IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 5, May 2018

Security Enhancement of MIC Outsourcing to a Cloud using CAT MAP

Ankit Pathak[1], Abhishek Mathur[2], Shailendra Kumar Shrivastava[3]

Research Scholar, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India [1]
Assistant Professor, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India[2]
Head of the Department, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India[3]

Abstract: Cloud computing possesses a wide range of great powerful computing resources at very cheap cost. Outsourcing of complicated and huge computational problems to a cloud relaxes the customers with limited resource as well as imposes a negligible charge. This outsourcing trend must be concerned with security and verifiability of the outsourced data. One of the very popular scientific and engineering problem is Matrix Inversion Computation problem (MIC) motivated us to discover a means which provide security, and robust cheating resistance, and an efficiency on MIC Input/output to a malicious cloud. The security involves key generation and encryption of an input to the cloud and decryption and verification of the output from the cloud. The proposed method enables us to combine the key generation and random permutation generation for the input to the cloud. Therefore there is no need to keep eye on key for security. Also, this algorithm decreases the random permutation time by a factor of 2 and cuts a linear key generation time which saves precious computation time and resource occupation.

Keywords: cloud outsourcing, MIC, data security, cat-map.

I. INTRODUCTION

Cloud computing offers a wide range of services, infrastructures, and platforms at a cheapest cost. Customers constrained with limited resources, less power, slow processing are not able to solve their huge complicated computational problems like MIC within a given cost and time. Therefore these limitations attract the customers to outsource their problems to the cloud which provide massive computational power, unlimited resources at a very low cost. These cloud outsourcing is bounded by time, cost, security of data, and verifiability of data [3]. With the tremendous benefits, the cloud outsourcing must be concerned with the list of critical security challenges. The security challenges are:

- Confidentiality of data
- Resistance to robust cheating
- Correctness
- Efficiency

Private data on a public cloud must be protected to maintain the confidentiality of the data. Publically available encrypted data is of no use to the unauthorized or eavesdropper. Therefore there is a requirement of encryption of data before input it to the public cloud and decrypt it when it output from there. The second challenge is the verification of the correctness of the output. Cloud deliberately may cheat and output the incorrect results or the incorrect output may be the consequences of software bug or hardware failure. In both the cases the result verification is needed. The third challenge is correctness of result, i.e. both the public cloud and the client must obey the same protocol. The MIC should be computed according to the peer protocol and the correct result of the original MIC is obtained by the client. The fourth challenge is to recognize the malicious cloud and the result obtained from the malicious cloud. Therefore the output from the malicious cloud must pass through the verification phase to check whether the obtained result is correct or not. Verification should be such that no false (incorrect) output from a cheating cloud can pass the verification phase with the expected threshold value. Additionally, the fifth challenge is to provide better efficiency on cloud to solve problem in comparison with solving by own in terms of time, and cost. This also includes the cost incurred to pre-processing the input to the cloud and post-processing the output from the cloud.

II. MOTIVATION

One of the basic and most popular engineering and scientific problem is MIC (matrix inversion multiplication) which have several applications. For example, the technique behind the computer graphics is MIC in particularly, in 2-Dimensional & 3-Dimensional graphics rendering and 3-Dimensional simulations [5]. Example of MIC in graphics is casting of screen to world ray and physical simulation. In scientific computation MIC plays an important role as an example taking a regression model of order one [4]:

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 5, May 2018

$$y = X\beta$$

Solution to the β using the least squared error method can be obtained by computing the following equation:

$$\beta = (X^T X)^{-1} X^T y$$

In addition to these applications, there are several other important fields where MIC can be applied such as image watermarking [8, 9], image encryption [6,7] etc. Summarily, the MIC is the most needed in the universe of scientific computation by clients. When the resource constrained and cost constrained clients have MIC with large matrix (or a batch of large matrices), then the outsourcing to the cloud possessing powerful resources is the cheapest solution. Confidential data on a naked internet needs security to maintain integrity, confidentiality, and correctness. Therefore these security parameters motivated us to design the protocol which is able to take responsibility of security of the data.

Securities are imposed before input to the cloud and remove from the output of the cloud, but output needs verification. Therefore, security to the input includes encryption and security to the output includes decryption as well as verification. Most of the time and computation client spends in key generation and encryption of the input and then decryption of the output. Therefore to minimize this complexity we proposed a new algorithm where there is no need to generate a separate key and maintain its confidentiality.

III.RELATED WORK

Securing data to be outsource to cloud requires some kind of strong encryption and the result from the cloud needs decryption as well as verification. *Xinyu Lei* and *Xiaofeng Liao [1]* proposed five phase method to provide security. In which there is two different algorithm for generating keys and a separate algorithm for random permutation for encryption.

A.Procedure Secret-Key-Generation: In the first phase they generated K_α and $K_\beta$sets. These sets merely contain a random number where $1 \leq \alpha, \beta_1 \leq n$. In the sets K_α and $K_\beta$no elements less than one is allowed.

B. Random permutation generation: In the second phase they created π_1 and $\pi_2$which initially contains identical n number of elements. These identical elements are then replaced one by one by a random numbers. Therefore, π_1 and $\pi_2$are merely the random permutation replacements of identical elements. The generation of π_1 and $\pi_2$is based on Fisher-Yates shuffle with optimal asymptotic complexity [10, 11].

C. MIC Encryption: Three matrices X, P_1, and $P_2$are involved in encryption process. Where matrix X is the original matrix needs to be invert. P_1, and $P_2$are auxiliary matrices used to help in encryption. P_1, and $P_2$are generated using the secret keys K_α and $K_\beta$and random permutations π_1 and π_2.

All elements of P_1 and P_2matrices are 0 except the diagonal elements. Therefore they are diagonal matrices. P_1 and P_2matrices are generated using the following formula:

$$P_1 = \alpha_i \delta_{\pi 1(i),j} \qquad P_2 = \beta_i \delta_{\pi 2(i),j}$$

Where the Kronecker delta function $\delta_{x,y}$ is defined by

$$\delta_{x,y} = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

Let X(i,j), x_{ij} or x_{ij} denote the entry in i^{th} row and j^{th}column in matrix X, where i and j are indexed from 1 to n.

After generation of diagonal matrices P_1 and $P_2$the original matrix X is encrypted E(X) using simple matrix multiplication i.e.

$$E(X) = Y = P_1 X P_2^{-1}$$

Upon receiving the inverted matrix X' from cloud, first of all the client has to perform decryption on it. The decryption of X' is done using the following sequence of matrix multiplications:

$$D(X') = P_2^{-1} X' P_1$$

Decryption of resultant matrix X' is followed by the verification phase.

Xinyu Lei and *Xiaofeng Liao [1]* proposed a very strong method to impose security for data outsourcing. They used two different algorithms for secret key generation and random permutation. In addition to this, we have to maintain the integrity and confidentiality of the key also because it may be steeled. Therefore instead of generating two key, we designed the algorithm which does not rely on any key generation hence, no need to maintain them. Furthermore, instead of permuting n*n number of elements of π_1 and π_2, we permute its odd numbered elements or vice versa. Then using this we can cut significant computing complexities by a factor of 2.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 5, May 2018

IV. PROPOSED WORK

To cut down the encryption computational complexity of MIC, we introduced a new key generation method which employees the modified cat-map permutation algorithm. In this algorithm initially, we have taken K_α and K_β sets as a value n*n where $1 \leq \alpha, \beta \leq n$ instead of randomly generated numbers. In the permutation phase the modified cat-map algorithm is used which generates permutations π_1 and π_2. The permutation within π_1 and π_2 is done only at the [(i,j)/2+1] positions where (i,j) is the coefficient of π_1 and π_2. At the time when permutations π_1 and π_2 are generating the K_α and K_β sets are also replaced by newly generated value, hence no need of separate random distribution. The algorithm for permutations π_1 as well as α is given below:

piMat = [1:n];//identical matrix of order n*n
ry = 2;//always even number
p = 1;
q = 2; // responsible for generating odd matrix coefficients
fori = 1:xx down to n
for j = 1:jumpFact:yy down to n
π_1(1,mod(ry + (p*q + 1)*(j - 1),yy) + 1) = piMat(i,j);
α(j) = q*(i - 1) + ry + (p*q + 1)*(j - 1);
end
end

The same algorithm is used to generate β and π_2. Using the above algorithm π_1 and π_2 now has permuted matrix space but at the odd places. Because ry, p, and q are fixed, no need to remember the keys because they are not random permutations.

COMPARISION MODEL

Comparison of proposed method and based method is illustrated using Fig. 1 and Fig. 2. Fig. 1 shows that there is no extra phase for key generation and random permutation and Fig. 2 show that these two phases occurs separately.
When client have MIC to outsource to cloud, then it first encrypts it by using key generation and random permutation.

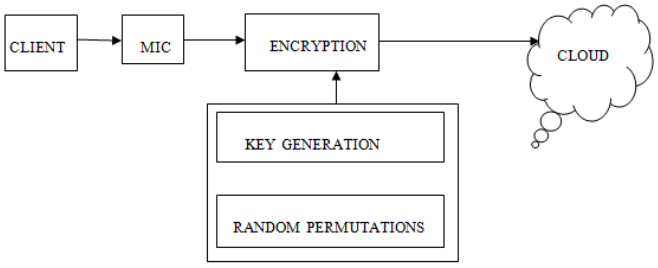

Fig.1. Proposed system architecture

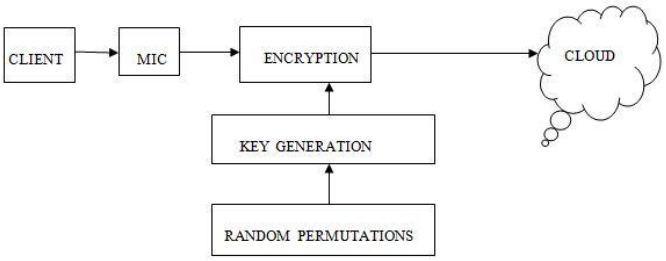

Fig.2. Based system architecture

V. RESULT ANALYSIS

Asymptotically analysis of the proposed method states that the outsourcing of MIC is beneficial to the clients and it beats the method suggested by *Xinyu Lei* and *Xiaofeng Liao [1]* by a factor of 2. This section describes the experimental analysis of the proposed method as well as comparative analysis with the method proposed by *Xinyu Lei* and *Xiaofeng Liao [1]*. The parameters for analysis are the time elapsed by the various algorithms i.e. key generation time, random permutation time, and encryption time.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 5, May 2018

*A. Key generation and random permutation:*In the base method [1], they uses two different algorithms for key generation and random permutation, and their experimental running complexities are O(n) and $O(n^2)$ respectively. Hence the combined running complexity of both the algorithm for base method can never be less than $O(n^2)$. Instead of using two different algorithms for key generation and random permutation, we combined these two into a single. Also while permuting the π_1 and π_2, at the same time we are generating key sets K_α and K_β. Therefore, the combined running complexity of both the algorithm can never be more than $O(n^{1.75})$ because the dominating complexity is because of random permutation which is $O(n^{1.75})$ at most. By cutting half of the permutation computation we are able to cut the execution complexity by a factor of 2, and it approximates the remained complexity 1.75. Summing all this, the proposed method beats the based [1] method

B. Encryption: Encryption of the MIC using previously generated key sets over the same environment for both base [1] and proposed method is analyzed in this section. This phase is totally depends on the dominating function of the random permutation. Therefore, the execution complexity for base method encryption cannot be less than the $O(n^2)$. The running time complexity of encryption process for the presented method cannot be greater than $\theta(n^{1.75})$, this is because the highest order term of the permutation is 1.75.

In this way, the experimental analysis estimates that the overall encryption time including key generation, permutation, and encryption process in the proposed method is less than that of the base [1] method. Table1 shows the asymptotic complexities for proposed and base method. Table II.shows the actual experimented complexities for proposed and base method when matrix size is 256*256. Table III. shows the actual experimented complexities for proposed and base method when matrix size is 512*512. Table IVshows the actual experimented complexities for proposed and base method when matrix size is 1024*1024.

Table I.

	Key generation & permutation	Encryption	Total execution time
Modified cat-map	$O(n^{1.75})$	$\theta(n^{1.75})$	$\theta(n^{1.75})$
Based method	$O(n^2)$	$O(n^2)$	$O(n^2)$

Table II. Execution Complexities for Matrix Order 256*256

	Base	Proposed
Key generation& permutation	0.398	0.0142
Encryption	0.0805	0.599
Total Execution Time	1.3286	0.8236

Table III. Execution Complexities for Matrix Order 512*512

	Base	Proposed
Key generation& permutation	0.0115	0.094
Encryption	0.6612	0.4599
Total Execution Time	8.4721	5.1365

Table IV. Execution Complexities for Matrix Order 1024*1024

	Base	Proposed
Key generation& permutation	0.0117	0.095
Encryption	6.9859	4.3378
Total Execution Time	58.0723	35.1136

The complexity comparison table I shows that the proposed model is asymptotically outperform the base [1] method. And the complexity comparison table II, table III, table IV shows that the proposed model is experimentally outperform the base [1] method. The overall execution complexity for all size of matrices can be formulized as follows:

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 5, May 2018

Proposed < base

VI. CONCLUSION

In this paper, we focused on to cut down the complexity of the key generation for encryption. The modified cat-map algorithm lets combine the two phase into single for generating four random distributions. The proposed algorithm showed that, we are able to save the half of the computational time while generating the key. we saved the random permutation time by a factor of $n^{1.75}$.Also we saved overall Encryption and decryption computation time. With the MIC is well known and popular engineering and scientific challenge, the proposed method can be utilized and bind with some higher level protocols.

REFERENCES

[1]. Xinyu Lei, Xiaofeng Liao, "*Outsourcing Large Matrix Inversion Computation to a Public Cloud*", IEEE transaction on cloud computing, vol. 1, pp 78-87, june 2013.
[2]. Xinyu Lei, Xiaofeng Liao, Tingwen Huang, FenoHeriniaina, "*Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud*", Information Science, Elsevier, vol. 280, pp. 205-216, Oct, 2014.
[3]. G. Brunette and R. Mogull, "*Security Guidance for Critical Areas of Focus in Cloud Computing V2.1*," Cloud Security Alliance, pp. 1-76, 2009.
[4]. G.A. Seber and A.J. Lee, Linear Regression Analysis. John Wiley & Sons 2012.
[5]. S.F. Gibson and B. Mirtich, "*A Survey of Deformable Modeling in Computer Graphics*," Technical Report TR-97-19, Mitsubishi Electric Research Laboratory, 1997.
[6]. R. Tao, X.-Y. Meng, and Y. Wang, "*Image Encryption with Multiorders of Fractional Fourier Transforms*," IEEE Trans. Information Forensics and Security, vol. 5, no. 4, pp. 734-738, Dec. 2010.
[7]. X. Zhang, "*Lossy Compression and Iterative Reconstruction for Encrypted Image*," IEEE Trans. Information Forensics and Security, vol. 6, no. 1, pp. 53-58, Mar. 2011.
[8]. S. Lee, C.D. Yoo, and T. Kalker, "*Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform*," IEEE Trans. Information Forensics and Security, vol. 2, no. 3, pp. 321-330, Sept. 2007.
[9]. X. Zhang, Z. Qian, Y. Ren, and G. Feng, "*Watermarking with Flexible Self-Recovery Quality Based on Compressive Sensing and Compositive Reconstruction*," IEEE Trans. Information Forensics and Security, vol. 6, no. 4, pp. 1223-1232, Dec. 2011.
[10]. R. Durstenfeld, "*Algorithm 235: Random Permutation*," Comm. the ACM, vol. 7, no. 7, p. 420, 1964.
[11]. D.E. Knuth, The Art of Computer Programming. Addison-Wesley, 2006.

BIOGRAPHY

Ankit Pathak received the B.E. degree in Computer Science and Engineering from Rajiv Gandhi ProudyogikiVishwavidyalaya Bhopal, India. He is currently a M. Tech. student in the Department of Computer Science, Samrat Ashok Technological Institute Vidisha, India. His research interest is in security in cloud computing.