International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0803201

Vol. 8 issue 3 Jun-Sep

Power of Homomorphic Encryption in Secure Data Processing

Muhammad Asif Ibrahim[1], Syed Khuram Hassan[2] and Maham Akhtar

[1]Department of Mathematics, The University of Lahore, Lahore.
[2] Institute of Quality and Technology Management, University of the Punjab, Lahore, Pakistan.
Corresponding author: khuramshah6515@gmail.com
Received: Jul 18, 2024; **Accepted:** Jul 30, 2024; **Published:** Sep 12, 2024

ABSTRACT

Homomorphic encryption is a form of encryption that allows computations to be performed on encrypted data without first having to decrypt it. This paper presents a detailed discussion of HE, a critical component in the protection of data in today's technology-driven environment. First, homomorphic encryption and its terminology will be introduced and then development process from the beginning to the present state will be discussed. Different classes of homomorphic encryption and analysis of internal workings and architecture of homomorphic encryption will be discussed. The usefulness of this technology in ensuring privacy in sensitive areas is discussed, as well as the limitations that may hinder the technology's advancement, including computation intensity and data growth. The paper also reasserts the massive application of homomorphic encryption in data security and privacy, stressing the need to continue the advancement to overcome existing drawbacks and enhance the application of the technique. While moving vast distances within the digital arena, the optimization of homomorphic encryption remains the guiding light to our freedom and privacy online.

Keywords: Encryption, data, digital, homomorphic, protection

1. INTRODUCTION

In this digital era where data and information serve as bargaining chips for malicious actors on platforms like the dark web, the need to protect these is increasing evermore as they have

formed an integral part of our lives. Data can literally be called the 'new gold' in this era. It is our property, one which encroaches deep upon our privacy and could significantly impact our lives if placed in the wrong hands. Our personal information, such as our name, gender, age, the websites we visit, and the words we search for in search engines like Google, etc., are all used to gauge our preferences and generate an online profile that is sold to advertisers [1]. While many companies may claim that they do not do this or that we consent to this operation once we agree to the 'Terms of Agreement,' the problem persists that we, as consumers, have limited control over what personal data is extracted, where it is sent and what is done with it. Even if we ignore the subject of the company whose services we are using, managing our data according to terms beneficial to them, there exists a chance that a malicious threat actor might hack the data placed on their servers and then auction it off to the black-market websites present on the dark web [2].

Many means have been adopted to counter security breaches, each with its respective pros and cons. The use of specific techniques, security software, and devices such as antivirus, firewalls, proxy servers, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), operating system hardening, frequent updating, use of VPN, Two Factor authentication, and backing up timely are some measures adopted for securing data. To protect data is to ensure that it remains confidential, maintains its integrity, is readily accessible when needed, and authentic, knows who created and manipulated it, and ensures its existence to avoid any claim of disputation by any party. This is in line with the extended CIA triad in Information Security: to maintain Confidentiality, Integrity, Availability, authenticity, accountability, and non-repudiation. These principles are fundamental to protecting data and maintaining trust in digital systems. They help ensure that data remains secure, reliable, and verifiable, which is crucial in today's digital age.

2. SECURE DATA PROCESSING

One may adopt the approach of hardening defenses around the data to be protected; for example, think of a fort many kilometers tall with only one entrance at the bottom, whose keys are in possession of a few. The fort is surrounded by a deep moat, which can only be passed with the help of a draw bridge. This is a layering approach in which one must overcome different obstacles to reach the target. This corresponds to having your OS hardened, antivirus installed, system up to date, and firewall in place. The firewall is your first line of defense, just like the moat. The other approach is to make the target incomprehensible to unauthorized personnel. This involves using the technique of encryption. The objective of encryption is to protect the secrecy of data during both communication and storage, as noted by Caroline Fontaine et al. [1]. Even if it falls into the hands of an actor with malicious intent, the data would be of no use if it makes no sense.

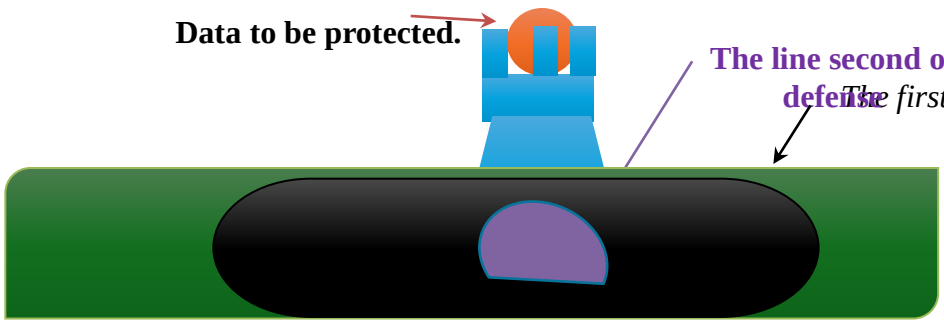

Figure 1: Analogy -A layered defense

But even with encrypted data, there are some concerns. For example, consider a scenario in which a party needs access to certain information placed on a server of another company to carry out tasks [3]. But a server contains more than just this information; it can contain sensitive, more personalized data. Hence, the concept of Homomorphic encryption comes into play. Computations are performed on encrypted data, and only the result is decrypted and sent back to the requestor. For consistency, the outcome after decryption must match the original computed value when applied to the initial data, as asserted by Caroline Fontaine et al. [1].

2.1. Homomorphic Encryption

The word 'Homomorphic' stems from Greek origin, comprising of 'homos' meaning 'same' and 'morphe' meaning 'shape.' In the field of abstract algebra, homomorphism is characterized as a mapping that retains all algebraic structures from the domain to the codomain within an algebraic set, as described by Abbas Acar et al. [2]. The map is simply a function, i.e., an operation, that takes the inputs from the set of domains and outputs an element in the range (e.g., addition, multiplication). In the cryptography field, homomorphic encryption is used as an encryption type. Homomorphic Encryption (HE) represents a type of encryption that enables a third party, such as a cloud provider or service, to carry out specific calculations on encrypted data, maintaining the function's properties and the data's encrypted format, as indicated by Abbas Acar et al. [2]. Homomorphic encryption is similar to public key cryptography, i.e., asymmetric encryption, in that it utilizes more than one key but differs slightly. The concept shall be explained in more detail in the fifth section.

2.2. Encryption and its types

To encrypt something is to render it incomprehensible. This is done by performing specific steps or mathematical calculations on a given text and converting it into a puzzling mystery. The text upon which these operations are performed is referred to as 'plaintext' while the output is called 'ciphertext.' One of the very first forms of encryption was seen in 58 BC by the famous Roman General Julius Caesar, who used it as a secret means of

communication in his military. Called 'Caesar Cipher' after the renowned general, the Caesar Cipher is a form of substitution cipher where plaintext units are substituted with ciphertext following a predefined system. In this cipher, the alphabetic characters are shifted to a set number of positions in the alphabet, as defined by the 'key.' For instance, with a key of 3, the word 'HELLO' is encrypted to 'KHOOR' by shifting each letter three places forward in the alphabet.

2.2.1. Symmetric Encryption

The term “symmetric” implies that the same key is used for both encrypting and decrypting data. Therefore, it is necessary for both the sender and the receiver to concur on a shared key prior to initiating any secure communication, as mentioned by Abbas Acar et al. [2]. This means only the concerned parties have knowledge of the key. However, this key might be leaked during a transfer, i.e., when the two parties communicate and agree upon a key, this conversation might be intercepted.

2.2.2. Asymmetric Encryption

Public key cryptography was developed primarily to address the issue of secure key exchange and to ensure authenticity. One of the significant benefits of symmetric vital systems is that it eliminates the necessity for the parties involved in the communication to have prior knowledge of each other, enabling secure encrypted exchanges, as highlighted by Nigel Smart. Asymmetric encryption employs a pair of keys: a private key, which remains with the sender, and a public key, which is openly distributed. A message encrypted with the private key necessitates its corresponding public key for decryption, and the same principle applies in reverse [2].

2.3. Lattice-based cryptography

Lattice-based cryptography, a variant of post-quantum cryptography, is predicated on the difficulty of solving specific lattice theory problems. A lattice is essentially a structured, repetitive pattern of points in a spatial arrangement. In cryptographic applications, these points are often depicted as vectors within a highly-dimensional space. Due to its discrete nature, there is a definable smallest element, aside from the zero vector, which is trivially the smallest by default. Many of the complex problems in computing, particularly in cryptography, are reducible to the task of finding the minor nonzero vector in a lattice, as stated by Nigel Smart [3]. Since it is a complex problem, it makes it difficult for an attacker to solve it to get the key.

Lattice-based cryptography has several advantages. They are gaining attention for their quantum resistance, positioning them as a viable substitute for existing public-key systems such as RSA and ECC, which are vulnerable to quantum computing attacks. Additionally, these cryptosystems offer advantageous features, including support for fully homomorphic encryption, enabling the execution of computations on data while it remains encrypted.

In two dimensions, you can think of a lattice as a grid of points on a piece of graph paper. Each point on the grid is an integer coordinate (x, y), where x and y

are both integers. The points are evenly spaced along both the x and y axes. A 3D lattice can be visualized like a cube, where each vertex of the cube represents a point in the Lattice. If you imagine stacking these cubes in a regular, repeating pattern along the x, y, and z axes, you would get a 3D lattice [4].

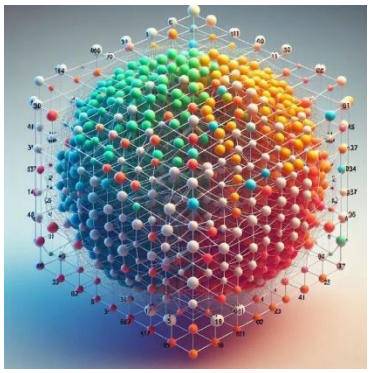

Figure 2: 3D Lattice

Each point in the Lattice has integer coordinates (x, y, z) and is evenly spaced along all three axes. The mathematical properties of lattices allow us to create encryption schemes that are currently unbreakable, even with the most powerful computers.

2.4. CVP and SVP Problems

The challenging issues linked with lattices, such as the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP), play a crucial role in the robustness of encryption methods. For instance, within specific lattice-based encryption frameworks, the secret key is often a concise vector within the lattice structure. Decrypting a message requires solving the SVP to uncover this succinct vector. However, pinpointing the shortest vector within a lattice of high dimensions is a task of significant computational complexity, rendering the decryption process exceedingly difficult for an adversary without knowledge of the secret key. Similarly, the CVP can be used in encryption schemes where the message is encoded as a point near the Lattice. To decrypt the message, you need to find the closest lattice point to this encoded point. Again, this is a complex problem, but it helps ensure the security of the encryption [5].

2.5. Certain types of attacks

Whatever encryption scheme is used, its goal is to be unbreakable. Hence, it should be able to withstand any attacks. Homomorphic Authenticated Encryption (HAE) is a cryptographic protocol that merges the functionalities of homomorphic encryption with those of authenticated encryption. It enables the execution of computations on encrypted data while simultaneously ensuring the integrity and authenticity of both the data and the computations,

all without the necessity of decrypting the data at any point.

In the context of encryption, a robust Homomorphic Authenticated Encryption (HAE) scheme must achieve indistinguishability under chosen plaintext attacks (IND-CPA) and ideally under chosen ciphertext attacks (IND-CCA), as defined by the standard find-then-guess scenarios according to Jeongsu Kim and colleagues [5]. Additionally, when functioning as an authentication mechanism, a secure HAE scheme is expected to be strongly unforgeable under both chosen plaintext attacks (SUF-CPA) and chosen ciphertext attacks (SUF-CCA), as detailed by Jeongsu Kim et al. [5]. These attacks are briefly explained below.

2.5.1. IND-CPA (Indistinguishability under Chosen Plaintext Attack)

This is an attribute of an encryption scheme whereby an attacker cannot gain any information about the plaintext if he/she is given two different ciphertexts. In other words, an adversary cannot distinguish between the two encrypted forms of two chosen plaintext messages.

2.5.2. IND-CCA (Indistinguishability under Ciphertext Attack)

This can be considered a more robust security measure in which an attacker cannot distinguish between the encrypted form of two chosen plaintexts even though he is allowed to decrypt more ciphertexts. This is to guarantee that even under more complicated scenarios, the encryption scheme will remain concealed.

2.5.3. SUF-CPA – Strongly Unforgeable under Chosen Plaintext Attack

This characteristic of a digital signature protocol guarantees that even though an attacker has several signatures for the message, they cannot generate a new signature for a message they did not sign. It is crucial for security measures to help prevent the creation of new signatures without permission.

2.5.4. SUF-CCA (Strongly Unforgeable under Chosen Ciphertext attack)

This is a more robust security notion where, based on the signatures that are placed on a number of ciphertexts, an attacker cannot forge a valid signature on a new ciphertext.

2.5.5. Lattice Reduction

Lattice reduction attempts to convert a given lattice into a reasonable basis, which is pretty short and orthogonal. In the GGH cryptosystem framework, the selection of the public key from a “bad” lattice base and the secret key from a “good” lattice base is strategic. This approach is based on the premise that for lattices with a known “good” base, problems like the Closest Vector Problem (CVP) and the Shortest Vector Problem (SVP) can be resolved efficiently in polynomial time, as explained by Abbas Acar et al. [2].

3. PROGRESSION

Homomorphic Encryption has developed a lot since its early days. Homomorphic Encryption has been an area of continuing research and development. In the course of the development of cryptography, a range of homomorphic encryption derivatives, including partially homomorphic encryption, somewhat homomorphic encryption, and fully homomorphic encryption, has been identified. Every variant is characterized by a specific set of supported operations and overall operational effectiveness. At the moment, homomorphic encryption is one of the hot topics for investigation, especially as a means of protecting privacy in the contexts of cloud computing and secure multi-party computations. It is also used for cases where partial disclosure of data is not possible, but the data needs to be processed. Being used as one of the critical components of contemporary cryptography, homomorphic encryption enables the execution of operations on encrypted data, thus promoting the development of the field of data analysis and ensuring privacy.

3.1. Recent Developments

Wenju Xu and associates [6] have described Kumar and his colleagues' recent proposal of a novel noiseless FHE scheme using the Euler theorem. This proposed FHE scheme is quite unique for its efficiency in encryption, decryption, and homomorphic operations. However, it has been noted that the authors failed to present any proof of security or even a comprehensive security evaluation of their proposed scheme.

3.1. Certain Properties of Homomorphic Encryption

The properties of homomorphic encryption, such as Semantic Security or IND-CPA Security, Compactness, and Efficient Decryption, are important aspects of how it works. These characteristics are essential for their security, effectiveness, and feasibility. They facilitate the ability to carry out calculations on encrypted data, preserving confidentiality and avoiding the need for substantial computational power.

3.1.1. Semantic Security or IND-CPA Security

A homomorphic encryption framework is deemed secure when no potential attacker can determine (with more than a 50% probability) whether a specific ciphertext corresponds to the encryption of two distinct messages. To achieve this, the encryption process must be varied, ensuring that separate encryptions of an identical message appear dissimilar, as described by Melissa Chase et al. [7]. This means that the encryption is so strong that even if someone has the encrypted message, they cannot guess anything about the original message better than a random guess.

3.1.2. Compactness

This means that no matter the number of calculations that are made on the encrypted data, its volume does not change, thus, efficiency. An evaluator is capable of performing any number of supported evaluation functions and coming up with a ciphertext within the

ciphertext space, regardless of the complexity of the evaluated functions [8].

3.1.3. Efficient Decryption

The efficient decryption property of a homomorphic encryption scheme ensures that the time it takes to decrypt does not vary based on the complexity of the functions that were applied to the ciphertexts. In other words, regardless of the operations performed on the encrypted data, the decryption process remains consistently swift. This means that when performing the decryption of the encrypted data, the process is fast and cannot be determined by the number of calculations that were performed on the data. This makes it possible for you to be able to retrieve your original data quickly, as said by Melissa Chase et al. [7], irrespective of what was done to it when it was encrypted.

4. APPLICATIONS OF HOMOMORPHIC ENCRYPTION

Homomorphic encryption is pivotal in fields such as genomics, healthcare, national security, and education, as David Archer and colleagues emphasize [9]. Furthermore, Kundan Munjal and others have thoroughly examined and underscored its impact on the healthcare sector in a systematic review [10].

4.1. Healthcare

The data owners are Hospitals or Health Care Providers. The service latency is dependent on the cloud computing resources, and the data volume is large (patient health records). The data is add-only, and the technical issues involve privacy and data security. The application of HE is possible now, with the main reason being the protection of sensitive medical information. The cost is expected to be borne by the healthcare providers. The cloud computing revolution has led to a demand for outsourcing applications. Users engage with the service by transferring their data to the cloud, where it undergoes processing, and they subsequently retrieve the processed results. This process is highly beneficial for the users; however, it also leaves their sensitive data vulnerable to exposure by third-party cloud service providers. Traditional encryption methods necessitate decrypting the data into its unencrypted state for computations, which poses a risk of disclosing sensitive medical information. Homomorphic Encryption (HE), as described by David Archer et al. [9] and Kundan Munjal et al. [10], offers a solution by enabling computations on data. At the same time, it remains encrypted, ensuring that only the encrypted form of the data is exposed to the service providers.

4.2. Genomics

Medical facilities, as the proprietors of data, can leverage homomorphic encryption to upload various genomic datasets to the cloud securely. This enables the delivery of tailored medical treatments, enhancing patient health and welfare. The expenses for these services are anticipated to be covered by health insurance providers [11].

4.3. National Security

In the event of a vehicular mishap necessitating the intervention of the city's emergency services, such as the Police, Fire Department, and several ambulances, the city's cloud infrastructure could promptly activate a server. This server would dispatch information solicitations to pertinent municipal divisions for instance, Police, Fire, Ambulance, and Transportation to allocate resources from each sector and devise optimal pathways from the accident location to appropriate medical facilities. The execution of these tasks demands diverse computational operations [12].

4.4. Education

The information employed in forecasting the likelihood of student dropouts is confidential and delicate. As such, encryption measures are essential to prevent any potential data breaches. Nonetheless, simply encrypting data while stored or during transfer is not adequate, as significant risks of data exposure persist throughout the processing phase [13].

4.5. E-Voting

Homomorphic encryption can increase the level of security and non-tampering of electronic voting systems. It allows the voter to verify the correct count of their votes and, at the same time, keep the voter's choice a secret. This approach could help increase the confidence of the people in the electoral process as well as the sanctity of election results [14].

4.6. Cloud-Based Systems

Homomorphic encryption is a very effective technique that provides security to the data stored in cloud systems. Since cloud storage means that data is stored on a server that many users can access, the data can be changed or even deleted. Homomorphic encryption safeguards this data since it can be processed in an encrypted form, and thus, the data is never revealed. This aids in the protection against unauthorized access and modification and can improve the users' confidence in the cloud storage services [13].

4.7. Machine Learning

Homomorphic encryption plays the role of an essential tool in enhancing the confidentiality and security of machine learning algorithms. It makes it possible to perform calculations on data that are encrypted to train and predict the next phase of a machine learning model without having to decode the actual data. It is most useful in situations where data security is of the essence, such as in the medical or finance fields. The model is able to learn from the data and provide accurate predictions, and at the same time, the data is protected. Therefore, homomorphic encryption is a powerful tool for machine learning that is focused on protecting data privacy [15].

4. CONCLUSION

Therefore, this paper has given a detailed analysis of homomorphic encryption, a significant component in a modern digital world where Cybersecurity is vital. In this paper, we have discussed the meaning of homomorphic encryption, its

background, categories, architecture, and practical use cases. However, homomorphic encryption is full of problems at the moment, such as computational costs and data blow-ups; however, it has great potential in the future. Thus, with the further enhancement of homomorphic encryption, the protection of our data in the age of digital development will be guaranteed, and our privacy will be preserved. Further research will be directed to the elimination of the present drawbacks and widening the usage of homomorphic encryption.

REFERENCES

[1] C. Fontaine and F. Garland, "A Survey of Homomorphic Encryption for Nonspecialists," *EURASIP Journal on Information Security*, vol.7, 2007.

[2] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys*, vol. 51, no. 4, 2018.

[3] N. P. Smart, Cryptography: An Introduction. New York, NY, USA: McGraw Hill, 2002.

[4] V. Lyubashevsky, C. Peikert, and O. Regev, "On Ideal Lattices and Learning with Errors Over Rings," *Journal of the ACM (JACM)*, vol. 60, no. 6, 2013.

[5] J. Kim and A. Yun, "Secure Fully Homomorphic Authenticated Encryption," *IEEE Access*, vol. 9, pp. 107279-107297, 2021.

[6] W. Xu, Y. Zhan, Z. Wang, B. Wang, and Y. Ping, "Attack and Improvement on a Symmetric Fully Homomorphic Encryption Scheme," *IEEE Access*, vol. 7, pp. 68373-68379, 2019.

[7] M. Chase, "Security of Homomorphic Encryption", *Proceedings of the Homomorphic Encryption Standardization Workshop, Microsoft Research, Redmond*, 2017.

[8] G. Tu, W. Liu, T. Zhou, X. Yang, and F. Zhang, "Concise and Efficient Multi-Identity Fully Homomorphic Encryption Scheme," *IEEE Access*, 2024.

[9] D. Archer, "Applications of Homomorphic Encryption," *Technical Report*, 2017.

[10] K. Munjal and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex and Intelligent Systems*, vol. 9, pp. 3759-3786, 2023.

[11] I. Mustafa, H. Mustafa, A. T. Azar, S. Aslam, S. M. Mohsin, M. B. Qureshi, and N. Ashraf, "Noise Free Fully Homomorphic Encryption Scheme Over Non-Associative Algebra," *IEEE Access*, vol. 8, pp. 136524-136536, 2020.

[12] M. Ogburn, C. Turner, and P. Dahal, "Homomorphic Encryption," in *Complex Adaptive Systems, Publication 3*, C. H. Dagli, Ed., pp. 502-509. 2013.

[13] M. Li, "Leveled Certificateless Fully Homomorphic Encryption Schemes from Learning with Errors," *IEEE Access*, vol. 8, pp. 26749-26763, 2020.

[14] T. Shen, F. Wang, K. Chen, K. Wang, and B. Li, "Efficient Leveled (Multi) Identity-Based Fully

Homomorphic Encryption Schemes," *IEEE Access*, vol. 7, pp. 84764-84775, 2019.

[15] R. L. Rivest, L. Adleman, and M. Dertouzos, ''On data banks and privacy homomorphisms,'' *Foundations Secure Computation*, vol. 4, no. 11, pp. 169-180, 1978.