March-April 2020
ISSN: 0193-4120 Page No. 3119 - 3128

Application Specific Statistical Analysis of Fully Homomorphic Crypto Systems for Cybersecurity

Dhananjay M. Dumbere, *Department of Computer Science and Engineering, Rajiv Gandhi college of Engineering and Technology, Chandrapur, India.*
Dr. Asha Ambhaikar, *Department of Computer Science & Engineering, Kalinga University, Raipur, Raipur, India.*

Article Info
Volume 83
Page Number: 3119 - 3128
Publication Issue:
March - April 2020

Abstract:
Cryptography has taken its roots since ages and has been evolving with each generation. From early day crypto-systems which were based on substitutions, to modern day elliptic-curve based systems, cryptography has come a long way. Generally, cryptographic systems are non-mutable in nature. This means that once the cryptographic data is generated, then in order to get perfect re-construction, we must not change it. This has been a major challenge with software-security-architects. Because, this immutability renders the data non-shareable. For instance, if Amazon wants to perform some complex computations on some confidential data, then the encryption and decryption keys for that data had to be shared with the analysis team. This team, would first decrypt the data, perform the computations, and then encrypt it back and send it over to Amazon. Which lead to chances of a major security breach from the analysis team. In order to avoid such breaches, homo-morphic systems were designed. Using homo-morphic encryption, security-engineers can encrypt data, and allow third parties to perform all kind of computational analysis on that data. Once the computations are done, then security-engineers decrypt the data, and obtain the same computations reflected on the original data. In this paper, we analyse various homo-morphic crypto-systems, and compare their performances in order to find which systems are best suited for what kind of applications. We also evaluate these systems, to find out certain advantages and drawbacks of these systems and provide in-depth analytical conclusions about them. This paper concludes with some interesting observations about the said algorithms, and also proposes ways to improve them.

Keywords: *Crypto-systems, homo- morphic, computations, confidential, cybersecurity.*

Article History
Article Received: *24 July 2019*
Revised: *12 September 2019*
Accepted: *15 February 2020*
Publication: *21 March 2020*

1. INTRODUCTION

Study of cryptography and its very many applications has been going on for more than 3 decades now. Initial crypto systems were based on simple manipulations of data, like adding offsets, XORing with keys, etc. But as the computation power of the personal computer increased, there was a linear increase in the complexity of these systems. Generally, crypto systems are classified into public key crypto systems and private key crypto systems. Public key crypto systems use different keys for encryption and decryption, while private key crypto systems use a single key for both encryption and decryption.Both modern day public and private key crypto systems modify the plain text (or the input data to be encrypted). This modification leads to generation of cipher text (or the encrypted data), which is generally in non-readable & non-understandable form. Any mathematical operation performed on the cipher text will make the cipher text unusable for decryption. This property of crypto systems makes them highly versatile, and non-mutable. But, due to this non-mutability, there are some very serious issues which makes these systems non-usable in a noisy environment (by noisy environment, we are referring to wireless communication with a noisy channel). Because even a single bit change in a large encrypted data array,

March-April 2020
ISSN: 0193-4120 Page No. 3119 - 3128

will cause an avalanche effect during decryption. Due to this, the decrypted data will not be same as the original input data, and thus rendering the system unusable.To overcome this issue, researchers are working towards building checks into the algorithm, which will cancel out the noise, and reconstruct an accurate data vector on the decryption side. But this increases the computational complexity of the system and adds some level of redundancy to the encrypted data. Because of an increase in computational complexity the power and speed efficiency of the system reduces. Due to redundancy, the overall system throughput reduces. Thus, there was a need of a different kind of encryption which should be resistive to minor changes in the input data. Rivest proposed Homo-morphic encryption (HE) in the year 1978, with the idea that small changes in the encrypted data, will be reflected in the decrypted data as it is, and the decryption process will generate identifiable data. Hill cipher was the first homo-morphic encryption system [6]which could reflect data addition changed of the encrypted data directly at the decrypted side. Thus, Hill cipher can handle adaptive white gaussian noise channel, which adds random noise to the input data.

RSA is a well-known public-key crypto system, which uses a public key for encryption, and a private key for decryption. The RSA algorithm supports multiplicative homomorphism [6], which means, any multiplication operation performed on the encrypted data will be reflected directly on the decrypted data, and the decryption process will pass. Using RSA, system designers can work with frequency related noises like power line interference, hormonic noise, etc. RSA is sufficiently secure and is used by many legacy systems world-wide. While, RSA and Hill ciphers support full multiplicative and additive homomorphism, some systems like ElGamal encryption only support partial homomorphism [6]. Due to the random nature of encryption key selection, there are possibilities that ElGamal might not support the multiplicative homomorphic property, but in most of the cases it does support the same.

In contrast to ElGamal cryptosystem, the Pallier cryptosystem support multiple homomorphisms. From research [6], it has been observed that the Pallier cryptosystem can support both additive and multiplicative homomorphism. Due to this property, it is the most frequently used single homomorphic algorithm. While single homo-morphic systems produce promising results for single type of mathematical operation. But we need a system which can reflect multiple types of changes in the encrypted data. To fulfil this requirement fully homo-morphic crypto systems are used. These systems can support both multiplicative and additive changes to the encrypted data and can successfully reflect those changes at the decrypted side. The next section describes these algorithms in detail and discusses their performance so that researchers can select between them as per the given application type.

II. Literature review

Over the past 5 years, many researchershave proposed different variations of effective homomorphic implementations. In this section, we will analyse each of them chronologically starting from the latest to the oldest. The work done by Jyun-Neng Ji and Ming-Der Shieh in [1], have optimized the calculations done during homomorphic encryption by aggregating plain text data. They propose an effective scheme to handle both comparison and swapping homomorphic operations. They claim that with the help of aggregation, size of the fully homomorphic encrypted (FHE) data is reduced, and double data rate can be achieved. They also claim that the system can be made at least 2.3 times faster, but that largely depends on the kind of data being encrypted. Their proposed algorithm has been compared with logistic regression based and bitwise FHE systems. During comparison they observed that for smaller file sizes logistic regression algorithm works faster than the proposed aggregation-based method, but as the file size increases the performance of the later system increases linearly. This is because, as file sizes

increases, so does the probability of data repetition. This data repetition is used by the proposed system for aggregation to improve the system performance. The claimed encrypted data sizes are also very small, they claim to have reduce the encrypted file size by more than 50 times, which might not be applicable to all kinds of data and must be evaluated before applying this protocol for real-time applications. This protocol can be used in wireless multimedia networks where bandwidth optimization is needed, along with a lot of comparison and swapping operations. Multimedia networks also require lower data size for communication, and this method does itprecisely.

NingBo Li&others in [2], mention about their research on homomorphic encryption for large amounts of data. Their research is very significant when it comes to bigdata based applications. They propose to optimize the msdExtract algorithm which is used during the FHE decryption phase. This optimization allows for the existing decryption system to increase its scale from Z2 to Zt, thereby adding more security to the system. As per their claims, the system can perform FHE operations on a large size of data and maintain the same security as the learning with errors or LWE system. They have used the FHEW library (W stands for West), for analysis, and evaluated their results on the same. The optimization is done by replacing the original rounding function with msdExtract algorithm which extracts most of the important data from the homomorphic stream. Once the data is extracted, then an aggregation layer is applied to reduce the data size. Finally, a bootstrapping layer is applied which accumulates the data, and further performs dimensionality reduction. There are no quantitative results produced in the text, but they claim to have tested their system on large quantities of plain text data. It is recommended for the researchers to first test this system on dummy data before applying the algorithm for any real time application.

An application of FHE is proposed by C.N.Umadevi and Dr.N.P.Gopalan in [3], wherein they have suggested the use of square matrices in order to achieve high level of security and homomorphic flexibility. In their system, they have also use Smith Normal Form which makes sure that all the properties of a homomorphic system are satisfied. Due to the use of square matrices, the crypto processes of encryption and decryption are performed using symmetric keys. They also claim to be using both cipher text policy for attribute-based encryption (CP-ABE), and key based policy for attribute-based encryption (KP-ABE). Amongst the two methods, the CP-ABE method is much better than KP-ABE method in terms of efficiency, security, access control and collision resistance. Both KP-ABE and CP-ABE based FHE methods used in the paper perform good in terms of third-party outsourcing applications and can be used for real-time implementation.

FHE heavily depends on exclusive sum of product operations or ESOPs. These ESOPs take a lot of processing time and constitute one of the major areas of optimization. In [4], the research proposed by Jheng-Hao Ye, Si-Quan Chen, and Ming-Der Shieh minimizes the computations needed by ESOPsand explores the constraints in it. The optimized ESOP implementation is based on Sierpinski gasket and the triangle rule. These algorithms reduce the degree of ESOP expressions and thereby minimize the requirement of re-encryption operations on the encrypted data. The minimization algorithm works in 4 steps,

- Distance-k Operation for Reducing Maximum Degree
- Solving the matching problem
- Negative Priority and Odd Number Detection
- Post-processing ESOP Minimization

Using these operations, the number of computations is reduced. The authors claim that the overall system complexity is reduced by 10%, and the maximum degree is reduced by almost 5%. Analytically these values are not very high, but when it comes to large amounts of data, then they make a huge difference in the computational speed of the system. This

algorithm can be used in any kind of crypto system which requires lowering of complexity, for example in internet enabled smart devices. The system is highly effective and produces good homomorphic capabilities as well.

While the performance of FHE system is known to be superior than its non-FHE counterparts, but some FHE systems are less secure than others. An analysis done by Mikhail Babenko, Nikolay Chervyakov& others in [5], proves that FHE systems based on residue number systems and secret sharing schemes can be attacked easily. The attacker can get the plain text by just getting n * log(n) * log(k*pn) randomly generated input files. Thus, using these number of files, the attacker can regenerate the secret keys which can decrypt the FHE data. Their method of cryptanalysis is based on RNS and number theory. They also analyse that if the complexity of the secret sharing scheme (SSS) & the RNS is high, then the FHE system is saved from these kinds of attacks. Thus, any researcher who is using SSS like Shamir Sharing with RNS, must take care of the system complexity in order to improve the security of the designed crypto system.

Talking about security of FHE systems, the research done by Min Zhao E & Yang Geng in [6], suggests that Gentry's homomorphic encryption scheme based on ideal lattice has variable security, where the security of this algorithm depends on the computational complexity of bounded distance coding on perfect lattices and the computational complexity of sparse subsets. Another scheme is studied in [6] namely DGHVfully homomorphic encryption scheme based on Integer, where DGHV is short for the scientists Dijk, Gentry, and Halevi who invented it. The DGHV algorithm is based on modular operations over integers for encryption and decryption. This scheme is very effective in terms of security and speed, but is non-implementable on day-to-day systems due to the large length of public key. In contrast, the BGV Scheme Based on RLWE doesn't require any kind of bootstrapping using key exchange and modulo reduction techniques. Instead, the key exchange technology is used to convert the input text into its FHE encrypted form, while the noise from the encrypted data is reduced using modular switching technology. This scheme is very light weight, and has high level of security, thus is suitable for wireless sensor network security systems. The GSW13 Scheme based on approximate eigenvectors is also discussed in [6]. It has lower computational efficiency than other schemes based on RLWE, but it reduces the length of the public key thereby improving space efficiency of the system. This scheme is generally used by crypto systems with low storage requirements that require high level of security. One final study done in [6] about Multi-key fully homomorphic encryption scheme based on NTRU suggests usage of multiple keys for data encryption. This ensures that the data can be decrypted only when all the users provide their keys for decryption. The system is very secure, and its security increases as the number of users for the data in use increase. This is in contrast with most of the systems presented by researchers over the years, and thus is a very widely used FHE algorithm for cloud-based systems. The researchers conclude that the circuit based FHE algorithms must be replaced by algebraic FHE systems, which makes sense because circuits are static and can be copied or hacked, while algebraic systems can change their internal calculations in order to avoid any kind of attacks. Moreover, machine learning must be used for most of these systems in order to improve their computational efficiency.

An interesting work of research which includes quaternions is proposed by Ahmed EL-YAHYAOUI and Mohamed Dafir ECH-CHRIF EL KETTANI in [7]. Their work expands the crypto space from simple numeric to quaternion. This allows the system to process large numbered inputs. While standard systems can process data in the form of 0s and 1s, the system proposed in [7], can use the $\frac{Z}{N^2 * Z}$ space for representing the input data. This increases the size of the secret key from 3*N (required by noise free non-commutative rings based FHE) to

16*N*N. But this increase in complexity brings in a lot of advantages, including but not limited to, reduced computation time, increased file size support, an almost undecryptable security system and ability to encrypt non-standard data like smileys and symbols.

While FHE systems are used for encryption and decryption of text or numeric data, there is very little research done on applying FHE to an entire database, and then retrieving data from the encrypted domain data. The work done by Zhibin Gong, Youan Xiao, Yihong Long and Yanli Yang in [8] performs this task very efficiently. The proposed system can work with unreliable data storage systems and can help to retrieve the data using counting joint index generation algorithm. This algorithm uses upper limit based on dual encoding function, which ensures that both time and space are saved. It utilizes fuzzy retrieval of character data which helps the server to filter out unwanted irrelevant data with high speed and space efficiency. Due to these advanced processing techniques, the delay is reduced by more than 80% than other conventional fuzzy database storage solutions based on FHE, while the filtering efficiency is more than 45% when compared to the same fuzzy FHE database solutions. The algorithm is faster, more efficient and safer as compared to most of the existing FHE database encryption and decryption systems. This algorithm can be used for large database systems like university management systems, enterprise resource planning systems, etc.

While most FHE systems consider security and computational delay as the primary aspects of evaluation, the role of policies in evaluation of access control must be considered for large scale data storage systems like cloud storage. The research done by Yong Ding and Xiumin Li in [9] uses fine-grained flexible access control along with response-to-query mechanism for efficient data retrieval based on simplistic queries. The system also allows easy revocability of access to data without giving any load on the client. The researchers claim that the algorithm is fully shielded from collusion attacks, due to its inherent access control mechanism. The text uses linear secret sharing schemes or LSSS in order to improve the overall security capabilities of the system. Thus, the combination of CP-ABE with FHE and fine-grained access control can be used for large scale cloud deployments and can assist cloud designers with high level of security without compromising on the QoS of the system.

Generally, FHE schemes are built for integral number space, but real-world data can contain both integers and real numbers. The work done by Keke Gai, MeikangQiu, Yujun Li and Xiao-Yang Liu in [10] proposes a real number based FHE system named as FHE-RN. The system is based on dual KP-ABE operation, where the first KP operation is performed on the integer portion of the data, and the other is performed on the floating part of the data. These two data are finally combined in order to obtain the final encrypted data. The decryption process is different for both addition and multiplication-based division and uses different equations for both. Due to a large number of random number combinations needed for deciphering the encrypted data, it is impractical for attackers to gain access to the original data. The error produced by this kind of scheme is of the order of 10^{-14}, thus this scheme is very accurate. Applications of this work include small to moderate scale data systems, like wireless battery powered networks or vehicular networks. The system must not be used for large scale systems due to its inherent complexity, which will tend to reduce the system QoS over the course of data generation.

As mentioned in the introduction section, RSA system possesses multiplicative homomorphism while Paillier possesses additive homomorphism. In order to use the power of both the algorithms, a hybrid homomorphic system is proposed by Xidan Song and Yulin Wang in [11]. The system performs subjective encryption, wherein RSA is applied to data which requires multiplication operations, while Paillier is applied to data which requires additive

operations. The system performs well for small sized data, but as the size of the data increases, the complexity of separating out the different data components also increases. This reduces the overall QoS of the system, and thus it becomes non-optimum to use such a system for large scale data systems.

For large scale data systems, the FHE algorithm proposed by Dan Wang and Bing Guo and others in [12] can be interesting. The system uses pseudo random public keys and replaces the existing linear form of encryption with cubic form in the public key elements. This allows the system to be more effective in encrypting large datasets and improves the data security. The system is proven to be error free by solving the approximate GCD problem. The system is based on DGHV scheme, and thus has high efficiency. Due to the introduction of pseudo random numbers, the key size is reduced to half, and thus the data rate is doubled. This algorithm can be used for practical on demand encryption systems like Voice over IP (VoIP), cloud storage and other multimedia applications. Another research on cloud based FHE system is proposed by Keke Gai andMeikangQiu in [13]. They are using the Kronecker Product (KP) in order to achieve highly optimal FHE design. Here too, there are different algorithms for multiplication and addition-based decryption, thus the system can be highly computationally complex. From the results, it can be observed that this system is shielded against both outside threats and inside threats due to the KP model. The paper doesn't provide any quantitative results, and thus researchers are advised to perform due diligence before implementing the algorithm for any real-world data. Due diligence is more important here because the research claims to have optimum accuracy and efficiency, but there is no comparison provided in the text.

A review of multiple types of HE systems, namely Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (PHE) is done in [14]. Here the researchers Dalia Tourky, Mohamed EIKawkagy and Arabi Keshk have studied different algorithms for performing HE, and they observe that parallel FHE systems which are based on multi-user sharing are the most effective for any practical application. Moreover, the implementations of FHE which involve lattice-based calculations are more effective both in terms of speed and security and must be used if processing power requirements are fulfilled. For lower powered systems PHE or SHE systems can be used if the kind of mathematical operations are limited.

For low powered and high-speed systems, there is a need for light weighted FHE systems. The work in [15] demonstrated by MohdRizuanBaharon, Qi Shi, and David Llewellyn-Jones satisfies this requirement. The system reduces the computation power during encryption and key-generation processes, thereby limiting the security but improving the system performance. This system is a truly FHE system and supports full addition and multiplication capabilities. For moderately secure systems like colleges or household data, where speed is more important than security of data, this system can be utilized, and implemented practically. While for highly secure systems, this algorithm is not recommended. Moreover, there are no practical results which demonstrate the superiority of this algorithm, and thus researchers must first test this algorithm on dummy data, before implementing the same for real-world moderate secure systems.

Apart from addition and multiplication, some cryptosystems also require higher level capabilities like square root and division. The work done by KamalakantaSethi, Amartya Majumdar and PadmalochanBera in [16] proposes a system which can use the parallel computational capabilities of CPUs and GPUs in order to perform these operations. They claim to have tested the data of different sizes on multiple execution threadsand evaluated the performance of the system. According to their analysis, the systems usually converge in performance for more than 8 threads, and thus

March-April 2020
ISSN: 0193-4120 Page No. 3119 - 3128

adding more than 8 processing units is not recommended for data sizes of maximum 64 MB. Looking roughly 64 MB might not look like a large data size, but cloud storage systems always store large data in chunks of 16 to 32 MB, thus this research is valid for Tera Bytes of data as well. They are able to achieve a high performance using ciphertext refresh procedure at the Keg Generation Server (KGS) side, and they have used DGHV scheme which supports parallel processing. Using this research, system designers can plan to integrate parallel processing into their crypto systems and allow them to obtain a higher level of performance as compared to their single threaded counterparts.

Symmetric FHE systems are prone to known plain text attacks, this theory has been proved by Baocang Wang, Yu Zhan, and Zhili Zhang in [17], where they have used the continued fraction algorithm for retrieving the first part of the key, and the Euclidean algorithm for the greatest common divisor problem for retrieving the second part of the key. They claim to have achieved a 98% success ratio in determining the encrypted text. Thus, it is recommended that symmetric FHE systems should not be used, for practical real-world data. Thus, symmetric systems need translation into non-symmetric FHE systems, and the work proposed by Ayantika Chatterjee and Indranil Sengupta in [18] does that. Their proposal describes methods for loop handling, condition handling and other data structure related issues while converting symmetric FHE systems to non-symmetric ones. The result analysis evaluates the systems after conversion, and their performances are found to be almost similar in terms of computational delay.

Thus, this algorithm can be used for algorithmic conversion without compromising on the QoS of the system.Other algorithms like DHCV and CAFED are proposed in [19], which perform FHE operations with ease, but there is a need of improved and secure key exchange mechanism in both. This key-exchange can be handled using the system proposed by Feng Zhao, Chao Li and Chun Feng Liu in [20]. In their system, they have used a dedicated key-management and authentication unit which allows for improved security.

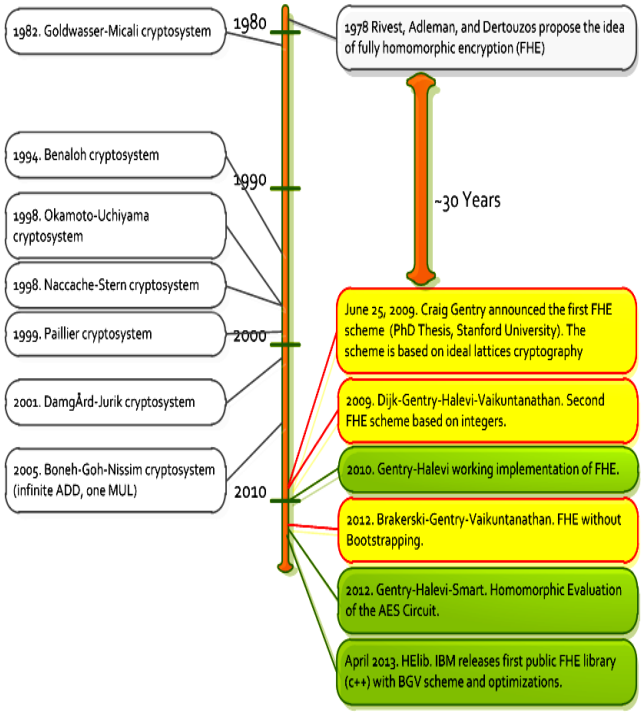

Fig.1 Literature Review of Homomorphic Encryption (Alexander Maximov).

Table I: Homomorphic properties of well-known PHE schemes (Abbas Acar, Hidayat Aksu and A.S.Uluagac)

	Homomorphic Operation	
Scheme	**ADD**	**MUL**
RSA [Rivest et al. 1978b]	NO	YES
GM [Goldwasser and Micali 1982]	YES	NO
El-Gamal [ElGamal 1985]	NO	YES
Benaloh [Benaloh 1994	YES	NO
NS [Naccache and Stern 1998]	YES	NO
OU [Okamoto and Uchiyama 1998]	YES	NO
Paillier [Paillier 1999]	YES	NO
DJ [Damgård and Jurik 2001]	YES	NO
KTX [Kawachi et al. 2007]	YES	NO
Galbraith [Galbraith 2002]	YES	NO

Table II: Some publicly available FHE implementations (Abbas Acar, Hidayat Aksu and A.S.Uluagac)

Name	Scheme	Language	Documentation	Libraries
HElib [Halevi and Shoup 2013b]	BGV [Brakerski et al. 2011]	C++	Yes [Halevi and Shoup 2013a]	NTL, GMP
libScarab [Perl et al. 2011a]	SV [Smart and Vercauteren 2010]	C	Yes [Perl et al. 2011b]	GMP, FLINT, MPFR, MPIR
FHEW [Ducas and Micciancio 2014]	DM14 [Ducas and Micciancio 2015]	C++	Yes [Ducas and Micciancio 2015]	FFTW
TFHE [Chillotti et al. 2017]	CGGI16 [Chillotti et al. 2016]	C++	Yes [Chillotti et al. 2016]	FFTW
SEAL [Laine et al. 2017]	FV12 [Fan and Vercauteren 2012b]	C++	Yes [Chen et al. 2017]	No external dependency

The systems in [19] and [20] can be combined in order to obtain a sufficiently secure system based on the FHE technique. Research work given in [21], [22] and [23] also propose simplistic approaches to security using FHE, and they claim that lattice-based systems are better when compared to non-lattice-based FHE systems.

Conclusion

From the above study we can observe that partial homomorphic systems (PHE) have good level of security, and are high in speed, but they provide either additive or multiplicative homomorphism. While fully homomorphic systems (FHE) provide both security and can perform both additive and multiplicative homomorphism. Some complex systems which use parallel processing for FHE can provide with division and square root operations too. It is observed that CP-ABE FHE systems are superior than KP-ABE FHE systems in terms of security and speed of operation, and thus should be used for all kinds of real-time encryption solutions.

Some systems work with real numbered data, and are complex, thus they must be used for small to moderate scale systems, but their performance can be improved with the help of parallel processing, although this has not yet been proposed, and is an open research issue. Random numbers play a very important role for improving the performance and security of FHE systems. They must be used during the key-generation and encryption phases. It has been observed that random numbers-based systems provide double data rate than conventional systems and are easier to deploy in real-time scenarios. Moreover, usage of quaternion for key-management is much better than using linear computations, as it improves the accuracy, and reduces the latency of the crypto system. Some researchers also claim that cubic interpolations are superior than linear interpolations in some cases and must be used wherever possible. Also, usage of lattice-based calculations for key generation and encryption is more secure and effective in terms of computational delay than their algebraic counterparts.

Future work

Although most of the work in the field of homomorphic encryption suggests usage of pseudo-random numbers, lattice calculations, cubic interpolations and quaternion representations. There is little work done in integrating machine learning as an assistive and optimization technology for FHE systems. Thus, researchers must pursue on how to integrate machine learning for FHE based systems, and make them more efficient in terms of security, delay and other QoS related parameters. Moreover, due to its adaptive nature machine learning can be used for strengthening the security of the FHE algorithms. Blockchain based systems can also be integrated with FHE systems in order to further optimize the security aspects of the system.

REFERENCES

1. Jyun-Neng Ji and Ming-Der Shieh “Efficient Comparison and Swap on Fully Homomorphic Encrypted Data “, 978-1-7281-0397-6/19/$31.00 ©2019 IEEE.
2. NingBo Li, TanPing Zhou, XiaoYuan Yang, YiLiang Han &YuJuan Sun “Efficient Fully

Homomorphic Encryption with Large Plaintext Space", IETE Technical Review-2018

3. C.N.Umadevi and Dr.N.P.Gopalan "Outsourcing Private Cloud Using Symmetric Fully Homomorphic Encryption using Qnp Matrices with Enhanced Access Control", Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018)
4. Jheng-Hao Ye, Si-Quan Chen, and Ming-Der Shieh "Minimizing ESOP Expressions for Fully Homomorphic Encryption" 978-1-5386-4881-0/18/$31.00 ©2018 IEEE.
5. Mikhail Babenko, Nikolay Chervyakov, Andrei Tchernykh, Nikolay Kucherov, Maxim Deryabin, GlebRadchenko, Philippe OA Navaux and Viktor Svyatkin, "Security Analysis of Homomorphic Encryption Scheme for Cloud Computing: Known-Plaintext Attack", 978-1-5386-4340-2/18/$31.00 ©2018 IEEE.
6. M.Zhao and Yang Geng "Homomorphic Encryption Technology for Cloud Computing"8th International Congress of Information and Communication Technology, ICICT 2019 ,Elsevier procedia.
7. Ahmed EL-YAHYAOUI and Mohamed Dafir ECH-CHRIF EL KETTANI, "A verifiable fully homomorphic encryption scheme to secure big data in cloud computing", 978-1-5386-2123-3/17/$31.00 ©2017 IEEE.
8. Zhilin Gong, Youan Xiao, Yihong Long and Yanli Yang," Research on Database Ciphertext Retrieval Based on Homomorphic Encryption" 978-1-5090--/1/$31.00 ©201IEEE
9. Yong Ding and XiuminLi , "Policy Based on Homomorphic Encryption and Retrieval Scheme in Cloud Computing", 978-1-5386-3221-5/17 $31.00 © 2017 IEEE DOI 10.1109/CSE-EUC.2017.105, 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC).
10. Keke Gai1, MeikangQiu, Yujun Li and Xiao-Yang Liu, "Advanced Fully Homomorphic Encryption Scheme Over Real Numbers", 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing, 978-1-5090-6644-5/17 $31.00 © 2017 IEEE.
11. Xidan Song and Yulin Wang," Homomorphic Cloud Computing Scheme Based on Hybrid Homomorphic Encryption",2017 3rd IEEE International conference on Computer and Communications, 978-1-5090-6352-9/17/$31.00 ©2017 IEEE.
12. Dan Wang and Bing Guo, Shun-Jun Cheng,Yan Shen and Yong-Hong li, "A Faster Fully Homomorphic Encryption Scheme in Big Data", 2017 IEEE 2nd International Conference on Big Data Analysis, 978-1-5090-3619-6/17/$31.00 ©2017 IEEE.
13. Keke Gai and MeikangQiu, "An Optimal Fully Homomorphic Encryption Scheme", 2017 IEEE 3rd International Conference on Big Data Security on Cloud, 978-1-5090-6296-6/17 $31.00 © 2017 IEEE.
14. Dalia Tourky, Mohamed EIKawkagy and Arabi Keshk," Homomorphic Encryption the "Holy Grail" of Cryptography", 2016 2nd IEEE International Conference on Computer and Communications, 978-1-4673-9026-2116/$31.00 ©20 16 IEEE.
15. MohdRizuanBaharon, Qi Shi, and David Llewellyn-Jones, "A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing", 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 978-1-5090-0154-5/15 $31.00 ©

2015 IEEE.

16. KamalakantaSethi, Amartya Majumdar and PadmalochanBera," A Novel Implementation of Parallel Homomorphic Encryption for Secure Data Storage in Cloud", IEEE 2017 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security).
17. Baocang Wang, Yu Zhan, and Zhili Zhang, "Cryptanalysis of a Symmetric Fully Homomorphic Encryption Scheme", , IEEE Transactions on Information Forensics and Security.
18. Ayantika Chatterjee and Indranil Sengupta, "Translating Algorithms to handle Fully Homomorphic Encrypted Data on the Cloud", 10.1109/TCC.2015.2481416, IEEE Transactions on Cloud Computing.
19. Baohuachen and Na Zhao, "FULLY HOMOMORPHIC ENCRYPTION APPLICATION IN CLOUD COMPUTING", 978-1-4799-7208-1/14/$31.00 ©2014 IEEE.
20. Feng Zhao, Chao Li and Chun Feng Liu,"A cloud computing security solution based on fully homomorphic encryption", IEEE16th International Conference on Advanced Communication Technology.
21. Darko Hrestak and StjepanPicek," Homomorphic Encryption in the Cloud", MIPRO 2014, 26-30 May 2014, Opatija, Croatia.
22. ZvikaBrakerski and Vinod Vaikuntananthan, "EFFICIENT FULLY HOMOMORPHIC ENCRYPTION FROM (STANDARD) LWE" SIAM J. COMPUT, Vol. 43, No. 2, pp. 831–871.
23. Jian Li, Danjie Song, Sicong Chen and Xiaofeng Lu," A SIMPLE FULLY HOMOMORPHIC ENCRYPTION SCHEME AVAILABLE IN CLOUD COMPUTING", Proceedings of IEEE CCIS2012, 978-1-4673-1857-0/12/$31.00 ©2012 IEEE.