iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

Fully Homomorphic Encryption Using Symmetric Key through Matrix Exponential of Jordan E^{tj} in Cloud Storage

G. Preethi[1] N. P. Gopalan[2]
[1,2] *Research and Development Centre Bharathiar University Coimbatore, India*

Abstract: ***Data owner have security of information keep it in their cloud and client can access anywhere in the world. The stored data should have kept it safe. The outsourced information has many security problems. There are lot of encryption methods ensure the data security even though the cipher text has not yet done. The existing system of Fully Homomorphic Encryption is complex and impractical for large factorization integers. The existing algorithms have high time complexity. This paper, present the circuit based symmetric key homomorphic encryption scheme for large integer factorization. The conventional binary computation is reduced by residue arithmetic operations. This is a improving the fast mechanism of polynomial computation. The stored large factorization data converted into matrix and the secret key is used to generate a matrix into Jordan Normal form of*** e^{tJ}***. It's called matrix exponential. The time complexity is reduced through the exponential computation rather than Fibonacci Qp-matrix. The cipher text can access by the user without the knowledge of original data which is stored in cloud service provider. The encryption scheme is assured to data owner have protect data from an unauthorized people.***
Keywords: Encryption with symmetric key, Matrix Exponential, Cloud Storage, Outsourcing Database

I. INTRODUCTION

The cloud computing is one of the famous filed of storing data in secure manner. The cloud has stored many outsourced data in very large amount of data from multi source and keeps it confidentially. With rapid growing of cloud computing many organization are store their confidential and sensitive data in it. The stored data is protecting from unauthorized people and untrusted cloud service providers. The encryption and decryption is one way of protecting the data. There are many encryption schemes are exist. The most of existing researcher concentrate only in client side encryption. The homomorphic encryption is a solution for these problems. The existing homomorphic encryptions are circuit based. The encryption algorithms are considered only bit by bit of cipher text. The arithmetic operations are achieved by an encryption of circuit based. The key size of circuit based encryption is time consuming process which is applicable for addition and subtraction. Another approach of homomorphic encryption is non-circuit based with fixed number of key size. This approach of non-circuit based encryption is more efficient algorithm called Polygraphic ciphers. The existing encryption of monographic ciphers may split the plaintext into numbers of single letters. The performance of monographic encryption is achieved through bit by bit of data stored in cloud. The proposing idea is to create a secret key in the form of matrix exponential which make convenient and practically easy to handle the large integer factorization.

$$f: m \xrightarrow{k} c \tag{1}$$

The cipher text is

$$C = Ek(M), \qquad M \in m \text{ and } C \in c \tag{2}$$

These are transmits to the receiver via public cloud. But the secrete key is transmits to the receiver via secure way of cloud, the receiver only knows the key k. The receiver can decrypt the C by transformation f is defined as

$$f^{-1}: c \xrightarrow{k} m \tag{3}$$

and finally

$$D_k(C) = D_k(E_k(M)) = M, C \in c \text{ and } M \in m \tag{4}$$

To gets the original text message. The message was splitting into number of groups of letters rather the single letter. There are many different types of secret-key cryptographic systems. This paper proposes an idea of reducing the secret-key size through matrix exponential for large integer's factorization problem in private cloud data.

Organization of this paper

In section 2 Related work in the field of our research of homomorphic encryption. Section 3 the proposed methodology. Section 4 experimental result, section 5 future work and finally conclusion of proposed scheme.

II. RELATED WORK

Rivest, Adleman and Dertouzos was introduced notations for privacy homomorphism in 1978[1]. There are lots of researches made by homomorphism schemes. After the privacy homomorphism, the Partial Homomorphism (PH) were addresses only the operations of addition and multiplication. Gentry's first research was published fully homomorphic [3] encryption from somewhat homomorphic supports few operations in the cipher text. The author gentry suggest a bootstrapping of encryption and decryption in polynomial lower degree of cipher text using refreshing keys. Gentry and Halevi implements the prime number for key generation and they present many optimization techniques. However they are suggesting optimization techniques of ideal lattices, it's hard to implement in practical [4]. Van Dijik, Gentry, Halevi and Vaikuntanathan (DGHV) [5] proposed the integers factorization instead of ideal lattices. The author's were suggests the scheme of conceptual simplicity done by all operations over the integers. The public key sizes were too large in this scheme. The public key size was reduced using a probabilistic decryption algorithm in [6]. Xiao et al [7] suggest a different approach for security of large integer factorization using symmetric keys. The little bit had an improvement in key size and computation time for practical deployment. IBM has released a new software package HELib [8] , which implements the homomorphic encryption are available in existing techniques. Fujitsu in 2013 [9] develops world's first homomorphic encryption rather than bit encryption for multi resource area.

III. OUR CONTRIBUTION

The existing fully homomorphic encryption are based on the public key. The core of this paper found in Liangliang Xiao et al [7]. The master key and more key agents are involved in the communication to the users. The master key has shared to multiple users they can encrypt secret data and send back to the data center. It has a drawback of accessing the same master key by many users. In this paper, proposed matrix exponential for reducing secrete key size and practically acceptable of fully homomorphic symmetric encryption. The proposed method of $\mathbf{e^{tJ}}$ is circuit based encryption in block cipher. The data are divided into number of blocks and the encryption technique is applied to each block cipher text. The result of encryption is written in the form matrix exponential with Jordan form of block cipher text. The circuit based encryption is a symmetric key encryption of the block cipher data. The arithmetic operations are obtained by residues for parallel and fast computation in number of block cipher data. The residue computation of the Chinese Remainder Therorem (CRT) is exactly the idea of divide-and-conquer used in algorithm design.

$$\left(\frac{\mathbb{Z}}{n}\right)^{*} = \frac{\mathbb{Z}}{n_1\mathbb{Z}} X \frac{\mathbb{Z}}{n_2\mathbb{Z}} X \ldots X \frac{\mathbb{Z}}{n_k\mathbb{Z}} \qquad (5)$$

The several small computations combine in each $\mathbb{Z} / n_i\mathbb{Z}$ to a final result in $\mathbb{Z} / n\mathbb{Z}$.

IV. PRELIMINARIES

A. *Block cipher*

The Polygraphic ciphers or block ciphers are more secure rather than the splitting of plaintext into groups of letters. Polygraphic cipher can be described as follows:

1) Splitting Message : Message is denote by M into Blocks of n letters such us (i.e. n=2 is called a digraphic cipher) M_1, M_2, M_3…, M_i, here each block M_j,$1 \leq i \leq j$ is a block contain n letters.

2) Translate the letter into Numerical cipher text

$$C_i \equiv SM_i + P \pmod N,\ i=1,2,\ldots,j \qquad (1)$$

Here S & P is namely secrete key and public key, A is an invertible n x n matrix with gcd(det(S),N)=1, $P= (P_1,P_2,\ldots,P_n)^T$, $Ci = (c_1, c_2,c_3,\ldots,c_n)^T$ and $M=(m_1,m_2,\ldots,m_n)^T$. however, the simplicity form as follows

$$Ci \equiv SMi \bmod 26 \qquad (2)$$

3) To perform decryption

$$M_i \equiv {}^{-1}(C_i - P) \pmod N \qquad (3)$$

Where A^{-1} is the inverse matrix of S. we can write the above equation

$$M_i \equiv S^{-1} C_i \pmod{26} \quad (4)$$

Lemma 1: The given plaintext is written in the form of matrix. The n is key size, the message split into n size. We take the exercise and applied the concept of matrix exponential [] to its. For example, the M = "PLEASE SEND ME THE BOOK, MY CREDIT CARD NO IS SIX ONE TWO THREE". Let n = 3 ,

$$A = \begin{pmatrix} 3 & 13 & 21 & 9 \\ 15 & 10 & 6 & 25 \\ 10 & 17 & 4 & 8 \\ 1 & 23 & 7 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix}$$

The 3 x 3 matrix is use block transformation for encryption. The cipher text can be written with separate block of matrix, $Ci \equiv SM_i + P \pmod{N}$. The encryption and decryption is described as in briefly as follows.

a) Split the message M into blocks of 4-letters and then translate these letters into their corresponding numerical equivalent.

P	L	E	A
15	11	4	0

S	E	S	E
18	4	18	4

N	D	M	E
13	3	12	4

T	H	E	B
19	7	4	1

O	O	K,	M
14	14	10	12

Y	C	R	E
24	2	17	4

D	I	T	C
3	8	19	3

A	R	D	N
1	17	3	13

O	I	S	S
14	8	18	18

I	X	O	N
8	23	14	13

E	T	W	O
4	19	22	14

O	N	E	T
14	13	4	19

H	R	E	E
7	17	4	4

b) Encrypt the above block as follows

$$C1 = A\begin{pmatrix} 15 \\ 11 \\ 4 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 13 \\ 16 \\ 23 \\ 17 \end{pmatrix}, C2 = A\begin{pmatrix} 18 \\ 4 \\ 18 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 15 \\ 19 \\ 22 \\ 17 \end{pmatrix}, C3 = A\begin{pmatrix} 13 \\ 3 \\ 12 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 9 \\ 25 \end{pmatrix}$$

$$C4 = A\begin{pmatrix} 19 \\ 7 \\ 4 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 8 \\ 9 \\ 3 \\ 9 \end{pmatrix}, C5 = A\begin{pmatrix} 14 \\ 14 \\ 10 \\ 12 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 23 \\ 3 \\ 2 \\ 21 \end{pmatrix}, C6 = A\begin{pmatrix} 24 \\ 2 \\ 17 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 24 \\ 5 \\ 18 \\ 6 \end{pmatrix}$$

$$C7 = A\begin{pmatrix} 3 \\ 8 \\ 19 \\ 3 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 20 \\ 23 \\ 14 \\ 21 \end{pmatrix}, C8 = A\begin{pmatrix} 1 \\ 17 \\ 3 \\ 13 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 15 \\ 3 \\ 7 \\ 4 \end{pmatrix}, C9 = A\begin{pmatrix} 14 \\ 8 \\ 18 \\ 18 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 11 \\ 11 \\ 6 \\ 3 \end{pmatrix}$$

$$C10 = A\begin{pmatrix} 8 \\ 23 \\ 14 \\ 13 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \\ 15 \\ 18 \end{pmatrix}, C11 = A\begin{pmatrix} 4 \\ 19 \\ 22 \\ 14 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 16 \\ 25 \\ 25 \\ 6 \end{pmatrix}, C12 = A\begin{pmatrix} 14 \\ 13 \\ 4 \\ 19 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 25 \\ 2 \\ 17 \\ 22 \end{pmatrix}$$

$$C13 = A\begin{pmatrix} 7 \\ 17 \\ 4 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 21 \\ 8 \\ 17 \end{pmatrix} = \begin{pmatrix} 9 \\ 4 \\ 25 \\ 25 \end{pmatrix}$$

c) The numerical nubers are translates into corresponding letters, the ciphertext C_i as follows:

13	16	23	17
N	Q	X	R

15	19	22	17
P	T	W	R

3	2	9	25
D	C	J	Z

8	9	3	9
I	J	D	J

23	3	2	21
X	D	C	V

24	5	18	6
Y	F	S	G

20	23	14	21
U	X	O	V

15	3	7	4
P	D	H	E

11	11	6	3
L	L	G	D

d)
e)

7	0	15	18
H	A	P	S

16	25	25	6
Q	Z	Z	G

25	2	11	22
Z	C	L	W

9	4	25	25
J	E	Z	Z

The given numerical numbers are assigned to each character and the grouped letters are having numbers for matrix exponential calculation.

4) The message M_i can be recover from C_i and to compute A^{-1} $(C_i - B)$ mod 26 as follows:

First compute A^{-1} modulo 26

$$A^{-1} = \begin{pmatrix} 3 & 13 & 21 & 9 \\ 15 & 10 & 6 & 25 \\ 10 & 17 & 4 & 8 \\ 1 & 23 & 7 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 26 & 13 & 20 & 5 \\ 0 & 10 & 11 & 0 \\ 9 & 11 & 15 & 22 \\ 9 & 22 & 6 & 25 \end{pmatrix}$$

And then compute A^{-1} $(C_i - B)$ mod 26

$$M1 = A^{-1}\begin{pmatrix} 13 \\ 16 \\ 23 \\ 17 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 15 \\ 11 \\ 4 \\ 0 \end{pmatrix} \qquad M2 = A^{-1}\begin{pmatrix} 15 \\ 19 \\ 22 \\ 17 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 18 \\ 4 \\ 18 \\ 4 \end{pmatrix}$$

$$M3 = A^{-1}\begin{pmatrix} 3 \\ 2 \\ 9 \\ 25 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 13 \\ 3 \\ 12 \\ 4 \end{pmatrix} \qquad M4 = A^{-1}\begin{pmatrix} 8 \\ 9 \\ 3 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 19 \\ 7 \\ 4 \\ 1 \end{pmatrix}$$

$$M5 = A^{-1}\begin{pmatrix} 23 \\ 3 \\ 2 \\ 21 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 14 \\ 14 \\ 10 \\ 12 \end{pmatrix} \qquad M6 = A^{-1}\begin{pmatrix} 24 \\ 5 \\ 18 \\ 6 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 24 \\ 2 \\ 17 \\ 14 \end{pmatrix}$$

$$M7 = A^{-1}\begin{pmatrix} 20 \\ 23 \\ 14 \\ 21 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 3 \\ 8 \\ 19 \\ 3 \end{pmatrix} \qquad M8 = A^{-1}\begin{pmatrix} 15 \\ 3 \\ 7 \\ 4 \end{pmatrix} - \begin{pmatrix} 1 \\ 21 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 1 \\ 17 \\ 3 \\ 13 \end{pmatrix}$$

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

$$M9 = A^{-1}\begin{pmatrix}11\\11\\6\\3\end{pmatrix} - \begin{pmatrix}1\\21\\8\\7\end{pmatrix} = \begin{pmatrix}14\\8\\18\\18\end{pmatrix} \qquad M10 = A^{-1}\begin{pmatrix}7\\0\\15\\18\end{pmatrix} - \begin{pmatrix}1\\21\\8\\7\end{pmatrix} = \begin{pmatrix}8\\23\\14\\13\end{pmatrix}$$

$$M11 = A^{-1}\begin{pmatrix}16\\25\\25\\6\end{pmatrix} - \begin{pmatrix}1\\21\\8\\7\end{pmatrix} = \begin{pmatrix}4\\19\\22\\14\end{pmatrix} \qquad M12 = A^{-1}\begin{pmatrix}25\\2\\17\\22\end{pmatrix} - \begin{pmatrix}1\\21\\8\\7\end{pmatrix} = \begin{pmatrix}14\\13\\4\\19\end{pmatrix}$$

$$M13 = A^{-1}\begin{pmatrix}9\\4\\25\\25\end{pmatrix} - \begin{pmatrix}1\\21\\8\\7\end{pmatrix} = \begin{pmatrix}7\\17\\4\\4\end{pmatrix}$$

5) Decryption

15	11	4	0
P	L	E	A

18	4	18	4
S	E	S	E

13	3	12	4
N	D	M	E

19	7	4	1
T	H	E	B

14	14	10	12
O	O	K,	M

24	2	17	4
Y	C	R	E

3	8	19	3
D	I	T	C

1	17	3	13
A	R	D	N

14	8	18	18
O	I	S	S

8	23	14	13
I	X	O	N

4	19	22	14
E	T	W	O

14	13	4	19
O	N	E	T

7	17	4	4
H	R	E	E

B. Matrix Exponential

The matrix exponential is a proposed new idea for fast computation in the cipher text. Let N is a prime number and M is denoted by block of plaintexts which are equivalent to it. The group of block letters of plaintext will be replacing by four digit letters of matrix in table ( ). The number of blocks M_i i.e $0 \leq M_i \leq N$. let e^{tJ} is an exponential with Jordan matrix of integer, $0 \leq tJ \leq N$ and gcd(tJ, N-1). The matrix exponential is defined as follows

$ME_i = E_e^{tJ}(M_i) \equiv M_i^{e^{tJ}} \pmod N$ Decryption is applied and gets the original message with help of matrix exponential. The decryption is faster compare to existing method of encryption and decryption. The encryption and decryption is computed by the different arithmetic units of addition (+), subtraction (-) and multiplication (.) with mod N. The conventional computation is having carry propagation but the residue computation is carry-free. This will give good accuracy and fastest computation. A large computation into number of several small computations is "divide-and-conquer" algorithm design. The idea behind in Chinese Remainder Theorem is first nontrivial divide-and-conquer algorithm [applied num theory]. Number of computation executed in parallel and fastest manner. The parallel computation is defined by the diagram in number of small chunks. The divide-and-conquer working process is shown as follows in diagram. The computation implemented in cloud computing process and working process have enhanced and fastest schemes of the data security. Finally the cipher text has been decrypt and gets the actual data with matrix exponential form. This primary step can be achieved by exponential normal form in cloud computing. The computation has provides solution with mod 1151 in few seconds.

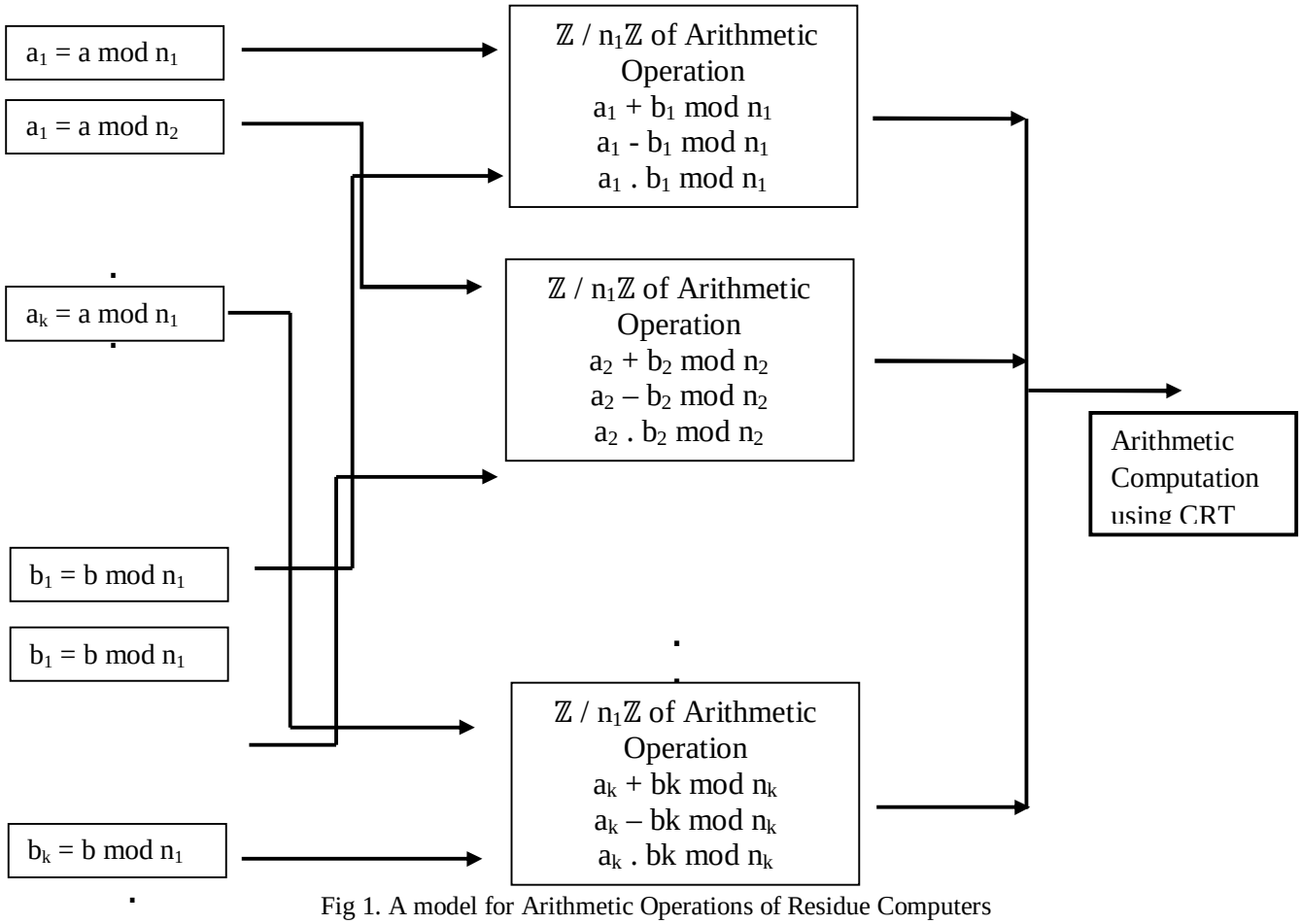


Fig 1. A model for Arithmetic Operations of Residue Computers

V. COMPARISON OF EXISTING ALGORITHM

The time complexity of FHS with Jordan Normal Form of e^{tJ} was analyzed. The matrix exponential computation of two matrix addition, subtraction and multiplication is described below the Table [1]. The performance of existing algorithm is compared to our proposed algorithm. The ƛ, n are security parameters of FHS with e^{tJ}, the modulo function is q, and O, Õ denote asymptotic notation. The table shows plain text, cipher text and effectiveness based on the reduced noise of FHS with matrix exponential. The key length is denoted by gigantic sizes. The time complexity of this FHS scheme is accept compare to the existing noise base schemes.

TABLE 1. Comparison of proposed scheme Vs Existing FHS schemes

Algorithm	Plain Text	Cipher Text	Key Length	Computational Overhead	Effectiveness
Our scheme	$\tilde{O}(ƛ^2)$	$\tilde{O}$ (m $ƛ^{2)}$	O(ƛ)	$\tilde{O}$ (1. ƛ)	O(0.7)
BGV	2dn.log q	2dn. log (q)	Same size of Plaintext	O(ƛ2)	2d.log(q)
DGHV	l . $\tilde{O}(ƛ^5)$	$O(ƛ^{10})$	O(log ƛ)	$\Omega(ƛ^{3.8})$	O(m ƛ)
Wang	Q	Q^2	$O(Q^n)$	O(q)	O(64 q)
Gentry	N	$n^{1.5}$	n^7	n^{2n}	Ω (n^3)

VI. RESULT ANALYSIS

There are several representative queries involved in the experiments. The experiments have different XML datasets as show in Table 1. These are downloaded from benchmark XML repository area. The extended version of the paper [13] having a dataset for verification and execution time for addition, subtraction and multiplication. The verification files are chosen by 15% to 45% of leaf elements without any recurrence. The dataset elements are dividing into number of matrix. The matrix exponential form is applied to data set for residue computation. The proposing idea of this paper is symmetric key with matrix exponential form of manipulations have been achieved in cloud computing through parallel accessing data from external sources.

TABLE 2. Execution time of various dataset and computation lengths of N files.

Dataset	File Size (MB) N	Execution Time (Sec)					
		Public Key Generate	Encryption	Decryption	Addition of chunks	Subtraction of chunks	Multiplication of chunks
Swiss_Prot	120	0.069	0.031	0.003	0.031	0.063	0.351
Auction Data	45	0.023	0.0061	0.001	0.030	0.061	0.985
SIGMOD	467	0.163	0.045	0.231	0.045	0.639	0.022
DBLP	152	0.152	0.020	0.0065	0.120	0.0165	0.152
University _ Course	277	0.275	0.351	0.0731	0.231	0.039	0.275
TPC- H	358	0.345	0.985	0.432	0.0065	0.053	0.346
Mondial	167	0.164	0.022	0.0086	0.0731	0.169	0.461

There are various FHE schemes are implemented and tested by many researchers. Our way of encryption and decryption in the confidential text has different responsibilities. The effective scheme of cryptologists has to allow the computation of multiplication and division requires 33 bit of prime number is confidential. There are various long bits can be implemented for matrix exponential of computation. The exponential computation is takes very few milliseconds for all manipulation. Our implementation algorithm is executed in 4.00 GB, Intel (R) Core i3- processor. The execution time of public key generation, encryption, decryption and manipulation functions in various length of N (file size) is shown in fig. The proposed scheme includes very large key factorization, that will be in the form of matrices contains four letters per group. The size of matrices has huge length but the parallel process of computation in residue computation achieved very fast and effectively.

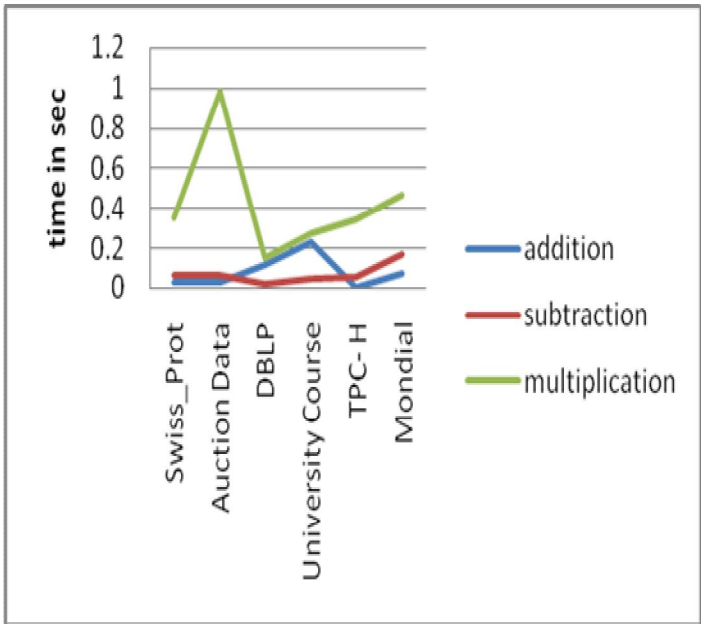


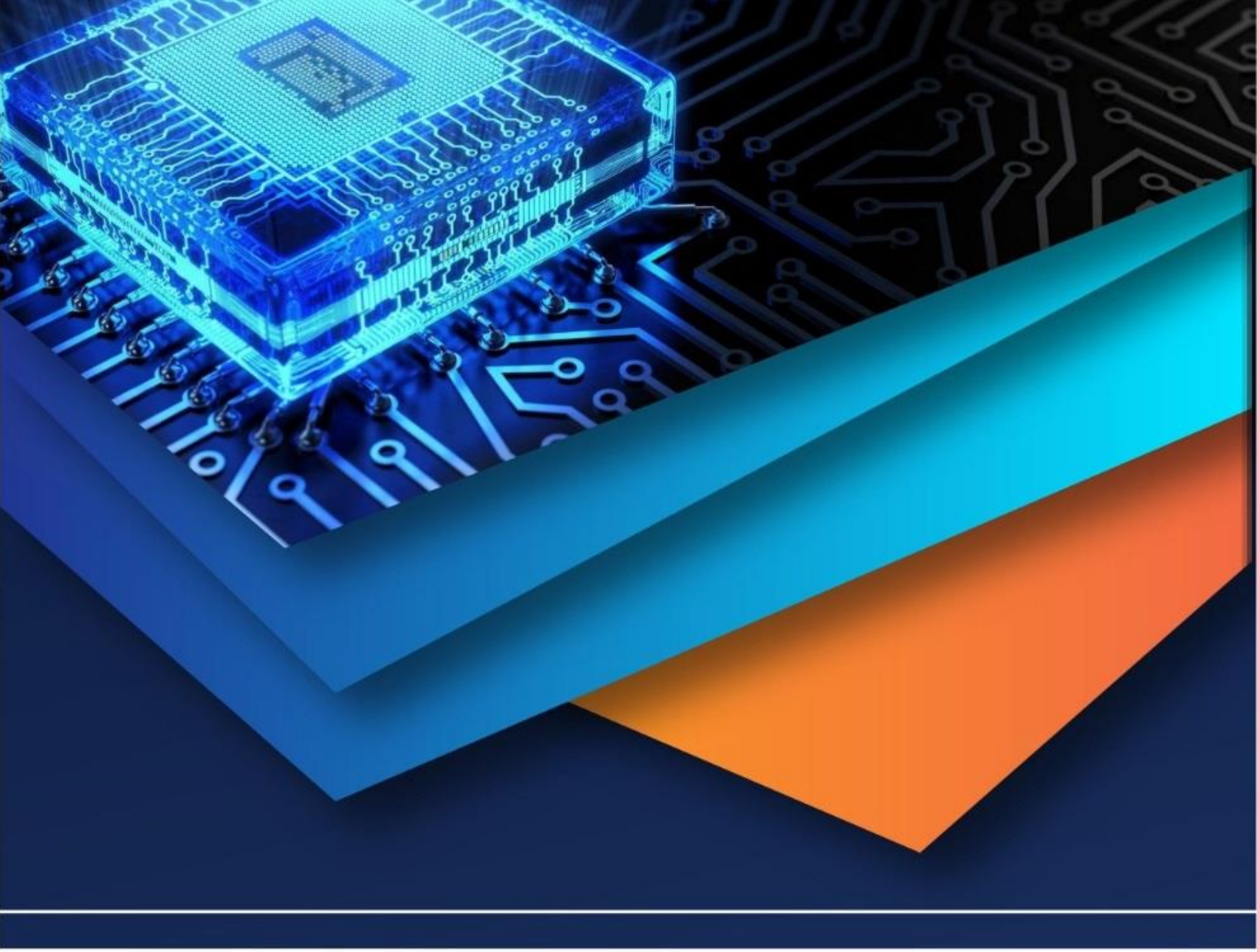
Fig 2. Time Overhead for Dataset in Length N

VII. CONCLUSION AND FUTURE WORK

We proposed a symmetric FHE with matrix exponential e^{tJ} form is based on the polynomial large factorization integers. Our scheme shows the experimental result is good and efficient to handle in practical applications. The proposed approach is working perfectly on the Tree Pattern Queries (TPQs) instead of measuring the distance between the node patterns. This was deployed in XPath queries to protect the confidential outsourcing of XML documents. The matrix exponential form is a new way of encryption in the outsourcing database. The time complexity of this scheme is reduced and compared to existing techniques, it will have low cost. The future work, the matrix exponential encryption is extending to privacy-preserving of auditing protocol into a multi-user storage setting. The TPQs and TPA combined and produce block cipher text of cryptography for better secure and efficiency.

REFERENCES

[1] R. Rivest, L. Adleman and M. Dertouzo. "On data banks and privacy homomorphism": In Foundations of Secure Computation, 1978, pp. 169-180.
[2] C. Gentry, "Fully homomorphic encryption using ideal lattices". In Proc. Of STOC, 2009, pp. 169 - 178.
[3] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits", Cryptology ePrint Archive, Report 2011/279.
[4] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE", in FOCS, 2011. Also available at Cryptology ePrint Archive, Report 2011/344.
[5] Ahmed El-yahyaoui and Mohamed Dafir Elkettani, "Fully homomorphic encryption: state of art and comparison", IJCSIS, Vol. 14, No.4, April 2016, pp. 159-167.
[6] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations", IACR Cryptology ePrint Archive, Report 2011/133.
[7] Z. Chen, J. Wang, ZN. Zhang, and X. Song, "A Fully Homomorphic Encryption Scheme with Better Key Size". Available at https://eprint.iacr.org/2014/697.pdf
[8] Mihai TOGAN, Luciana MOROGAN, Cezar PLESCA, "Comparison- Based Applications for fully homomorphic encryption data", proceedings of the Romanian academy, series A, Vol 16, 2015, pp. 329-338
[9] Haibo Hu, Jianliang Xu, Chushi Ren and Byron Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism", ICDE Conference 2011, pp. 601-612.
[10] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Order-preserving encryption for numeric data", in Proc. SIGMOD, 2004, pp. 563 – 57
[11] Smaranika Dasgupta and S. K. Pal, "Design of a polynomial ring based symmetric homomorphic encryption scheme", PPerspectives in Science, 2016, pp. 692 – 695
[12] Maha TEBAA and Said EL HAJII, "Secure cloud computing through Homomorphic Encryption", IJACT, vol 5, 2013, pp. 29 – 38.
[13] G. Preethi, and N.P. Gopalan, "Integrity Verification for Outsourced XML database in Cloud Storage", ACM proceedings of the ICIA, Article no. 42, 2016.
[14] R. Endsuleit, W. Geiselmann, R. Steinwandt, "Attacking a Polynomial- Based Cryptosystem: Polly Cracker", International Journal of Information Security, Vol. 1, No. 3, 2002, pp. 143- 148.
[15] M. van Dijk, C. Gentry, S. Halevi and V.Vaikuntanathan, "Fully homomorphic encryption over the integers", Accepted to Eurocrypt 2010. Available at http://eprint.iacr.org/2009/616
[16] Preethi Singh, Praveen Shende, "Symmetric key cryptography: Current Trends", IJCSMC, Vol 3, 2014, pp. 410-415
[17] Batch Fully Homomorphic Encryption over the integers", Advances in Cryptology – EUROCRYPT, 2013, PP. 315 – 335.
[18] Manish M Potey, C A Dhote and Deepak H Sharma, "Homomorphic Encryption for Security of Cloud Data", Vol. 79, ICCCV 2016, pp. 175 – 181
[19] Song Y, Yan, Number Theory for Computing, 2002, pp. 303 – 413. eBook Pub: Springer, Berlin, Heidelberg
[20] Jaideep Vaidya, Ibrahim Yakut and Anirban Basu, "Efficient integrity verification for outsourced collaborative filtering", 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY