TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

Securing Big Data Processing With Homomorphic Encryption

Tan Soo Fun, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Sabah, Malaysia .
Azman Samsudin, School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
Suraya Alias, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Sabah, Malaysia.

Article Info
Volume 82
Page Number: 11980 - 11991
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 21 February 2020*

Abstract

The arrival of Big Data era has challenged the conventional end-to-end data protection mechanism due to its associated high volume, velocity and variety characteristics. This paper reviews the security mechanisms of dominated Big Data processing platform – Hadoop and examines its capabilities on providing the end-to-end data protection: data-in-transit, data-at-rest and data-in-transform. While Hadoop is limited to protect data-in-transit with its built-in security mechanism and relies on third-party vendor tools (e.g. HDFS disk level encryption or security-enhanced Hadoop security distribution) for securing data-at-rest, the homomorphic encryption scheme that capable of performing computation on encrypted data serve as a promising tool to provide end-to-end data protection Big Data processing. However, existing circuit-based homomorphic encryption schemes still insufficient enough for supporting Big Data applications due to their high complexity of computation, huge generated ciphertext and public key size. To address this problem, this paper proposed homomorphic encryption from a non-circuit-based approach. Our result shows that the newly proposed non-circuit based homomorphic encryption has greatly reduced the computation time and ciphertext size as compared to existing circuit-based homomorphic encryption schemes, therefore amenable to support the high volume and high-velocity requirement of Big Data processing.

Keywords: *Big Data Security, Hadoop Security, Homomorphic Encryption*

I. INTRODUCTION

The terminology of "Big Data" can be traced back to the discussion of processing a large group of datasets in both academia and industry during the 1980s. Today, Big Data is generally described in the association of the "Three V's": high volume, high velocity and high variety of digital data that exceed the capacity of conventional systems to handle them efficiently [1]. Recently, millions of enterprises are investing in Big Data project to better support their decision making the process, customer relationship management, research and development activities, etc. While enterprises are facing the difficulty of in-house Big Data processing, most of them are outsourcing their data solution to third-party such as cloud service providers for cost saving and performance efficiency. However, these outsourced computations has threatening the privacy of enterprise's data since enterprises are delegating a direct access control over their data to a third-party service providers, who might be able to abuse their access in order to infer or more seriously to sabotage some valuable information such as customer information, enterprise's intellectual property, trade secrets, financial information, etc.

II. SECURITY DESIGN OF BIG DATA PROCESSING PLATFORM: HADOOP

A. Security Model of Hadoop

The Hadoop was initially developed by Doug Cutting and Mike Cafarella [3,4] for processing a

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

massive amount of public web data without any security mechanisms. When the Hadoop became a popular platform for analysing and processing data in private networks, many security incidents and insider threats had happened (e.g. malicious user can read arbitrary data blocks from DataNodes, delete massive amounts of data within seconds with distributed nodes, impersonate other users with a simple command switch, etc.) which forced a team from Yahoo group [3] to integrate the Kerberos authentication protocol into Hadoop in 2009. With the integration of Kerberos security mechanism, only a valid user is granted permission (Ticket Granting Ticket, TGT) to access the Hadoop environment. The granting procedure is constructed based on the group permissions and Access Control Lists (ACLs) that maintaining in Kerberos Key Distribution Center (KDC). After granted TGT, the users can request a Hadoop Service Tickets (ST) for accessing data analysis and processing services (e.g. reading Hadoop Distributed File System (HDFS), submitting job queries to Hive or accessing Cassandra services, etc.). Then, the corresponding Hadoop service providers must authenticate the user by decrypting the ST with the corresponding Service Key (SK) from KDC. If the decryption is successful, users are granted to access their requested services. Otherwise, their access request is rejected. In Hadoop configuration documentation [4], it is notable that the Kerberos security mechanism does not enable in a default installation setting. The developers are needed to configure it explicitly. While Kerberos security mechanism focuses on authenticating the user and controlling access on Hadoop services, alternative network encryption schemes are also available for protecting the data-in-transit, depending on the use of communication protocols as follows.

- For communication via web consoles with the Hypertext Transfer Protocol (HTTP), Secure Socket Layer (SSL) is used to secure job monitoring, transferring data from Map tasks to Reducer tasks, transferring metadata between the NameNode and the secondary NameNode;
- For communication via direct Transmission Control Protocol (TCP/IP) communication, the HDFS Data Transfer Protocol is used to protect the data reading and writing between the clients and DataNodes by using the block tokens that extending from the Kerberos service ticket.
- For communication via Hadoop Remote Procedure Call (RPC), Simple Authentication and Security Layer (SASL) mechanisms are used to authenticate the connection between the clients and Hadoop services. The different Quality level of Protection (QoP) can be further configured optionally. For those users who obtained Kerberos tokens after authenticated with Kerberos protocol, the SASL MD5-DIGEST mechanism can be configured for subsequently connection authentication. Otherwise, the SASL GSSAPI mechanism is applied for RPC communication.

Similar to the Kerberos security mechanism, these network encryption schemes are unavailable on the default installation. To protect data processing in Hadoop, enterprises are needed to well-understand the complexity of these Hadoop security models and configure them separately according to their' needs.

B. End to End Data Protection for Hadoop

The end-to-end data protection is referred to as the capabilities to provide data protection during the data-in-transit, data-at-rest, and data-in-transform stage. This section investigates the capabilities of Hadoop platform on providing end-to-end data protection for Big Data processing.

Data-in-Transit. As discussed above, the protection of data-in-transit in Hadoop can be explicitly configured with Kerberos authentication protocol and network encryption schemes (e.g. SSL, HDFS Data Transfer Protocol, SASL mechanisms). However, the Kerberos authentication protocol is associated with the limited authorization capabilities, which is based on simple Access Control Lists (ACLs) maintaining on KDC. For supporting more sophisticated access control policies such as role-based access control,

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

identity-based policy or attribute-based access policy, the enterprise can alternatively purchase the third-party security packages such as Cloudera Sentry, IBM Infosphere Optima Data Masking, DataGuise, Datastax Enterprise, Zettaset Secure, etc.

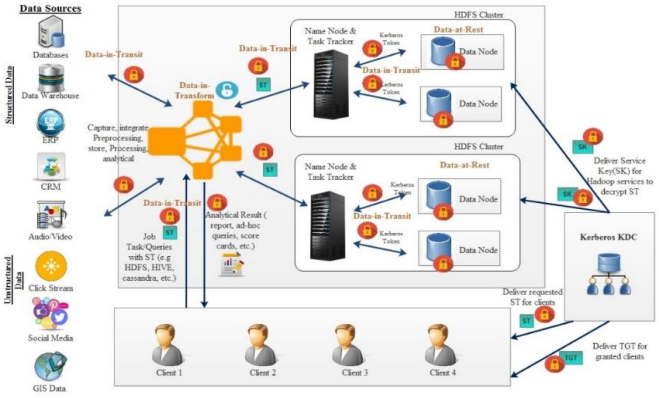

Fig. 1. Security Model of Hadoop Big Data Processing Platform

Data-at-Rest. In Hadoop, data is stored in a disk file (e.g. HDFS) without encryption. For protecting data-at-rest, the enterprise is forced to purchase third-party vendor tools for HDFS disk level encryption (e.g. IBM InfoSphere Optim Data Masking) or security-enhanced Hadoop security distribution (e.g. Intel's secure Hadoop distribution, Zettaset Secure Enterprise Hadoop, etc.).

Data-in-Transform. The last criteria for Hadoop to satisfy the end-to-end data protection are to protect the data during the transformation. Currently, Hadoop does not provide any protection for data-in-transform as shown in Figure 1. Based on our best knowledge, there is no third-party vendor tools are available in the marketplace currently for securing the data during the analytical and computation. The Patient Controlled Encryption (PCE) proposed by Benaloh *et al*. [5] in 2009, that focuses on preserving the privacy of electronic health record might be applicable to protect the data-in-transform in Hadoop. The PCE allows the patient to control the sharing and access to their records by sharing the secret keys with particular service providers. However, their functionality is limited to search the encrypted data in healthcare providers, and cannot support any computation on encrypted data. Meanwhile, the CryptDB created by Popa *et al*. [6] is another attempts to protect the data during transformation. The CryptDB enables the execution of SQL queries over the encrypted data in MySQL databases using a collection of adjustable query-based encryption scheme with overhead approximately 14.5 to 26 per cent of normal database performance [6,7]. However, for supporting different types of queries, every piece of data in CryptDB need to be encrypted under different encryption schemes (e.g. Paillier encryption scheme [8] is used for supporting count query, Song *et al*. encryption scheme [9] is used for supporting keyword search query, etc.). These were resulting in dramatically increased data storage size (approximately 3.76 times [6]), high communication cost and bandwidth required to transfer these data to third-party service providers. To protect data during the transformation, this paper suggests the use of homomorphic encryption that capable is to perform computation tasks on the encrypted data, thus making the end-to-end data protection for Big Data Processing possible.

III. RECENT WORKS ON HOMOMORPHIC ENCRYPTION

Generally, the homomorphic encryption scheme allows computations to be performed on ciphertext without the need for the ciphertext to be decrypted. The notion of homomorphic encryption was originally called privacy homomorphism, introduced by Rivest [10], that aimed to preserve the privacy on bank data. Unfortunately, this scheme was broken by Brickell and Yacobi [11]. Some of the conventional algebraic cryptosystems enjoy some sort of homomorphism, such as ElGamal scheme [12], Goldwasser and Micali scheme [13], Paillier scheme [8], Okamoto-Uchiyama scheme [14], Naccache-Stern scheme [15], Benaloh scheme [16], Domingo-Ferrer scheme[17], etc. However, these schemes are classified as Partial Homomorphic Encryption (PHE) due to their's low versatility [19,42] in which they are only able to perform one type of computational operation on encrypted data. For instance, both the ElGamal scheme and RSA

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

scheme only support for the multiplicative operation; meanwhile, Pallier and Goldwasser and Micali schemes are limited for additive operation. Meanwhile, most of the homomorphic encryption schemes [21–23] proposed in early of the twentieth century are categorized into Somewhat Homomorphic Encryption (SWHE) scheme due to the fact that only limited numbers of operations can be performed on the encrypted data, in order to keep the noise parameters as small as possible.

The first Fully Homomorphic Encryption (FHE) scheme [23] based on the ideal lattice approach was theoretically demonstrated by Gentry in 2009, after three decades of research exploration on homomorphic encryption. A FHE scheme is similar to SWHE scheme excepts that FHE enables arbitrary computations without having to compensate on the increasing of the noise parameters with some noise management techniques either bootstrapping [23], modulus switching [24] or flattening techniques [25]. Most of the latest homomorphic encryption schemes [23–39, 44–50, 52] are categorized into FHE schemes. They are either focuses on improving original Gentry's work [18,26,29,31,35,43], or employing the advanced of mathematical algorithm such as DGHV approach [30, 32, 38, 39], NTRU approach [31, 33] and Learning with Error (LWE) approach [24, 27, 28, 34, 45, 49, 51]. Almost all of these FHE schemes are constructed based on circuit-based approach, where each ciphertext represents a single bit plaintext, or so-called as bit-by-bit encryption [40]. The main advantage of the circuit-based approach is all operations on various operands can be computed by constructing the corresponding circuits [41,42], thus leading to the FHE scheme easily. For instance, if the scheme supports both binary addition and multiplication operations on encrypted data, then it can support arbitrary Boolean functions that involve a series of binary addition and multiplication operations in the encrypted domain.

However, these works are still inadequately to support practical deployments, especially for outsourced Big Data processing due to their high complexity of computation, huge generated ciphertext and public key size [42–44], thus increasing communication costs and bandwidths required for transferring the ciphertexts [43] to third-party service providers. For instance, the implementation result of Gentry's scheme [45] shown that their scheme took more than 900 seconds to add two 32-bit integers and more than 67,000 seconds to multiply them. For encrypting a single bit, the Gentry and Halevi FHE scheme [35] requires a ciphertext of more than 780,000 bits. Furthermore, the circuit-based approach involving the additional task and time spending to decompose the computation function into binary operations.

In order to lessen the computation complexity of these circuit-based approaches, some researchers are looking for hybrid alternatives. These include: (i) ciphertext packing techniques to pack multiple ciphertexts into a single ciphertext [26,46,47]; (ii) batching technique for parallelize any given repeated operations by encrypting multiple bits within the same ciphertext and performs same operations on these bits [28, 48, 49]; and (iii) scheme conversion techniques [18,36,44] for speeding up the re-encryption algorithm by encrypting the plaintext with symmetric encryption scheme (e.g. AES, Prince) which has a simple decryption circuit such as Advanced Encryption Standard (AES) or Prince algorithm. If there is a computation on ciphertexts that need to be carried out, the decryption circuit of the targeted ciphertext is evaluated homomorphically to re-encrypt this plaintext under the FHE scheme. However, these approaches still inherent the bottleneck problem from circuit-based approach, thus difficult to implement for preserving the privacy of Big Data processing.

While most of the recent research work on homomorphic encryption is directed to achieve Fully Homomorphic Encryption (FHE) based on the circuit-based approach that capable to support arbitrary computations, however, such approach is infeasible to support real-world application due to slow speed performance and huge ciphertext and public key size. In this paper, the proposed homomorphic encryption is constructed from the

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

non-circuit-based approach in order to speed up computation time and reduce the ciphertext size for supporting high volume and velocity of Big Data processing

IV. THE NON-CIRCUIT BASED HOMOMORPHIC ENCRYPTION SCHEME

While the recent homomorphic encryption schemes that constructed based on LWE lattice problem such as [24, 51] still suffering from the computational overhead problem, we proposed the more efficient homomorphic encryption scheme based on Ring-LWE problem for preserving the privacy of Big Data processing.

The encryption scheme that we studied is a natural extension of the Lyubashevsky *et al*. basic scheme [52] from a circuit-based approach to a non-circuit based approach. The main differences between the original R-LWE scheme and our variant are twofold. Firstly, we modified the encoding algorithm in order to support the Big Data computation such as means, variance, standard deviation predictive analysis and other statistical analysis such as ANONA, regression analysis, etc. The encoding algorithm of existing R-LWE schemes [29,34, 37,52-54] are representing the message as a collection of binary bits, thereby more efficient to compare two encrypted messages (e.g. private information retrieval application). However, for handling the integer computation, this encoding algorithm requires a deep circuit for simple integer multiplication and for adding two integers, it involves expensive carry operations on homomorphic multiplication over $\mathbb{Z}_2$ as discussed in [53]. Instead, the proposed scheme encoding them as a polynomial rings, $\mathbb{Z}_q[x]/\langle x^d+1\rangle$ (e.g. integer 7 can be randomly presented as x^4-x^3-1 or x^2+x+1) using the technique as discussed in [53,54]. This encoding algorithm enhances the efficient fundamental computation such as addition and multiplication over the integers. The integer addition and multiplication are simply corresponding to the addition and multiplication of polynomials respectively with the assumption q must be large enough for supporting the computation result. Secondly, instead of following the blueprint of bit-by-bit encryption as existing R-LWE schemes [29,34,37,52-54] by encrypting each single bit as a separate ciphertext, the modified encoding algorithm enables us to encode the whole plaintext as a single ciphertext, thus greatly reduce the computation time, ciphertext size and bandwidth required to transfer them to a third-party service providers, therefore capable to support the high volume and high velocity of Big Data processing. Next, the proposed scheme is defined as follows.

Setup (1^λ). Given a security parameter λ, select a sufficiently large prime modulus $q = 1 \bmod (2\lambda)$, and a smaller positive integer p, where $p \ll q$ and $gcd(p,q) = 1$. Let $f(x) = (x^d+1)$, where d is a power of 2. Let $R_q = \mathbb{Z}_q[x] / \langle f(x)\rangle$ be the ring of integer polynomials modulo both $f(x)$ and q. Let $\mathcal{X} = \mathcal{X}(\lambda)$ be an error distribution over R_q. Select a uniformly random secret key, $SK \leftarrow R_q$ and random element, $a_i \leftarrow R_q$. Next, choose a small error term, $e_i \leftarrow \mathcal{X}$. Output the secret key, SK and public parameters, PP as a pair $(a_i, PK_i = a_i.SK + pe_i)$.

Encrypt ($PP, M_1, ..., M_t$). Given the public parameters, PP and t message, M_i, encoding M_i as a polynomial rings $\mathbb{Z}$q[x]/<xd+1> randomly, denoted as $Encode(M_i)$. Next, select a uniformly random, $r_i \leftarrow R_q$ and error terms, $g_i, h_i \leftarrow \mathcal{X}$. Outputs the t ciphertext, $CT_i = (C^0_i, C^1_i)$, where :

$$C^0_i = PK_i \times r_i + Encode(M_i) + pg_i \in R_q$$
$$C^1_i = a_i \times r_i + ph_i \in R_q$$

Evaluate ($PP, F, CT_1, ..., CT_t$). Given the polynomial time computation function F that consists of two fundamental operations: addition and multiplication, and t ciphertext CT_i, performing all operations of F on t ciphertext CT_i. Homomorphic addition is done by simple coefficient-wise addition of t ciphertext CT_i. Homomorphic multiplication is corresponding to the multiplication of two ring polynomials $\mathbb{Z}_q[x]/\langle x^d+1\rangle$. Outputs the computed result in ciphertext space, such that $CT^* = (C^{0*}, C^{1*}) = F((CT_1, ..., CT_t), PP)$, where $CT_1 = (C^0_1, C^1_1), ..., CT_t = (C^0_t, C^1_t)$.

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

Decrypt (*SK, CT**). Given the decryption key, *SK* and computed result in ciphertext space, $CT^* = F((CT_1, ..., CT_t), PP)$, recover the computed result in plaintext space, M^*, such that $F((CT_1, ..., CT_t), PP) = F(M_1, ..., M_t)$. Let $CT^* = (C^{0*}, C^{1*}) = F((CT_1, ..., CT_t), PP)$ and $M^* = F(M_1, ..., M_t)$, compute $M^{*\prime\prime} = C^{0*} - C^{1*} \times SK$ and outputs computed result in polynomial rings $M^{*\prime} = M^{*\prime\prime} \bmod p$. Last, decoding the computed result in plaintext space, where $M^* = Decode(M^{*\prime})$.

Computation on Encrypted Data. Let f be the function consists a tuple of homomorphic addition and multiplication operations iteratively. Let $E(M_1) = CT_1 = (C^0_1, C^1_1) = (PK_1r_1 + Encode(M_1) + pg_1, a_1r_1 + ph_1)$ and $E(M_2) = CT_2 = (C^0_2, C^1_2) = (PK_2r_2 + Encode(M_2) + pg_2, a_2r_2 + ph_2)$ be the two ciphertexts. Homomorphic addition is done by simple coefficient-wise addition of two ciphertext as follows.

$$\begin{aligned} E(M_1) + E(M_2) &= CT_1 + CT_2 \\ &= (C^0_1 + C^0_2, C^1_1 + C^1_2) \\ &= ((PK_1r_1 + PK_2r_2 + Encode(M_1) + Encode(M_2) + p(g_1 + pg_2)), \\ &\quad (a_1r_1 + a_2r_2 + p(h_1 + h_2)) \\ &= E(Encode(M_1 + M_2)) \end{aligned}$$

Meanwhile, the homomorphic multiplication of two ciphertexts is corresponding to the multiplication of two ring polynomials $\mathbb{Z}_q[x]/\langle x^d+1\rangle$, thus avoiding a deep circuit calculation. The homomorphic multiplication is as follows.

$$\begin{aligned} E(M_1) \times E(M_2) &= CT_1 \times CT_2 \\ &= (C^0_1 \times C^0_2, C^1_1 \times C^1_2) \\ &= ((PK_1r_1 (PK_2r_2 + Encode(M_2) + pg_2) + PK_2r_2 (Encode(M_1) + pg_1) + pg_1 (Encode(M_2) + pg_2) + pg_2M_1 + Encode(M_1 \times M_2), (a_1r_2 + ph_1) \times (a_2r_2 + ph_2)) \\ &= E(Encode(M_1 \times M_2)) \end{aligned}$$

The retrieval computed result in plaintext space is similar to the homomorphic addition, except the decryption key is using the joint secret key of CT_2, with the decryption key $:= SK^2$. For certain Big Data computation scenario, it might not accept that the secret key required to decrypt the homomorphic multiplication is dependent on the number of multiplication operations being evaluated. To avoid this, the relinearization technique [24,53] to transform and switch the secret key of ciphertext after every multiplication operation is used. To ensure the computed result can be decrypted correctly, for performing x multiplications, the message m must be encoded as a polynomial of degree at most d/x for correct decryption [53,54]. For handling the Big Data computation such as means, variance, standard deviation predictive analysis and other statistical analysis (e.g. ANONA, regression analysis, etc.)., it is an acceptable trade-off as their computation generally involves a single multiplication or a limited number of multiplications

V. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the proposed scheme with the existing homomorphic encryption schemes: Gentry's Scheme[23], DGHV scheme [30], BGV scheme [27], SCC Scheme [46] and CG scheme [50]. The comparison result is summarized in Table 1. Similar to the existing homomorphic encryption schemes[24, 30, 27, 46, 50], the proposed scheme is constructed based on the hardness of lattice problem. However, Gentry scheme[24], SCC scheme[46] and CG scheme[50] are constructed based on LWE problem that still suffering from the inherent quadratic overhead problem. DGHV Scheme [30] solved this overhead problem by simplifying the conceptual of Gentry Scheme [24] into elementary modular arithmetic with the compensation of reducing their assumption into worst-case hardness - Approximate Greatest Common Divisor (GCD) problem. Meanwhile, the proposed scheme and BGV scheme [27] are constructed based on the more efficient lattice problem - R-LWE problem. each sample $(a, PK) \in R_q \times R_q$ from the Ring-LWE distribution is used to replace n samples $(a, PK) \in Z^n_q \times Z_q$ from the standard LWE distribution, thus reducing both public key and

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

private key size by a factor of n. This subsequently results in a smaller ciphertext size and faster computation as compared to [24,46,50].

The major difference between the proposed scheme and existing homomorphic encryption schemes [26, 32, 29, 48, 52] is the scheme construction method. While existing schemes [26, 32, 29, 48, 52] are constructed based on circuit-based approach, the proposed scheme is constructed with the non-circuit based approach. In circuit-based approach, the plaintext space is represented as a collection of binary bits {0,1} and the computation function is explicitly converted into a deep Boolean circuit that consists of bitwise XOR and AND gates. The homomorphic evaluation is executed by evaluating deep Boolean circuits over binary bits. In the proposed non-circuit based, the plaintext space is represented as a polynomial over the ring. Both integer addition and multiplication operations in the proposed scheme are corresponding directly to the addition and multiplication of polynomials over the rings respectively thus reducing the computation time of homomorphic evaluation as compared to circuit-based approach.

Table 1: Comparison of the Proposed Scheme with Existing Homomorphic Encryption Scheme

	Security Assumption	Algorithm Design	Computation Functions	Encoding Approach
Gentry Scheme [24]	LWE Problem	Circuit-Based Approach	Boolean circuits that consist a tuple of addition (AND) and multiplication (OR) gates.	Bit by Bit Encryption
DGHV Scheme [30]	Approx. GCD Problem	Circuit-Based Approach	Boolean circuits that consist a tuple of addition (AND) and multiplication (OR) gates.	Bit by Bit Encryption
BGV Scheme [27]	Ring-LWE Problem	Circuit-Based Approach	Boolean circuits that consist a tuple of addition (AND)	Bit by Bit Encryption
SCC Scheme [46]	LWE Problem	Circuit-Based Approach	Boolean circuits that consist a tuple of addition (AND) and multiplication (OR) gates.	Non Bit by Bit Encryption
Proposed Scheme	Ring-LWE Problem	Non-Circuit Based Approach	A tuple of polynomial addition and multiplication over the rings, Rq	Non Bit by Bit Encryption

Besides that, compared to the bit by bit encryption in the existing scheme [26, 32, 29, 52] that encoding every single bit as a separate ciphertext, the proposed scheme encoding the whole plaintext as a single ciphertext, thus dramatically reducing the computation time and generated ciphertext size as illustrated in Figure 2(a) and 2(b) respectively. This experiment is implemented with C++ language by using the Microsoft Research Simple Encrypted Arithmetic Library (SEAL) v1.0 and v2.0. The experiment ran on an Intel(R) Core(TM)i7-3612M CPU @ 2.10 GHz 8GB RAM. For 1024 bits of plaintext, the computation time can speed up as much as 1024 times faster than circuit-based approach; and the ciphertext size can be reduced as much as 1024 times smaller than circuit-based approach.

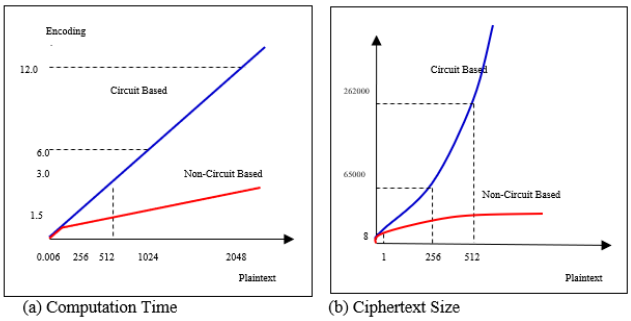

Figure 2. Comparison of Computation Time and Ciphertext Size of the Circuit based homomorphic encryption scheme and the proposed Non-Circuit Based scheme.

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

I. CONCLUSION

This paper discusses the security mechanism of Big Data processing platform-Hadoop and examines its capabilities on providing end-to-end data protection: data-in-transit, data-at-rest, data-in-transform. Our results showed that the Hadoop platform did equip with authentication and authorization protocol (e.g. Kerberos authentication protocol, Secure Socket Layer (SSL), Simple Authentication and Security Layer (SASL) mechanisms, etc.). However, these conventional security mechanisms are limited to protect the data-in-transit; and for protecting data-at-rest, enterprises are forced to purchase third-party vendor tools (e.g. HDFS disk level encryption or security-enhanced Hadoop security distribution). The homomorphic encryption that capable of performing computation on encrypted data serve as a promising tool to ensure end-to-end data protection for Big Data processing. The encryption scheme that we studied is a natural extension of the Lyubashevsky *et al*. basic scheme [51] from a circuit-based approach to a non-circuit based approach in order to support high volume and high velocity of outsourced Big Data computation. Compared to existing circuit-based homomorphic encryption schemes, the proposed scheme has greatly reduced the computation time and ciphertext size. The proposed scheme is secure under the Decision R-LWE$_{d,q,x}$ problem. In future, several interesting aspects of this work can be further explored. For instance, how to extend the proposed work to address another characteristic of Big Data processing - "Variety"? How to provide general access control on these outsourced Big Data computation such as working with identity-based encryption [57] or attributed- based encryption [24,58] is another interesting topic to be explored.

ACKNOWLEDGMENT

This work was by a research grant from Universiti Malaysia Sabah [SLB0159/2017]. The authors also thank the anonymous reviewers of this manuscript for their careful reviews and valuable comment.

II. REFERENCES

[1] Voor, H.G., Klievink, A.J., Arnaboldi,M., Meijerc, A.J. “Rationality and politics of algorithms. Will the promise of big data survive the dynamics of public decision making”, Government Information Quarterly, 2019, Vol. 36(1), pp. 27–38. DOI: 10.1016/j.giq.2018.10.011.

[2] Global Hadoop Market 2019 by Company, Regions, Type and Application, Forecast to 2024, Global Info Research, 2019, pp. 1-124.

[3] O. O. Malley, K. Zhang, S. Radia, R. Marti, and C. Harrell, Hadoop Security Design, 2009; pp. 1–19.

[4] Apache Software Foundation, Apache Hadoop 2.9.2: Hadoop in Secure Mode. [Online]. Available: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SecureMode.html

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient controlled encryption: ensuring privacy of electronic medical records”, Proceedings of the ACM Workshop on Cloud Computing Security, 2009, pp. 103–114.

[6] R. Popa and C. Redfield, “CryptDB: protecting confidentiality with encrypted query processing”, Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP ’11), 2011, pp. 85–100.

[7] Stilgherrian, Encryption’s holy grail is getting closer, one way or another, ZDnet, July, 2015.

[8] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes”, Lecture Notes in Computer Science, vol. 1592, Springer : Berlin, 1999, pp. 223 – 238. DOI: 10.1007/3-540-48910-X_16.

[9] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data” , Proceedings 2000 IEEE Symposium on Security and Privacy, Berkeley, CA, 2000, pp. 44 - 55. DOI: 10.1109/SECPRI.2000.848445.

[10] R. Rivest, “On data banks and privacy homomorphisms”, Foundations of Secure Computation, vol.4, Academia Press, 1978, pp. 169-180.

[11] E. Brickell and Y. Yacobi, “On privacy homomorphisms”, Proceedings of the Advances in Cryptology -EUROCRYPT' 87 (Lecture Notes in Computer Science, vol. 304), Springer: Berlin,

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

1988, pp. 117–125. DOI : 10.1007/3-540-39118-5_12

[12] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms}, IEEE Transactions on Information Theory, Jul. 1985, Vol. 31(4), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[13] S. Goldwasser and S. Micali, "Probabilistic Encryption", Journal of Computer and System Sciences, 1984, Vol.28(2), pp. 270–299.DOI: 10.1016/0022-0000(84)90070-9.

[14] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of the Advances in Cryptology -EUROCRYPT' 98 (Lecture Notes in Computer Science,, vol. 1403), Springer: Berlin, 1998, pp. 308-318. DOI : 10.1007/BFb0054135.

[15] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues", Proceedings of the ISC, 2002, pp. 59–66. DOI : 10.1145/288090.288106.

[16] J. Benaloh, Verifiable secret-ballot elections, Ph. D. Thesis, Yale University, US, 1987.

[17] J. Domingo-ferrer, "A provably secure additive and multiplicative privacy homomorphism", Proceedings of the Advances in Cryptology -EUROCRYPT' 98 (Lecture Notes in Computer Science, vol. 2433), Springer: Berlin, 2002, pp. 471-483. DOI : 10.1007/3-540-45811-5_37.

[18] Y. Doroz, Y. Hu, and B. Sunar, "Homomorphic AES evaluation using NTRU", IACR Cryptology. ePrint Arch., 2014, pp. 1–16.

[19] J. M. Kukucka, An investigation of the theory and applications of homomorphic, Master Thesis, Rensselaer Polytechnic Institute, US, 2013.

[20] C. Castelluccia, "Efficient aggregation of encrypted data in wireless sensor networks", 2nd Annual Conference on Mobile and Ubiquitous Systems: Networking and Services, (MobiQuitous), July 2005; pp. 109 – 117. DOI: 10.1109/MOBIQUITOUS.2005.25

[21] Kipnis and E. Hibshoosh, "Efficient Methods for Practical Fully Homomorphic Symmetric-key Encryption", Randomization and Verification, IACR Cryptology. ePrint Arch., 2012, pp. 1–20.

[22] Sharma, Fully Homomorphic Encryption Scheme with Symmetric Keys, Master Thesis, University College of Enginering, Rajasthan Technical University, India, 2013.

[23] Gentry, A Fully Homomorphic Encryption Scheme, Ph.D Thesis, Standford University, US, 2009.

[24] Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE", Proceedings 52nd IEEE Annual Symposium on Foundations of Computer Science, 2011, pp. 97–106. DOI: 0.1109/FOCS.2011.12.

[25] Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based" , Proceedings of the 33rd Annual Cryptology (Lecture Notes in Computer Science,, vol. 8042), Springer: Berlin, 2013. pp. 75-92. DOI : 10.1007/978-3-642-40041-4_5.

[26] N. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes, Proceedings of the 13th international conference on Practice and Theory in Public Key Cryptography" (Lecture Notes in Computer Science,, vol. 6056), Springer: Berlin, 2010, pp. 420-443. DOI : 10.1007/978-3-642-13013-7_25.

[27] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping", Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 309–325. DOI : 10.1145/2090236.2090262.

[28] Gentry, S. Halevi, and N. Smart, "Better bootstrapping in fully homomorphic encryption", Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography (Lecture Notes in Computer Science,, vol. 7293), Springer: Berlin, 2012, pp. 1-16. DOI : 10.1007/978-3-642-30057-8_1.

[29] C. Gentry and S. Halevi, "Fully Homomorphic encryption without squashing using depth-3 arithmetic circuits", Proceedings of the IEEE 52nd annual Symposium on Foundations of Computer Science, Oct. 2011, pp. 107–109. DOI : 10.1109/FOCS.2011.94.

[30] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers", Proceedings of the 9th

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science,, vol. 6110), Springer: Berlin, 2010, pp. 308-318. DOI : 10.1007/978-3-642-13190-5_2.

[31] Stehlé and R. Steinfeld, "Faster fully homomorphic encryption", Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security (Lecture Notes in Computer Science,, vol. 6477), Springer: Berlin, 2010, pp. 377-394. DOI : 10.1007/978-3-642-17373-8_22.

[32] Chunsheng, "More Practical Fully Homomorphic Encryption", International Journal of Cloud Computing and Service Science. 2012; Vol.1(4), pp. 1–17.

[33] López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption", Proceedings of the 44th Annual ACM symposium on Theory of computing (STOC), 2012, pp. 1219-1234. DOI: 10.1145/2213977.2214086

[34] W. Zhang, S. Liu, and Y. Xiaoyuan, "RLWE-Based Homomorphic Encryption and Private Information Retrieval", Proceedings of the Intelligent Networking and Collaborative Systems, Sep. 2013, pp. 535–540.

[35] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme", Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science, vol. 7417), Springer: Berlin, 2012, pp. 308-318. DOI : 10.1007/978-3-642-20465-4_9.

[36] C. Gentry, S. Halevi, and N. Smart, "Homomorphic evaluation of the AES circuit", Proceedings of the 32nd Annual Cryptology Conference (Lecture Notes in Computer Science,, vol. 7417), Springer: Berlin, 2012, pp. 850-867. DOI : 10.1007/978-3-642-32009-5_49.

[37] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP", Proceedings of the 32nd Annual Cryptology Conference (Lecture Notes in Computer Science,, vol. 7417), Springer: Berlin, 2012, pp. 868-886. DOI : 10.1007/978-3-642-32009-5_50.

[38] J. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys", Proceedings of the 31st Annual Cryptology Conference (Lecture Notes in Computer Science,, vol. 6841), Springer: Berlin, 2011, pp. 487-504. DOI : 10.1007/978-3-642-22792-9_28.

[39] J. Kim, M. Lee, A. Yun, and J. Cheon, "CRT-based Fully homomorphic encryption over the integers," Information Sciences, 2015; Vol.310, pp. 149-162.DOI: 10.1016/j.ins.2015.03.019

[40] Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications", Proceedings of the Information Theory and Applications Workshop (ITA), Feb. 2014, pp. 1–9.

[41] L. Xiao, O. Bastani, and I. Yen, "An Efficient Homomorphic Encryption Protocol for Multi-User Systems", IACR Cryptol. ePrint Arch., 2012.

[42] J. Sen, "Homomorphic Encryption — Theory and Application", Theory and Practice of Cryptography and Network Security Protocols and Technologies, INTECH Publishers, 2013, pp. 1–32.

[43] Y. Hu, "Improving the efficiency of homomorphic encryption schemes", Ph.D. Thesis, Worcester Polytechnic Institute, US, 2013.

[44] Y. Doröz, A. Shahverdi, T. Eisenbarth, and B. Sunar, "Toward practical homomorphic evaluation of block ciphers using prince", Proceedings of the FC 2014 Workshops(BITCOIN and WAHC 2014) (Lecture Notes in Computer Science,, vol. 8438), Springer: Berlin, 2014, pp. 208-220. DOI : 10.1007/978-3-662-44774-1_17.

[45] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping", Proceedings 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 309–325. DOI: 10.1145/2090236.2090262.

[46] X.Song, Z. Chen and L. Chen, "A Multi-Bit Fully Homomorphic Encryption With Shorter Public Key From LWE", IEEE Access, 2019, Vol. 7, pp.

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

50588-50594. DOI: 10.1109/ACCESS.2019.2909286

[47] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Practical packing method in somewhat homomorphic encryption", Proceedings of the 8th International Workshop (DPM 2013) and 6th International Workshop, SETOP 2013, (Lecture Notes in Computer Science,, vol. 8247), Springer: Berlin, 2011, pp. 34-50. DOI : 10.1007/978-3-642-54568-9_3.

[48] C. Gentry, S. Halevi, and N. Smart, "Fully homomorphic encryption with polylog overhead", Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science,, vol. 7237), Springer: Berlin, 2011, pp. 465-482. DOI : 10.1007/978-3-642-29011-4_28.

[49] N. Smart and F. Vercauteren, Fully homomorphic SIMD operations, Designs, codes and cryptography,), Jul. 2012, Vol 71(1), pp. 57–81.DOI: 10.1007/s10623-012-9720-4.

[50] R. Challa, V. Gunta, "An Efficient LWE-Based Additively Homomorphic Encryption with Shorter Public Keys", In: Sa P., Sahoo M., Murugappan M., Wu Y., Majhi B. (eds) Progress in Intelligent Computing Techniques: Theory, Practice, and Applications. Advances in Intelligent Systems and Computing, 2018, Vol. 719, pp.171-177. Springer, Singapore. DOI: 10.1007/978-981-10-3376-6_19

[51] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings", Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science,, vol. 6110), Springer: Berlin, 2010, pp. 1-23. DOI : 10.1007/978-3-642-13190-5_1.

[52] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from Ring-LWE and security for key dependent messages", Proceedings of the 31st Annual Cryptology Conference (Lecture Notes in Computer Science,, vol. 6841), Springer: Berlin, 2011, pp. 505-524. DOI : 10.1007/978-3-642-22792-9_29.

[53] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?", Proceedings of the 3rd ACM workshop on Cloud computing security, 2011, pp. 113–124. DOI: 10.1145/2046660.2046682.

[54] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data", Journal of Biomedical Informatics, 2014; vol. 50, pp. 234–243.

[55] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for Ring-LWE cryptography", Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science,, vol. 7881), Springer: Berlin, 2013, pp. 35-54. DOI : 10.1007/978-3-642-38348-9_3.

[56] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential", Health Information Science and Systems, Dec 2014; Vol. 2(3), pp. 3-13. DOI: 10.1186/2047-2501-2-3.

[57] Wang, Q. Liu and J. Wu, "Achieving fine-grained access control for secure data sharing on cloud servers", Concurrency and Computation: Practice and Experience 2011*; Vol. 23(12), pp. 1443-1464. DOI: 10.1002/cpe.1698.*

[58] S. Tiu, S. Niu and H. Li, "A fine-grained access control and revocation scheme on clouds", Concurrency and Computation: Practice and Experience 2012, Vol. 22(6). DOI: 10.1002/cpe.2956.

AUTHORS PROFILE

Tan Soo Fun is a senior lecturer at the school of Computing and Informatics, Universiti Malaysia Sabah. She completed her PhD in 2017 from Universiti Sains Malaysia(USM). She received her Bachelor of Information Technology (majoring in E-Commerce) and Master of Science (Computer Science) from Universiti Malaysia Sabah (UMS) in 2006 and 2009 respectively. Her research interest includes Cryptography, Information and Network Security. She has published over 30 papers includes book chapters, journals, technical reports and proceedings and received 7 research grants in the related fields.

January - February 2020
ISSN: 0193 - 4120 Page No. 11980 - 11991

Azman Samsudin is a Professor at the School of Computer Science, Universiti Sains Malaysia (USM). He earned his B.Sc. in Computer Science from University of Rochester, New York, USA, in 1989. Later, he received his M.Sc. and Ph.D in Computer Science, in 1993 and 1998, respectively, both from the University of Denver, Colorado, USA. Recently, he serves as Deputy Dean of School of Computer Science, USM. His research interests include Cryptography, Switching Networks and Parallel Computing. He has published more than 100 articles over a series of books, professional journals and conferences.

Suraya Alias is a senior lecturer at the Faculty of Computing and Informatics, Universiti Malaysia Sabah. She completed her PhD in 2018 from Universiti Sains Malaysia (USM). She received her Bachelor of Information Technology and Master of Science (Information Technology) from Universiti Utara Malaysia (UUM) in 2000 and 2007 respectively. Her research interest includes Text and Web Mining, Natural Language Processing and Chatbot technology. She has published over 13 papers includes book chapters, journals, technical reports and proceedings and received 6 research grants in the related fields.