IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-6, Issue-10, Oct 2018 E-ISSN: 2347-2693

Homomorphic Encryption for Big Data Security: A Survey

Galal A. AL-Rummana[1*], G. N. Shende[2]

[1]School of Computational Sciences, S.R.T.M University, Nanded, India
[2]Indira Gandhi Collage, Cidco-Nanded, India

[*]*Corresponding Author: galal300z@gmail.com, Tel.: +917410796725*

Available online at: www.ijcseonline.org

Accepted: 11/Oct/2018, Published: 31/Oct/2018

Abstract— The size of data generated every second has crossing the boundary of usual data size as a result of the rapid growth and spread of communication technology. This notable increase is of great importance and has gained the scholars' interest. In other words, the increasing of data make the current ear an era of big data. Nowadays, one of the vital challenges is to get big data secured. Cryptography is an important technique that provides a high data security in many environments and applications. Homomorphic Encryption (HE), a special direction of cryptography, can address such security issues in big data environment. This paper concerns with the HE schemes which can play a vital role in securing big data environment. Therefore, big data concepts and characteristics are reviewed in the current paper along with full description of HE schemes, covering the HE types and illustrating their mechanisms for securing big data. In addition, the current paper offers some interpretations on the base of some security features along with big data security model.

Keywords—Big Data, Big Data Security, Homomorphic Encryption, PHE, SWHE, FHE.

I. INTRODUCTION

Big data is a new and important issue in the field of academic research as well as in the field of industry research [1]. Due to its high value, variety, volume, velocity, variability, and veracity, big data is distinguished from the other traditional data [2]. Big data analytics provide new ways for businesses and government to analyze unstructured data. Nowadays, Big data is one of the most common topics in IT industry. It is going to play an important role in the future. Big data changes the way that data is managed and used. Some of the applications are commonly used in healthcare, traffic management, banking, retail, education and so on. Organizations have become more flexible and more open. However, new types of data will give new challenges as well [3].

Academic scholars and companies define big data equally the same with little difference to be noted. For example, Chen et. al define it as datasets that could not be perceived, acquired, managed, and processed by traditional IT and software or hardware tools within a tolerable time [4]. Similarly, Apache Hadoop in 2010 defined big data as "data sets which could not be captured, managed, and processed by general computers within an acceptable scope" [4]. Big data can be defined as a collection of data sets, which is very large in size and complex as well. Generally, the data size is Petabyte and Exabyte. It exceeds the processing capability of conventional data management systems and software techniques. However, big data comes with big values. Traditional database systems are not able to capture, store and analyze this large amount of data. As the internet is developing, an amount of big data continues to be developed [3]. Therefore, the big data environment must be secured against different attacks such as spamming attacks, Search Poisoning, Botnets, Denial of service attack (DOS), phishing, Malware, and website threats as well as the leakage of confidential data should be protected [5].

There are specific characteristics that can affect the security of big data which can be represented as follows:

1. Volume:

Big data implies an enormous amount of data generated by machines, networks, and systems like social media etc. Analyzing such large amount of data is not an easy task.

2. Variety:

Variety refers to various types of structured and unstructured data originating from miscellaneous sources. Traditionally, data was stored structurally in database files or excel sheets, but today data comes in the form of images, audio files, videos, photos etc. This variety of unstructured data causes several problems in storing and analyzing data.

3. Velocity:

Velocity refers to the enormous rate at which the data is produced. This flow is massive and continuous. Analyzing real-time data at such a great speed is again a challenge to big data systems.

4. Veracity:

The data contains a lot of biases, noise, and abnormalities and thus, has to be cleaned before processing. Veracity refers to the resistance provided by big data to keep the "dirty data" from accumulating.

5. Volatility:

Validity refers to the point after which the data becomes futile for further analysis and hence, must be truncated.

6. Validity:

Validity tackles the issue of data trustworthiness and data precision. While analyzing the colossal amount of data, it is important that the data be correct and accurate [6][7]. However, the fig. (1) shows a brief description of big data characteristic [8].

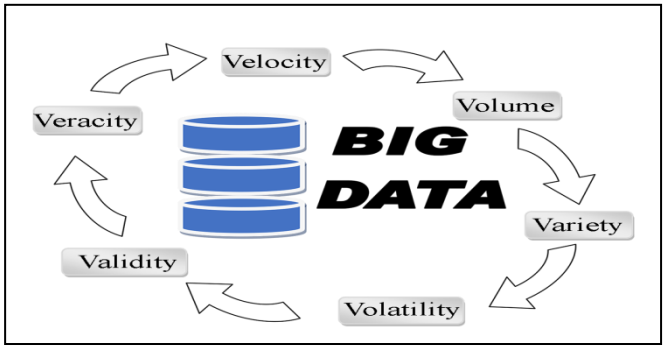

Figure 1. The Characteristics of Big data.

In past, the data was very small and the cryptographic algorithms were designed with the appropriate size where the cryptographic systems depend on sharing a key. This key may be a public key or a private key between peers involved in an encrypted message exchange. This key is shared with the user and the service provider who has exclusive rights to the data. Even when the keys are not shared, the encrypted content is shared with a third party that does not necessarily need to access the content. In fact, the technology requires a special type of encryption scheme. These concerns allow any third party to work on encrypted data without decrypting it in advance.

The value and the security issue that big data brings to us are equally striking. For a consideration of security, the data are generally encrypted before storing in a cloud database. Thus, it is quite difficult for unauthorized users to understand the encrypted data and get the corresponding plain text back. However, encrypting data in a traditional way makes it impossible to leverage data's value, because the data needs to perform the decryption first whenever it is to be processed [9]. Homomorphic Encryption (HE) is a proper solution to deal with such concerns due to its ability to allow any third party to run the encrypted data without being decrypted in advance.

The rest of the current paper is organized four parts. The second part provides a brief review about HE. The third part discusses the different types of HE, and the fourth is devoted for the discussion of interpretation, and the last part is for the conclusion.

II. THE HOMOMORPHIC ENCRYPTION (HE)

The ancient Greek language, the term "όμός" (homos) was used to mean the "same" and "μορφή" (morphe) was used to indicate "form" or "shape" [10]. Then, the term homomorphism was coined and used in different areas.

In abstract algebra, the term of Homomorphism is defined as a map i.e. an operation or function, preserving all the algebraic structures between the inputs from the set of domain and outputs an element in the range of an algebraic set [11].

In information security, the homomorphic is a cryptographic method used for data encryption.

HE is an encryption scheme which allows a third party (e.g., cloud, service provider) to perform certain computable functions on the encrypted data while preserving the features of the function and format of the encrypted data, in order to match the result of operations that performed on the plaintexts after being decrypted [12]. The properties of HE can be illustrated through the following definition.

Definition: An encryption scheme is called homomorphic over an operation " $*$ " if it supports the following equation:

$$E(m_1) * (m_2) = E(m_1 * m_2), \forall m_1, m_2 \in M$$

Where E is the encryption algorithm and M is the set of all possible messages.

The addition and multiplication operations are sufficient to create an encryption scheme allowing the homomorphic evaluation of arbitrary function, due to their functionally complete sets over finite sets. Particularly, any boolean circuit can be represented using only XOR (addition) and AND (multiplication) gates. HE scheme can use the same key for encryption and decryption purpose in a symmetric form. Moreover, various keys can also be used in HE. The HE scheme is mostly characterized by four procedures:

Encryption, Decryption, Eval, and KeyGen. These four procedures are elaborated as follows:

A. Encryption procedure

It is a procedure of converting a message from readable form into unreadable form. In other words, it is a process used to convert a plaintext into a ciphertext taking into account that without a secret key, no unauthorized users cannot access the original message. Encryption does not itself prevent interference but denies intelligible content to a potential interceptor. The master aim of encryption is to protect the confidentiality of digital data stored and transmitted via the internet, computer systems and other communication channels. However, the encryption algorithms play a vital role in the security assurance of IT systems and communications as they can provide Authentication, Integrity, Confidentiality, and Nonrepudiation.

B. Decryption procedure

It is the process of taking encrypted or encoded text then, converting it back into text that you can read and understand. Basically, decryption is the inverse process of encryption.

C. Eval procedure

It is a HE specific operation, which takes ciphertexts as input and outputs a ciphertext corresponding to a functioned plaintext. Eval performs the function f ( ) over the ciphertexts (c1; c2) without seeing the messages (m1; m2).

D. Key Generation procedure

In cryptography, key generation is the process of generating secret keys; regardless, the process is data encryption or decryption. It generates a secret and public key pair for the asymmetric version of HE or a single key for the symmetric version. Key gen, Enc, and Dec are actually similar to the ones used in conventional encryption schemes.

III. TYPE OF HOMOMORPHIC ENCRYPTION

The Homomorphic encryption schemes can be categorized into three types as shown in fig. (2) In respect to the number of operations allowed on the data encryption the categorized schemes are enlisted below:

- **Partially Homomorphic Encryption (PHE)**

Allows only one type of operation with an unlimited number of times; this is to say, no bound on the number of uses).

- **Somewhat Homomorphic Encryption (SWHE)**

Allows certain types of operations with a limited number of times.

- **Fully Homomorphic Encryption (FHE)**

Allows an unlimited number of operations with an unlimited number of times. In the following a full description of Homomorphic encryption types are provided:

*A. **Partially Homomorphic Encryption (PHE) Schemes***

A PHE is called partially due to the fact that it takes only one operation either multiplication or addition. The following algorithms are useful examples of PHE.

- **RSA Scheme**

Rivest, Shamir, and Adleman Introduced RSA after the invention of public key cryptography by Diffie Helman [13]. RSA is an early example of Partially Homomorphic Encryption and the first feasible achievement of the public key cryptosystem. The security of the RSA cryptosystem depends on the hardness of factoring problem of the product of two large prime numbers. RSA is only homomorphic over multiplication. Hence, it does not allow the homomorphic addition of ciphertexts [14].

- **Goldwasser-Micali (GM) Scheme**

In 1982, *Shafi Goldwasser* and *Silvio Micali* proposed the first probabilistic public key encryption scheme [15]. The GM cryptosystem is dependent on the hardness of quadratic residuosity problem [16]. GM does not allow the homomorphic multiplication of ciphertexts. Hence, it is only homomorphic over addition for binary numbers.

- **El-Gamal Scheme**

In 1985, *Taher Elgamal* has proposed a new public key encryption scheme this scheme is the improved version of the original Diffie-Hellman Key Exchange algorithm [17], which depends on the hardness of certain problems in discrete logarithm [18]. It is frequently used in hybrid encryption systems to encrypt the secret key of the asymmetric encryption system. El-Gamal cryptosystem is only homomorphic over multiplication. Hence, it does not allow the homomorphic addition of ciphertext.

- **Benaloh Scheme**

In 1994, *Benaloh* proposed an extension of the GM Cryptosystem by improving it to encrypt the message as a block instead of bit by bit, Benaloh's proposal was dependent on a higher residuosity problem (x^n). It is the generalization of quadratic residuosity problems x^2 that is used for the GM cryptosystem. The Benaloh cryptosystem is additively homomorphic [19].

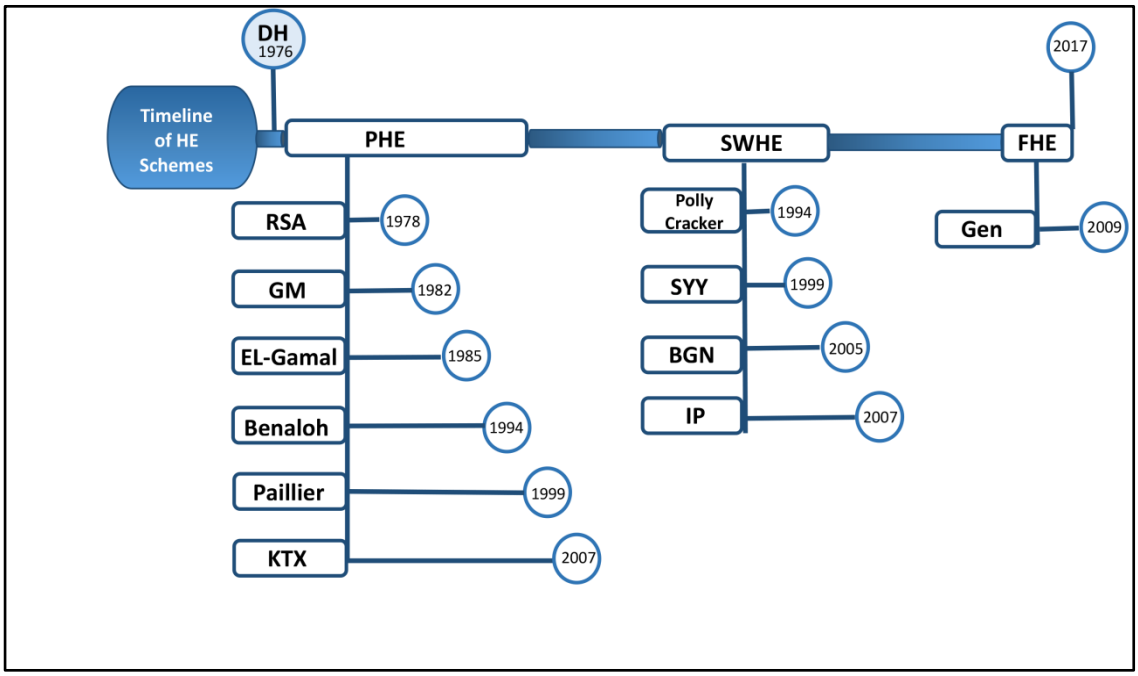

Figure 2. Time Line of HE Scheme.

- **Paillier Scheme**

Another novel probabilistic encryption scheme has been proposed by *paillier* in 1999, its novelty is dependent on composite residuosity problem and it is closely similar to quadratic and higher residuosity problems that are used in Benaloh and GM cryptosystems [20].

- **Kawchi (KTX) Scheme**

In 2007, *kawachi* et. al. suggested an additively homomorphic encryption scheme over a large cyclic group, which depends on the hardness of underlying lattice problems. The homomorphic property of their scheme can be called pseudohomomorphic.

However, Pseudohomomorphism is an algebraic property and still allows homomorphic operations on ciphertext, whereas, the decryption of the homomorphically operated ciphertext works with a small decryption error [21]. Homomorphic properties of well-known PHE schemes are briefly summarized in Table 1.

B. Somewhat Homomorphic Encryption (SWHE) Schemes

The three basic operations of standard encipherment scheme are encryption, decryption, and key generation. In homomorphic scheme, an evaluation operation must be added. The main purpose of evaluation operation is to apply the operation over ciphertexts and obtain the ciphertext of the result.

Moreover, homomorphic encryption scheme is called an evaluation scheme that has correct decryption and correct evaluation. There is no requirement for compactness. So, the ciphertexts can increase substantially in length with each homomorphic operation [22].

Table (1): Homomorphic properties of well-known PHE schemes**.**

Scheme	**Homomorphic Operation**		***Description***
	Add	***Mult***	
RSA 1978		√	hardness of factoring problem of the product of two large prime numbers.
GM 1982	√		hardness of quadratic residuosity problem
EL-Gamal 1985		√	hardness of certain problems in discrete logarithm
Benaloh 1994	√		higher residuosity problem
Paillier 1999	√		composite residuosity problem
KTX 2007	√		hardness of underlying lattice problems

The major SWHE schemes, which are used as a stepping-stone to the first plausible FHE schemes are listed as follows:

- **Polly Cracker Scheme**

Polly Cracker scheme is one of the first SWHE that applies two operations, i.e., multiplication and addition over the

ciphertexts. Yet the size of the ciphertext grows exponentially with the homomorphic operation, and the multiplication operation is extremely expensive [23]. However, more efficient variants are proposed but all of them are later shown vulnerable to attacks, so they are either insecure or impractical [24]. Recently, a Polly Cracker has introduced with Noise cryptosystem, where the homomorphic addition operations do not increase the ciphertext size while the multiplications square does [25].

- **Sander, Young, and Yung (SYY) Scheme**

SYY describes first SWHE scheme over a semi-group. It is another idea of evaluating operations on encrypted data that is realized over different sets. This idea requires fewer properties than a group. This scheme polynomially supports many ANDing of ciphertexts with one OR/NOT gate. Therefore, the ciphertext size increases due to the constant multiplication with each OR/NOT gate evaluation. However, the limits of the circuit depth evaluation are increased [26].

- **Boneh- Goh- Nissim (BGN) Scheme**

Boneh et al. have Introduce one of the most significant steps toward an FHE scheme. In BGN, 2-DNF5 formulas on ciphertext have been evaluated and resulting in supporting an arbitrary number of additions and one multiplication by keeping the ciphertext size constant [27].

- **Yuval Ishai and AnatPaskin (IP) Scheme**

In 2007, *Yuval* et al. have expanded the set of branching programs which are directed acyclic graphs in such a way that every node has two outgoing edges with labeled binary 0 and 1. Furthermore, they proposed a public key encryption scheme by evaluating the branching programs on the encrypted data. In addition, *Melchor* et al. proposed a generic construction method to obtain a chained encryption scheme that allowing the homomorphic evaluation of constant depth circuit over ciphertext. The chained encryption scheme is obtained from well-known encryption schemes with some homomorphic properties [28]. For example, they showed how to obtain a combination of BGN and KTX. Hence, the resulting combined scheme allows arbitrary additions and two multiplications. As mentioned, BGN and KTX have shown how this procedure is applied to the scheme in allowing a limited number of homomorphic additions, to obtain a scheme which allows an arbitrary number of multiplications as well. However, in multiplication, ciphertext size grows exponentially while it is constant in a homomorphic addition. In brief, table (2) shows the summary of somewhat well-known SWHE schemes with respect to homomorphic operations and the cipher text size.

Table (2): Summary of Somewhat well-known SWHE Schemes

Scheme	Homomorphic Operation	Ciphertext Size
Polly cracker 1994	Mult and Add	Grows exponentially
SYY 1999	Polynomially many AND & one OR/NOT	Grows exponentially
BGN 2005	Arbitrary	Constant
IP 2007	Arbitrary	doesn't depend on the size of the function

C. Fully Homomorphic Encryption (FHE) Schemes

The last two type of homomorphic encryption includes only one or two operations of encryption. In 2009, *Gentry* describes the first feasible construction of a fully homomorphic cryptosystem in his seminal Ph.D. thesis to a long-term open problem, which is obtaining an FHE scheme.

Fully Homomorphic Encryption (FHE) scheme once allows an unlimited number of evaluation operations on the encrypted data and resulting output is within the ciphertext space. He also proposed a general framework to obtain an FHE scheme. Hence, many researchers have attempted to design a secure practical FHE scheme after Gentry's contribution. So, along with his contribution, similar attempts can be categorized into four main types as shown in Fig 3, and can be illustrated as follows [29] :

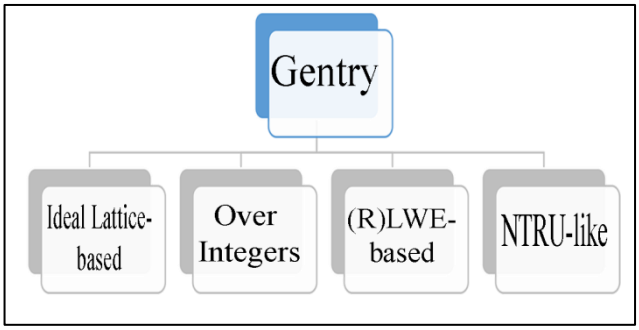

Figure 3. The Main Four Categories of Gentry.

- **Ideal Lattice-based FHE Schemes**

Gentry started his breakthrough work from SWHE scheme based on ideal lattices. *Gentry* encrypted the message by embedding noise using double layer instead of one layer idea in other cryptosystems.

Although Gentry's proposed ideal lattice-based FHE scheme is very favorable, it also has a lot of restriction. For example, some of its advanced mathematical concepts

make it complex and computational cost in terms of applicability in real life and hard to implement. So, many new schemes and optimization have followed his work in order to address such aforementioned restriction. As mentioned, a SWHE scheme can evaluate the ciphertext homomorphically for only a limited number of operations. After a certain threshold, the decryption function fails to recover the message from the ciphertext correctly. The amount of noise in the ciphertext must be decreased to transform the noisy ciphertext into a proper ciphertext. *Gentry* used genius blueprint methods called squashing and bootstrapping to obtain a ciphertext which allows a number of homomorphic operations to be performed on it. These processes can evaluate unlimited operations on the ciphertexts which make the scheme fully homomorphic [29].

A brief description of the obtained methods is shown as follows:

A. Squashing

Gentry's bootstrapping technique is allowed only for the decryption algorithms with small depth. Therefore, he used some "tweaks" to reduce the decryption algorithm's complexity [29].

B. *Bootstrapping*

Bootstrapping is basically "recrypting" procedure to get a "fresh" ciphertext from the noisy ciphertext corresponding to the same plaintext. A scheme is called bootstrappable if it can evaluate its own decryption algorithm circuit [Gentry 2009]. Firstly, the ciphertext is transformed into a bootstrappable ciphertext using squashing. Then, by applying a bootstrapping procedure, one gets a "fresh" ciphertext [30].

In addition, *Gentry* used ideals and rings without lattices to design the homomorphic encryption scheme, where an ideal is a property preserving subset of the rings as even numbers. Then, each ideal used in his scheme was represented by the lattices [29].

- **FHE Schemes over Integers**

A year after Gentry's original scheme, *Van Dijk* et al. presented another scheme which suggests Gentry's ingenious bootstrapping method in order to obtain an FHE scheme [31]. This scheme is over integers and the hardness of the scheme is based on the Approximate-Greatest Common Divisor problems.

In fact, this scheme was a symmetric version of the HE and transforming into an asymmetric HE scheme. Nowadays, this scheme is considered a public key encryption scheme since it uses different keys for encryption and decryption. Conceptually, this proposed is very simple, meantime, it is costly in term of computations. Thus, this scheme is not very efficient. As such, some early attempts directly tried to improve its efficiency.

- **Learning With Error (LWE)**

LWE was introduced by *Oded Regev* as an extension of "learning from parity with error" problem it is considered as one of the hardest problems to solve in practical time for even post-quantum algorithms [32]. Since then, it becomes one of the most attractive and promising topics for post-quantum cryptology with its relatively small ciphertext size. *Lyubashevsky* et al. suggested another significant improvement on the LWE problem which might lead to new applications by introducing ring-LWE (RLWE) problem [33]. The RLWE problem is an algebraic variant of LWE, which is more efficient for practical applications with strong security proofs. They proved that the RLWE problems are reducible to worst-case problems on ideal lattices, which is hard for polynomial-time quantum algorithms. In 2011, *Brakerski* and *Vaikuntanathan* have made an important step towards to a practical FHE scheme established a new SWHE scheme based on Ring-Learning with Error (RLWE) to take advantage of the efficiency feature of RLWE [34]. On the other hand, it can be used in LWE and RLWE problem as the hardness assumption of an FHE scheme. However, this is to say that the performance of RLWE is better than LWE.

- **NTRU-Like FHE Schemes**

NTRU encryption scheme is one of earliest attempts based on lattice problems. To get an applicable and practical FHE scheme, just one of crucial steps is taken by showing the construction of an FHE scheme from NTRU Encrypt, which is an old encryption scheme proposed by *Hoffstein* et al. [35]. In 1998 especially, how to obtain a multi-key FHE from the NTRUEncrypt (called NTRU) was shown by *López-Alt* et al. in 2012 [36]. Since the security of this scheme is improved, standardization issues, efficiency, and easy implementation, attract researchers' interest again. At the same time, *López-Alt* et al. used the NTRU encryption scheme to obtain a practical FHE with three differences. Firstly, the set from which the noise is sampled is changed from a deterministic set to a distribution. Secondly, the modification introduced by *Stehlé* and *Steinfeld* in 2011 [37] which makes the scheme more secure, is used and in

the last, the parameters are chosen to allow fully homomorphism.

IV. DISCUSSION AND INTERPRETATION

Encrypting data in the traditional ways is useless. It does not reach at leverage data's value this is because the need of the data to be decrypted after being processed [38]. However, the FHE makes it possible to reach at the leverage data's value, since it allows applying arbitrary number of operation additions and multiplications to be applied on the encrypted data, without the need to decrypt them. Using FHE help to gaining the same result of applying plain data and encrypt data. Nevertheless, according to the point of view of *Benzekki* et al. FHE, which nearly achieved the fully homomorphic scheme in cloud computing along with distributed servers, is slow for a practical system [39]. Therefore, FHE is still valuable for further research.

However, in order to apply FHE in big data environment, such scenarios and models should be structured. Our discussion will be based on the assumption of *Wang* et al. They assumed that the storage service, computing service and processing service were provided by a cloud provider [40]. The entities and the operating principles involved in our model are as follows, and as shown in Fig. 4:

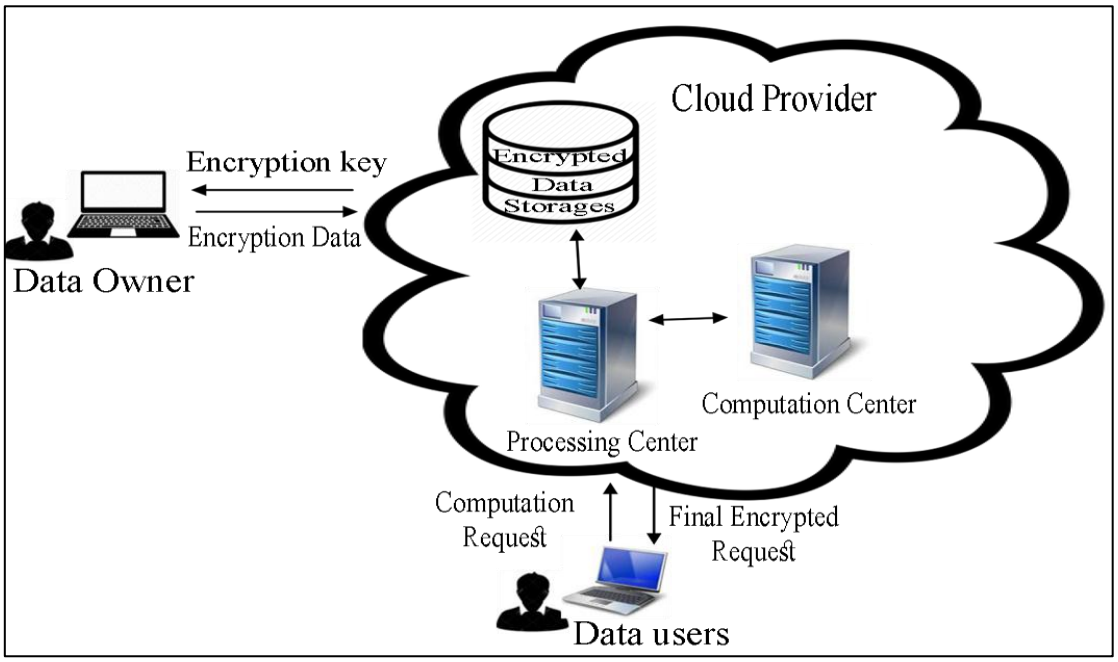

Figure 4. A System Model Using FHE Scheme.

- Data owner: Data owner is usually the enterprise or organization who possess the raw data. It receives the public key from processing center and encrypts the private data. Then it sends the encrypted data to the cloud database. It has only encryption functionality.
- Storage: Storage is a cloud database that is responsible for storing encrypted data.
- Processing center: The processing center produces the public key and secret key. It has functionality of key generation. Additionally, it analyses the user's request and tells computation center which data requires certain computation.
- Computation center: Computation center is responsible for performing the operations requested by the data user. It receives ciphertexts and formula and returns final encrypted result to the processing center. It has an evaluate functionality.
- Data user: Data user has intention to perform operations on encrypted data. It obtains the final result from processing center without accessing any intermediate result. It has decrypt functionality.

V. CONCLUSION

In this paper, a descriptive review of big data and its security through the homomorphic encryption schemes have been presented. The classification of HE schemes, interpretation of some security features, and explanation of an existing model of big data security have been the central issues of this paper. The paper concerns encryption data for security purpose and making data more confidential. It can be noted that in the previous studies one should first decrypt the data for operating it. But now, it becomes much easier. One can add new mathematic operation without the process of decryption. Therefore, conditionality of data is much more secured. Using homomorphic encryption technique, one can be assured that data cannot be stolen. The three types of homomorphic encryption have also been discussed briefly in this paper. The last type (FHE) is one of the primary technologies that can be implemented to protect and emphasize the application value of huge data. Therefore, it is valuable to do further research.

REFERENCES

[1] D. S. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in big data," in 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), Dec 2015, pp. 202–207.

[2] Mohammed, A. F., Humbe, V. T., &Chowhan, S. S. (2016, February). A review of big data environment and its related technologies. In Information Communication and Embedded Systems (ICICES), 2016 International Conference on (pp. 1-5). IEEE

[3] Tarekegn, G. B., &Munaye, Y. Y. (2016). "Big Data: Security Issues, Challenges, and Future Scope". International Journal of Computer Engineering & Technology (IJCET), 7(0976-6367), 12-24.

[4] Chen, M., Mao, S., & Liu, Y. (2014). "Big data: A survey". Mobile Networks and Applications, 19(2), 171-209.

[5] Duhan, B., & Singh, D. (2018). Big Data and its Security Issues. 828-831

[6] Jhaveri, M., Jhaveri, D., &Shekokar, N. (2015). "Big Data Authentication and Authorization using SRP Protocol". International Journal of Computer Applications, 130(1), 26-29.

[7] Muthulakshmi, P., & Udhayapriya, S. (2018). A SURVEY ON BIG DATA ISSUES AND CHALLENGES. 1238-1244.

[8] Moura, J., &Serrão, C. (2016). "Security and privacy issues of big data". arXiv preprint arXiv:1601.06206.

[9] M. Beunardeau, A. Connolly, R. Geraud, and D. Naccache, "Fully homomorphic encryption: Computations with a blindfold," IEEE Security Privacy, vol. 14, no. 1, pp. 63–67, Jan 2016.

[10] Henry George Liddell and Robert Scott. 1896. "An intermediate Greek-English lexicon": founded upon the seventh edition of Liddell and Scott's Greek-English lexicon. Harper & Brothers.

[11] Malik, D. S., Mordeson, J. N., & Sen, M. K. (2007). MTH 581-582 "Introduction to Abstract Algebra".

[12] Yi, X., Paulet, R., &Bertino, E. (2014). "Homomorphic encryption and applications" (Vol. 3). Cham: Springer.

[13] Rivest, R. L., Shamir, A., &Adleman, L. (1978). "A method for obtaining digital signatures and public-key cryptosystems". Communications of the ACM, 21(2), 120-126.

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

[15] Goldwasser, S., &Micali, S. (1982, May). "Probabilistic encryption & how to play mental poker keeping secret all" partial information. In Proceedings of the fourteenth annual ACM symposium on Theory of computing (pp. 365-377). ACM

[16] Kaliski, B. (2011). Quadratic Residuosity Problem. In Encyclopedia of Cryptography and Security (pp. 1003-1003). Springer US.

[17] ElGamal, T. (1985). "A public key cryptosystem and a signature scheme based on discrete logarithms". IEEE transactions on information theory, 31(4), 469-472.

[18] Kevin, S. M. (1990). The discrete logarithm problem. Cryptology and computational number theory, 42, 49.

[19] Benaloh, J. (1994, May). Dense probabilistic encryption. In Proceedings of the workshop on selected areas of cryptography(pp. 120-128).

[20] Paillier, P. (1999, May). "Public-key cryptosystems based on composite degree residuosity classes". In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 223-238). Springer, Berlin, Heidelberg.

[21] Kawachi, A., Tanaka, K., &Xagawa, K. (2007, April). "Multi-bit cryptosystems based on lattice problems". In International Workshop on Public Key Cryptography (pp. 315-329). Springer, Berlin, Heidelberg.

[22] Pisa, P. S., Abdalla, M., & Duarte, O. C. M. B. (2012, December). "Somewhat homomorphic encryption scheme for arithmetic operations on large integers". In Global Information Infrastructure and Networking Symposium (GIIS), 2012 (pp. 1-8). IEEE.

[23] Fellows, M., &Koblitz, N. (1994). Combinatorial cryptosystems galore!. Contemporary Mathematics, 168, 51-51.

[24] Le, V. L. (2003). "Polly two-a public key cryptosystem based on Polly cracker" (Doctoral dissertation, Ruhr University Bochum, Germany).

[25] Albrecht, M. R., Farshim, P., Faugere, J. C., & Perret, L. (2011, December). "Polly cracker, revisited". In International Conference on the Theory and Application of Cryptology and Information Security (pp. 179-196). Springer, Berlin, Heidelberg.

[26] Sander, T., Young, A., & Yung, M. (1999). Non-interactive cryptocomputing for NC/SUP 1. In Foundations of Computer Science, 1999. 40th Annual Symposium on (pp. 554-566). IEEE.

[27] Boneh, D., Goh, E. J., & Nissim, K. (2005, February). Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography Conference (pp. 325-341). Springer, Berlin, Heidelberg.

[28] Melchor, C. A., Gaborit, P., &Herranz, J. (2010, August). Additively homomorphic encryption with d-operand multiplications. In Annual Cryptology Conference (pp. 138-154). Springer, Berlin, Heidelberg.

[29] Gentry, C. (2009). A fully homomorphic encryption scheme. Stanford University.

[30] Gentry, C. (2010, August). Toward basing fully homomorphic encryption on worst-case hardness. In Annual Cryptology Conference (pp. 116-137). Springer, Berlin, Heidelberg.

[31] 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.

[32] Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM), 56(6), 34.
[33] Lyubashevsky, V., Peikert, C., &Regev, O. (2013). On ideal lattices and learning with errors over rings. Journal of the ACM (JACM), 60(6), 43.
[34] Brakerski, Z., &Vaikuntanathan, V. (2011, August). Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Annual cryptology conference (pp. 505-524). Springer, Berlin, Heidelberg.
[35] Hoffstein, J., Pipher, J., & Silverman, J. H. (1998, June). NTRU: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium (pp. 267-288). Springer, Berlin, Heidelberg
[36] López-Alt, A., Tromer, E., &Vaikuntanathan, V. (2012, May). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Proceedings of the forty-fourth annual ACM symposium on Theory of computing (pp. 1219-1234). ACM.
[37] Stehlé, D., & Steinfeld, R. (2011, May). Making NTRU as secure as worst-case problems over ideal lattices. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 27-47). Springer, Berlin, Heidelberg.
[38] Beunardeau, M., Connolly, A., Geraud, R., &Naccache, D. (2016). Fully homomorphic encryption: Computations with a blindfold. IEEE Security & Privacy, 14(1), 63-67.
[39] Benzekki, K., El Fergougui, A., &Elbelrhiti, E. A. (2016). A secure cloud computing architecture using homomorphic encryption. International Journal of Advanced Computer Science and Applications (IJACSA), 7(2), 293-298.
[40] Wang, D., Guo, B., Shen, Y., Cheng, S. J., & Lin, Y. H. (2017, March). A faster fully homomorphic encryption scheme in big data. In Big Data Analysis (ICBDA), 2017 IEEE 2nd International Conference on (pp. 345-349). IEEE.

Authors Profile

Galal A. AL-Rummana

Currently, he is a Ph.D candidate in the School of Computational Sciences, at Swami Ramanand Teerth Marathwada University, Nanded. He has received his M.Sc. degree in Computer Networking from the School of Computational Sciences at Swami Ramanand Teerth Marathwada University, in 2017, Nanded, India. He has received his B.E degree in Computer Engineering form the faculty of Computer Science & Engineering , at Hodeidah University, Yemen, 2014. He is currently working on Big Data Security.

G. N. Shinde

Currently, he is Principal of Indra Gandhi College, Nanded. Erlier he was Pro-Vice Chancellor, SRTM University, Nanded, Maharashtra, INDIA. He has received "Ideal State Teacher Award" from Government of Maharashtra, India for 2008-09 and "Best Principal Award" for 2009-2010 from S.R.T.M. University, Nanded, Maharashtra. He has received M. Sc. & Ph.D. degree from Dr. B.A.M. University, Aurangabad. He has awarded Benjonji Jalnawala award for securing highest marks at B.Sc. Thirteen research scholars were awarded Ph.D. degree under his guidance. He has published more than 75 papers in the International Journals and presented more than 50 papers in International Conferences. He was the Chairperson for F-9 session of "International Conference on Computational and Experimental Science & Engineering" (ICCES2008) which was held at Honolulu, U.S.A & Development and Application of Web 2.0 Technology for Education Purpose session of "International Conference on Multimedia & ICT in Education (mICT2009)", April 24, 2009, (Hall 1), Lisbon (Portugal) and "Wave Propagation and Wave Interaction with media" Progress In Electromagnetic Research Symposium (PIERS), March 25, 2010, Session 3A4 (Room D) Xian, China. In his account one book is published, which is reference book for different courses. He is also member of different academic & professional bodies such as IAENG (Hon Kong), ANAS (Jordan). He is in reviewer panel for different Journals such as IEEE (Transactions on Neural Networks), International Journal of Physical Sciences (U.S.A.), Journal of Electromagnetic Waves and Applications (JEMWA, U.S.A.). His abroad Visit includes U.S.A., Thailand, Portugal, Germany, Switzerland, Italy, Vatican City, Monaco, France, Maldives, Sri Lanka, U. K., Scotland, China and New Zealand. He was Chairman of Grievances Committee and member of Management Council & Senate of S.R.T.M. University, Nanded, INDIA. His research interest includes Filters, Wireless Sensor Network System, Image processing and Multimedia analysis and retrieval system.

