Evolution of Homomorphic Encryption

George Asante
Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development
Department of Information Technology Education

James Ben Hayfron-Acquah
Kwame Nkrumah University of Science and Technology
Department of Computer Science

Michael Asante
Kwame Nkrumah University of Science and Technology
Department of Computer Science

ABSTRACT

Data encryption has been a major tool for ensuring data confidentiality and integrity. Data encryption is mostly applied to data *at rest*, and data *in motion*. For data *in use*, encrypted data is usually decrypted by the user before performing any operation on the data. The security of such data is usually compromised when the user is not trusted. In cloud computing, for instance, where computing services are delegated, decrypting data by a third party before performing any operation is not appropriate. The operation must be performed on the data in its encrypted form; hence, the need for homomorphic encryption. This paper presents the evolution of homomorphic encryption schemes as a basis for research into modern homomorphic encryption schemes. The various stages identified are Privacy homomorphism, Partial homomorphism, Somewhat homomorphism and fully homomorphic encryption.

Keywords

Data security; Encryption; homomorphism; homomorphic encryption; Evolution.

1. INTRODUCTION

The use of information technology comes with a lot of risks such as loss of data [4], loss of privacy [15], identity theft [25], denial of service [20], and repudiation [22]. These risks pose the following threats: Interruption, Interception, Modification, and Fabrication of data [6].

Encryption helps to mitigate Interception and Modification threats. Data encryption converts original data into a form that will be difficult for a third party to understand without the use of decryption key. The user usually needs to decrypt the encrypted data, using a decryption key, in order to understand the encrypted data [21].

Most encryption schemes allow third party to decrypt data for computation to be performed on such data. There are always security challenges whenever a third party is involved in computation of data. An encryption scheme that allows for computation on encrypted data (without decryption) is much preferred [2]. Such schemes are referred to as homomorphic encryption schemes. Homomorphic encryption allows computation on encrypted data and yields same result that would have been yielded when the computation is done on unencrypted form of the data.

There has been much research on homomorphic encryption but much has not been done on the evolution of modern homomorphic encryption schemes. This paper provides a brief historical account of the gradual development of homomorphic encryption since its inception in 1978. Since the introduction of homomorphic encryption scheme, it has gone through the following stages: Privacy homomorphism, Partial homomorphism, Somewhat homomorphism and fully homomorphic encryption schemes.

The paper proceeds as follows: In the next section, the concept of homomorphic encryption is explained. Next, an account of its evolution is provided. The review is then concluded with a discussion on the way forward.

2. CONCEPT OF HOMOMORPHIC ENCRYPTION

Data becomes vulnerable when in use. The reason is that, most encrypted data needs to be decrypted before use. Only homomorphic encryption allows you to use encrypted data without necessarily decrypting it.

Homomorphism was derived from the greek word “homos” and “morphe”. “homos” means “same” and “morphe” means “shape” [17]. So homomorphic could be considered as objects having “same shape or form”. If encryption is considered as homomorphic, then it means whatever you could do to plaintext, you could do same to the encrypted text and get the same results. Hayward and Chiang in 2015 [18] defined homomorphic encryption as “an encryption scheme which allows for computations on encrypted data and obtains an encrypted result which when decrypted, produces the same result as computations on the original (plaintext) data”.

For instance, if two plaintexts p1 and p2 are encrypted to enc (p1) and enc (p2) respectively, then if p1 + p2 = pr andenc(p1) + enc (p2) = er , then pr = dec (er). This idea is depicted in figure 1.

Homomorphic encryption enables one to delegate computation but still ensures data confidentiality. Here, it should be noted that, the computation could be done by a person at one end and the decryption by a person at the other end, without either of them able to find the value of the individual numbers [23]. For this reason, the homomorphic encryption scheme should present a particular structure that allows for delegation of computation.

Unlike other encryption schemes that have three algorithms, homomorphic encryption schemes have four algorithms, namely: Key generation algorithm, Encryption algorithm, Decryption algorithm and Evaluation algorithm [26].

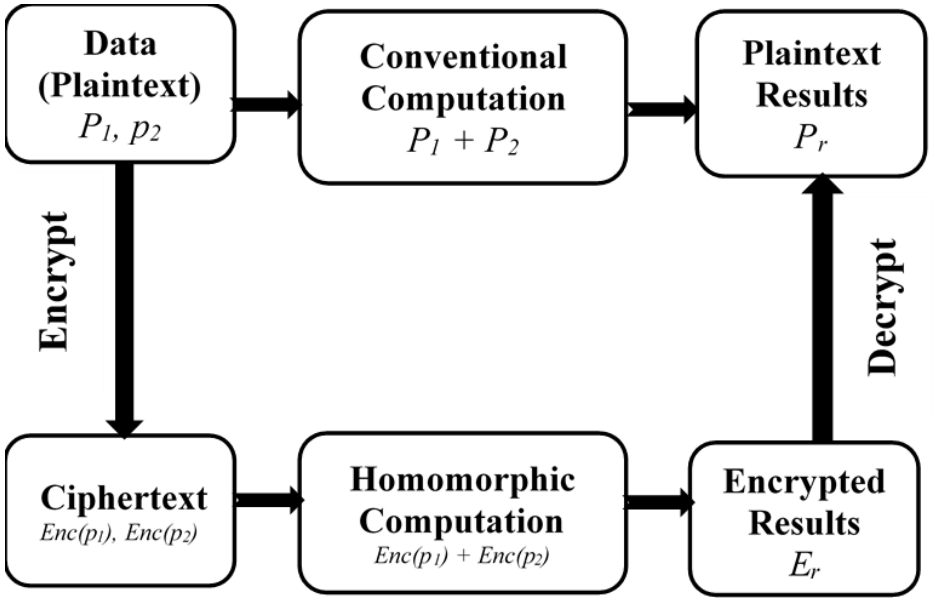

Fig 1: Homomorphic Encryption model

3. EVOLUTION OF HOMOMORPHIC ENCRYPTION SCHEMES

Ever since homomorphic encryption was introduced by Riverst, Adleman, and Dertouzous in 1978 [24], there has been tremendous improvement in the scheme. The improvement is in either speed, security, efficiency or simplicity of the scheme. Riverst et al [24] created encryption algorithm that permits computation on encrypted data without prior decryption of ciphertext. They called that scheme as Privacy homomorphism. Brickell and Yacobi (1987) as cited in [10], identified some security flaws in the proposed scheme of Riverst et al.

In 1978, Riverst, Shamir, and Adleman proposed a public-key cryptosystem (RSA scheme) which uses ideas from number theory [16]. Their scheme was considered as multiplicative homomorphism since it allows only multiplication operations. The scheme was named after them, using their initials: RSA. RSA is very popular asymmetric encryption scheme. The scheme could be used to ensure secrecy and provide digital signatures.

In 1982, Goldwasser and Micali (1982) as cited in [19], proposed an encryption scheme that is able to encrypt one bit. The scheme was referred to as Additive homomorphic scheme since it allows only addition operations.

ElGamal (1985) also proposed another multiplicative homomorphic encryption scheme. Pallier (1999) proposed another additive homomorphic encryption scheme. The scheme was probabilistic asymmetric algorithm based on composite residuosity classes [16].

These schemes are referred to as Partial homomorphic encryption schemes [27]; they either perform addition or multiplication operations but not both.

In 2005, BGN scheme was proposed. The scheme is based on pairings over elliptic curves. The scheme supports both additions and multiplication operations but with only depth-one multiplication [4].

In 2009, Gentry [11] proposed the first encryption scheme that supports both addition and multiplication. Gentry's multiplication goes beyond depth-one as compared to [4]. Gentry progressively created a fully homomorphic encryption scheme. He first created what is called "somewhat homomorphic scheme". This scheme allows only a limited number of homomorphic computations on ciphertexts. This is because homomorphic computations introduce what is called "noise". This noise grows as the number of computations increase. When the noise reaches a threshold, decryption gives undesired results. To resolve this issue, Gentry constantly reduced the noise through what is called refresh. Such a process of refreshing ciphertext to reduce noise is referred to as bootstrapping [27]. Giving two refreshed ciphertexts, one can apply again the homomorphic operations that were not possible with original ciphertexts. Finally, Gentry "squashed" the decryption circuit [27]. The squashing aims to simplify the decryption circuit [27]. Gupta and Sharma [16] referred to Gentry's blueprint scheme as an algebraically homomorphic scheme.

Gentry [12] later pointed out that their scheme in [11] was not quite bootstrappable. So in [12], they describe a public key encryption system using ideal lattices. In 2010, Stehle and Steinfeld as cited in [27] and Smart and Vercauteren (2010) as cited in [16], made improvements in Gentry's scheme.

Gentry and Halevi [13] later came out with a new scheme to do away with squashing but the construction still relies on ideal lattices. This was considered as the first implementation of Gentry's blueprint [27]. Again, Gentry et al [14], presented a variant of Gentry's scheme that is simpler, easier to describe and implement.

A different fully homomorphic encryption scheme was proposed by Dijk, Gentry, Halevi and Vaikuntanathan (DGHV). The scheme was based on integers instead of utilizing ideal lattices [26]. The scheme was conceptually simple as compared to original Gentry's Scheme.

In 2013, Cheon et al [7] extended the fully homomorphic encryption scheme over the integers of [26] to batch fully homomorphic encryption. Their scheme supports encrypting and homomorphically processing a vector of plaintext bits as a single ciphertext [7].

Gupta and Sharma [16] proposed a fully homomorphic encryption scheme with symmetric keys. Their scheme operations were based on matrix. The scheme maps the operations on integers to operations on matrix. Sharma et al

claim their scheme is computationally light, efficient, multi-hop, circuit-private and can be deployed in multiple user environment [16].

In 2016, Dasgupta and Pal [8] designed a polynomial ring based symmetric homomorphic encryption scheme. That is, polynomial over ring of integers. They started with a somewhat Homomorphic encryption and extended to fully homomorphic encryption through refreshing [8].

In 2017, [9] introduced a fully homomorphic scheme. They claim their scheme is efficient and based on a new mathematical structure that is noise free [8]. Their scheme uses the ring of Lipschitz's quaternions and permits computation on data encrypted under a symmetric key.

In 2018, [17] designed a fully homomorphic encryption by prime modular operation. They claim their scheme is faster than fully homomorphic encryption over the integer (DGHV Scheme) and simple fully homomorphic encryption scheme available in cloud computing (SDC Scheme). Hamad and Sagheer scheme does not convert a message to a binary format as DGHV and SDC schemes do. Hamad and Sagheer [17] scheme encrypts message character by character by using a prime secret without converting to binary format. This makes it faster.

In July 2020, [3] developed a fully homomorphic encryption scheme based on decomposition ring. Their scheme allows parallel computation and that makes it faster.

4. CONCLUSION

Homomorphic encryption has evolved from privacy homomorphism, partial homomorphism, somewhat homomorphism to fully homomorphism. Each subsequent phase brought about improvement in the homomorphism. The improvement is in either speed, security, efficiency or simplicity of the scheme. Homomorphic encryption is very expensive comparatively for real-life applications. This challenge makes fully homomorphic encryption impractical on many platforms. Extensive study, therefore, needs to be done on the efficiency of the scheme in various application areas. With the security of the fully homomorphic encryption schemes, the unbounded fully homomorphic encryption schemes are Indisguishable under Chosen Ciphertext Attacks (IND-CCA) [1]. Extensive study, therefore, needs to be done on fully homomorphic encryption schemes to improve security. Acar et al in [1] opined that fully homomorphic encryption with unbounded number of users are still not feasible. Extensive research also needs to be conducted on fully homomorphic encryption with unlimited number of users.

5. 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] Al Badawi, A. Q. A., Polyakov, Y., Aung, K. M. M., Veeravalli, B., &Rohloff, K. (2019). Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme. *IEEE Transactions on Emerging Topics in Computing*.

[3] Arita, S., &Handa, S. (2020). Fully Homomorphic Encryption Scheme Based on Decomposition Ring. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *103*(1), 195-211.

[4] Badrinarayanan, N., Krishnan, S., &Andireddi, K. (2019). *U.S. Patent No. 10,496,291*. Washington, DC: U.S. Patent and Trademark Office.

[5] Boneh, D., Gentry, C., Halevi, S., Wang, F., & Wu, D. J. (2013, June). Private database queries using somewhat homomorphic encryption. In *International Conference on Applied Cryptography and Network Security* (pp. 102-118). Springer, Berlin, Heidelberg.

[6] Cernov, A. M. (2018). Security in Computer Networks. International Journal of Information Security and Cybercrime (IJISC), 7(1), 45-52.

[7] Cheon, J. H., Coron, J. S., Kim, J., Lee, M. S., Lepoint, T., Tibouchi, M., & Yun, A. (2013, May). Batch fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 315-335). Springer, Berlin, Heidelberg.

[8] Dasgupta, S., & Pal, S. K. (2016). Design of a polynomial ring based symmetric homomorphic encryption scheme. *Perspectives in Science*, *8*, 692-695.

[9] El-Yahyaoui, A., & El Kettani, M. D. E. C. (2019). An Efficient Fully Homomorphic Encryption Scheme, International Journal of Network Security, Vol.21, No.1, PP.91-99, Jan. 2019 (DOI: 10.6633/IJNS.201901 21(1).11)

[10] Fontaine, C., &Galand, F. (2007). A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, *2007*, 1-10.

[11] Gentry, C. (2009a). *A fully homomorphic encryption scheme*. Stanford university.

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

[13] Gentry, C., &Halevi, S. (2011, May). Implementing gentry's fully-homomorphic encryption scheme. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 129-148). Springer, Berlin, Heidelberg.

[14] Gentry, C., Halevi, S., & Smart, N. P. (2012, May). Better bootstrapping in fully homomorphic encryption. In *International Workshop on Public Key Cryptography* (pp. 1-16). Springer, Berlin, Heidelberg.

[15] Gross, H. (2017). Privacy and autonomy. In *Privacy and Personality* (pp. 169-181). Routledge.

[16] Gupta, C. P., & Sharma, I. (2013, October). A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds. In *2013 Fourth International Conference on the Network of the Future (NoF)* (pp. 1-4). IEEE.

[17] Hamad, S.S &Sagheer, A. (2018). Design of Fully Homomorphic Encryption by Prime Modular Operation. Telfor Journal. 10. 118-122. 10.5937/telfor1802118S.

[18] Hayward, R. & Chiang, C. (2015): Parallelizing fully homomorphic encryption for a cloud environment. Journal of Applied Research and Technology v13; 245-252.

[19] Jabbar, I., &Najim, S. (2016). Using fully homomorphic encryption to secure cloud computing. *Internet Things Cloud Comput*, *4*(2), 13.

[20] Jamal, T., Haider, Z., Butt, S. A., &Chohan, A. (2018). Denial of service attack in cooperative networks. *arXiv preprint arXiv:1810.11070*.

[21] Kahate, A. (2013). *Cryptography and network security*. Tata McGraw-Hill Education.

[22] Nurhaida, I., Ramayanti, D., &Riesaputra, R. (2017). Digital signature & encryption implementation for increasing authentication, integrity, security and data non-repudiation. *vol*, *4*, 4-14.

[23] Ogburn, M., Turner, C., &Dahal, P. (2013). Homomorphic encryption. *Procedia Computer Science*, *20*, 502-509.

[24] Rivest, R. L., Adleman, L., &Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of secure computation*, *4*(11), 169-180.

[25] van de Weijer, S. G., Leukfeldt, R., &Bernasco, W. (2019). Determinants of reporting cybercrime: A comparison between identity theft, consumer fraud, and hacking. *European Journal of Criminology*, *16*(4), 486-508.

[26] Van Dijk, M., Gentry, C., Halevi, S., &Vaikuntanathan, V. (2010, May). Fully homomorphic encryption over the integers. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 24-43). Springer, Berlin, Heidelberg.

[27] Zhang, L., Zheng, Y., &Kantoa, R. (2016, June). A review of homomorphic encryption and its applications. In *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications* (pp. 97-106).