technology (IJACT), Volume-5, Number-16, December 2013.
6. Reem Alattas, Khaled Elleithy, "Cloud Computing Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem," The American Society of Engineering Education, ASEE 2013, Northfield, VT, USA, 09 December 2016.
7. Ryan Hayward, Chia-Chu Chiang, "Parallelizing fully homomorphic encryption for a cloud environment," Journal of Applied Research and Technology 13 (2015), 245-252.
8. Frederik Armknecht et al., "A Guide to Fully Homomorphic Encryption," IACR, 2015.
9. Rivest, Ronald L., Len Adleman, Michael L. Dertouzos, "On data banks and privacy homomorphisms," Foundations of secure computation 4.11 (1978): 169-180.
10. C. Yao, "Protocols for secure computations" (extended abstract). In 23rd Annual Symposium on Foundations of Computer Science (FOCS '82), pages 160-164. IEEE, 1982.
11. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.
12. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," In Advances in Cryptology—EUROCRYPT’99, Springer Berlin Heidelberg, pp. 223-238, 1999.
13. N. Koblitz and A. J. Menezes, "Pairing-based cryptography," in Encyclopedia of Cryptography and Security, Springer US, pp. 925-926, 2011.
14. ElGamal, Taher, ”A public key cryptosystem and a signature scheme based on discrete logarithms”, Advances in cryptology. Springer Berlin Heidelberg, 1985.
15. Jean-Sébastien Coron et al., "Fully Homomorphic Encryption over the Integers Revisited," Advances in Cryptology – EUROCRYPT 2015, Springer Berlin Heidelberg, 2015.
16. Brakerski, Zvika, and Vinod Vaikuntanathan. "Fully homomorphic encryption from ring-LWE and security for key dependent messages." Advances in Cryptology – EUROCRYPT 2011. Springer Berlin Heidelberg, 2011.
17. Dent, Alexander W., et al. "Exploiting the power of homomorphic encryption for data sharing in the cloud." 2013 IEEE International Conference on Cloud Computing Technology and Science. IEEE, 2013.