A CRYPTOGRAPHIC SECURITY MECHANISM FOR DYNAMIC GROUPS IN PUBLIC CLOUD ENVIRONMENTS

Sheenal Malviya[1], Sourabh Dave[2]
[1]Department of Computer Science, Medicaps University, Indore, India.
[1]sheenalmalviya@gmail.com
[2]Department of Computer Science, Medicaps University, Indore, India
[2]sourabh.dave@gmail.com

Abstract
Cloud computing has emerged as an extremely important domain of technology, primarily due to the emergence of big data, machine learning and quantum computing applications. While earlier, cloud computing services were primarily focussed on providing storage and some infrastructures/platforms for applications, the need to advances computational power analysis of extremely large datasets has made cloud computing almost inevitable from most client based applications, be it mobile applications or web applications. With the necessity to access public clouds, the allied challenge to protect data shared from and to cloud based platforms has also cropped up. While conventional cryptographic algorithms have been used for securing and authenticating cloud data, advancements in cryptanalysis and access to faster computation has let to possible threats to conventional cloud security mechanisms. This has led to extensive research in the field of homomorphic encryption pertaining to cloud security. In this work, a security mechanism is designed which is targeted towards dynamic groups using public clouds. A big challenge in terms of overhead, throughput and execution time is generally faced by cloud security mechanisms which need to encrypt data form dynamic groups with frequent member addition and removal. In this work, a two-stage homomorphic encryption process is proposed for data security. The performance of the proposed system is evaluated in terms of the salient cryptographic metrics which are the avalanche effect, throughput and execution time. A comparative analysis with conventional cryptographic algorithms show that the proposed system outperforms them in regards to the cryptographic performance metrics.

***Keywords*—Cloud Data Security, Dynamic Groups, Homomorphic Encryption, Chaotic Network, Throughput.**

1. INTRODUCTION

Cloud computing has revolutionized the way computing has been performed conventionally. The emergence of cloud computing has allowed applications with constrained computational and memory resources access platforms with extremely high computational and storage prowess [1]. Using cloud based services, users can harness the power of sophisticated hardware or software through their remote machines. A typical cloud platform interacting with a remote client is depicted in figure 1. In figure 1, multiple applications are being run by the cloud platform which are accessible by the client (often termed as host) through the virtual machine (VM). The VM emulates a virtual interface between the host operating system and the cloud platform.

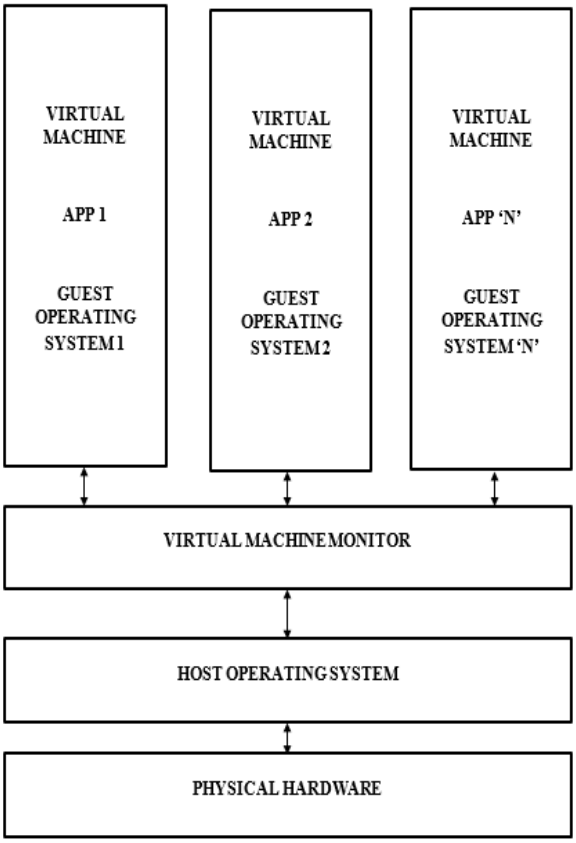


Fig. 1 Typical illustration of a cloud platform interacting with a remote client.

. If user groups are accessing the cloud platform, then there is generally a group admin to oversee the group members [2]-[3]. The group members may have the authority to access, manipulate or upload data. Moreover, the groups are generally dynamic in nature due to the frequent addition and deletion of group members. In general, several cloud service providers may or may not have access to the internal functionality of the cloud architecture which often leads to loopholes in cloud security which employ conventional cryptographic techniques for cloud security. Generally, the authentication of group users is accomplished based on a group key sharing mechanism. The second issue to be addressed is the design of security mechanisms which can upload data to the cloud based server in less time and high security so as to seamless operate between the remote client and the server. In this work, a key generation approach is proposed to authenticate users in the group and a homographic encryption algorithms is employed to encrypt data to the cloud server.

2. System Model For Dynamic Groups

Workgroups are generally dynamic in nature with the number of members changing and/or members migrating to other groups [3]. Hence it is necessary to authenticate the members joining a group which have access to the cloud platform having the following accesses:

1) Fetching data
2) Manipulating data
3) Uploading data

The dynamic nature of such groups leads to the chance of impersonation, eavesdropping and man in the middle attack. With networks migrating from the wired to the wireless realm, authentication has become a serious challenge. Most of the intrusions generally occur at this level due to the ease of breakthrough into the network. The key however should be renewed intermittently so as to decrease the chances of fraudulent intercepts. This is generally implemented based on a public key infrastructure (PKI) based system to authenticate users. The PKI infrastructure is generally preferred due to its relatively low complexity and ease of use. In the proposed architecture, the PKI based authentication mechanism has the admin portal (AP), user portal (UP) and Cryptographic Server (CS) [4]. The admin portal oversees the functioning of user portals and grants access to files. The user portals can upload, download or manipulate data files [5]-[6]. The user portals of each of the entities have unique login credentials and are authenticated using the key. The encryption is however done solely by the cryptographic server.

3. Proposed Approach For Homomorphic Encryption

Homomorphic encryption has gained popularity due to its ability to allow computations directly on the cypher text without the mandatory necessity of decrypting the plain text in the first place. This approach as two major benefits [2]:

1) Enhanced Security: Since data processing can be done directly on the cipher text, data hiding, manipulations and multi-layer security can be accomplished relatively easily compared to conventional encryption.
2) Lesser Execution Time: Since the need to decrypting the actual data is thwarted, the time of execution plummets.

Homomorphic encryption is generally categorized additive homomorphic encryption and multiplicative homomorphic encryption.

3.1 Additive Homomorphic Encryption:

In this case, two large primes p and q are selected which yield an integer n given by:

$$n = p.q = \text{p.q} \quad (1)$$

Next, a key generator G is designed such that:

$$G \in I^{+} \in \text{I}\hat{}+ \quad (2)$$

Here,

I^{+} is the positive set of integers

Typically, n and G are public while the values of p & q are private.

Let a plain text (p) and key (k) yield a cipher text (c) given by

$$c = f(p, k, r, E) = \text{f(p, k,r,E)} \quad (3)$$

Here,

E is the encryption algorithm

f stands for a function of:

r is a chosen random number

Then, the following relations hold true:

$$k(p, r) = G^{p} r^{n} mod(n^2) \quad (4)$$

$$k(p1, r1)k(p2, r2) = g^{p1} r_1^{n} g^{p2} r_2^{n} mod n^2 \quad (5)$$

$$\text{Or } k(p1, r1)k(p2, r2) = g^{p1+p2} r1 r2^{n} mod n^2 \quad (6)$$

$$\text{Or } k(p1, r1)k(p2, r2) = k(p_1 + p_2, r_1 r_{2)} \quad (7)$$

Here,

p1 and p2 are two distinct plain texts,

r1 and r2 are the two distinct random numbers chosen

Equation (7) reveals that the separate plain texts can be obtained as a sum of the individual plain texts with the same key (k)

3.2 Multiplicative Homomorphic Encryption:

In this case, the following relations hold true:

$$k(p1)k(p2) = p_1^{b} p_2^{b} mod(n) \quad (8)$$

$$\text{Or } (p_1 p_2)^{b} mod\ (n) = k(p_1 p_2) \quad (9)$$

Equation (9) reveals that the separate plain texts can be obtained as a product of the individual plain texts with the same key (k).

Here a and b are chosen such that $ab \equiv 1\text{b} \equiv 1$

3.3 posed Technique

The proposed approach is an amalgamation of the Message Digest 5 (MD5) and the Advanced Encryption Standard (AES) algorithms [7]. The key is generated using the relation:

$$S = v_b + v_a = \text{v\_b+ v\_a} \quad (10)$$

$$key_{hash} = MD5(S) + f \quad (11)$$

$$Key_{enc} = key_{hash} + (v_b + v_a) \quad (12)$$

Here,

v_b & v_a are chosen byte file and attribute file respectively

f is the flag value

key_{hash} is the hash key at stage 1

Key_{enc} is the encryption key of stage 2

The first stage of encryption is done based on the AES algorithm given by:

$$CT = AES\ (CT, key_{AES}\text{T=AES (CT, key\_AES)} \quad (13)$$

The homomorphic substitution for the second stage is done using logistic maps based on a chaotic network given by:

$$Z_{n+1} = rZ_n(1 - Z_n) \quad (14)$$

Here,

r is the growth rate

Z_n is the state of the generator at iteration n

Z_{n+1} is the state of the generator at iteration n+1

The diagrammatic representation of the generation of the random states is depicted in figure 2.

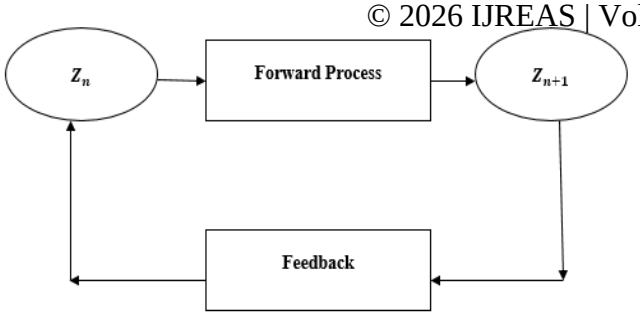


Fig. 2 Generation of the random states based on the recursive two state machine

Only the authentic user has the value of r and initial value Z_0 which uniquely determine the state Z_n.
This type of a generator is capable to generate random values using deterministic machines.

3.4 Evaluation Parameters

The evaluation of the proposed system is based on the following metrics [8]-[9]:

$$Throughput = \frac{Data\ Size\ Processed}{Time\ of\ Execution}$$hroughput=Data Size Processed/Time of Execution (15)

$$Avalanche\ Effect = \frac{No.of\ bits\ changed\ in\ Cipher\ Text}{No.\ of\ bits\ changed\ in\ Plain\ Text}$$valanche Effect=No. of bits changed in Cipher Text/No. of bits changed in Plain Text (16)

$$Mean\ Square\ Error\ (mse) = \frac{1}{n}\sum_{i=1}^{i=n}(P_{ia} - P_{id})^2$$ean Square Error mse=1/n_i=1^i=n?(P_ia-P_id)^2 (17)

Here,
n is the number of bits
P_{ia} is the actual plain text
P_{id} is the deciphered plain text after decryption

4. Simulation Results

The obtained results are presented in this section along with the values of the performance metrics:

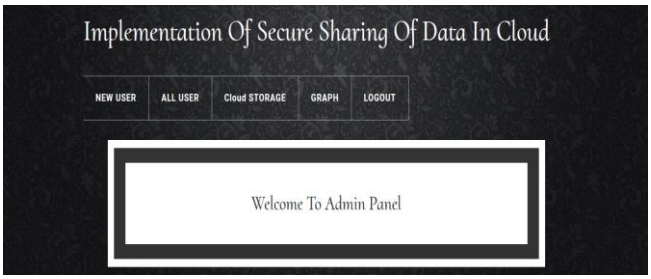
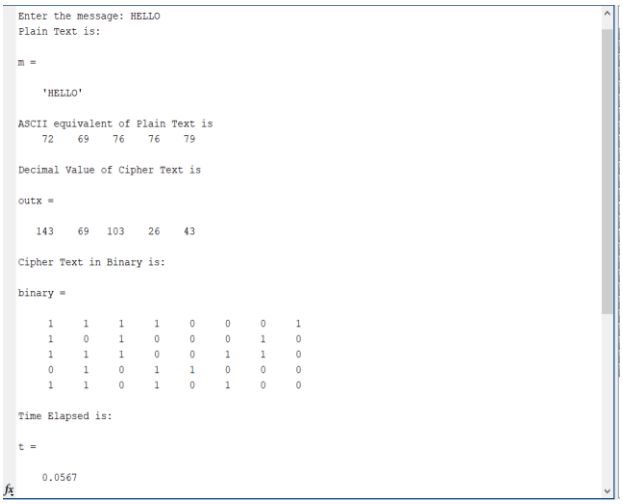


Fig. 3 Creation of Admin Panel

Fig. 4 Creation of User Panel

Figures 3 and 4 depict the creation of the admin and user panels. For simulation, both textual and image data formats have been utilized. The same logic can be extended to test audio and video files.



```
Enter the message: HELLO
Plain Text is:

m =

    'HELLO'

ASCII equivalent of Plain Text is
    72    69    76    76    79

Decimal Value of Cipher Text is

outx =

   143    69   103    26    43

Cipher Text in Binary is:

binary =

     1     1     1     1     0     0     0     1
     1     0     1     0     0     0     1     0
     1     1     1     0     0     1     1     0
     0     1     0     1     1     0     0     0
     1     1     0     1     0     1     0     0

Time Elapsed is:

t =

    0.0567
```

Fig. 5 Encryption of Text Data

Figure 5 depicts the plain text (text data), its ASCII equivalent, decimal equivalent of cipher text and the binary equivalent of the cipher text. The time elapsed is also depicted.

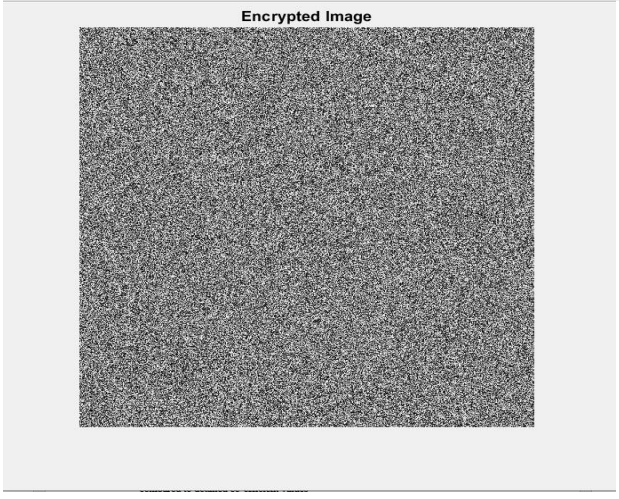


Fig. 6 (a) Plain Text Image (b) Encrypted Image Data

Figure 6 (a) depicts the plain text (image data), which is a standard test image (cameraman.jpg) and 6(b) depicts its encrypted counterpart.

The performance of the proposed system is evaluated in terms of the throughput, execution time and the avalanche effect. The generator set of the chaotic network (often termed as the Mandelbrot set) is depicted in figure 7. Multiple runs of the proposed algorithm for varying data size has been depicted in figure 8. Figures 9 and 10 depict the throughput and avalanche effect obtained by the proposed algorithm. A comparative analysis has been made w.r.t. Data Encryption Standard (DES), AES and Blowfish algorithms. It can be seen from the comparative analysis, that the proposed approach outperforms the existing conventional techniques in terms of the avalanche effect and throughput. The mse obtained is 1.08. The mean square error is significant for image data since the decrypted image shows slight variations or errors compared to the original plain text data. Due to positive and negative polarities of errors, it is customary to compute the mean square error.

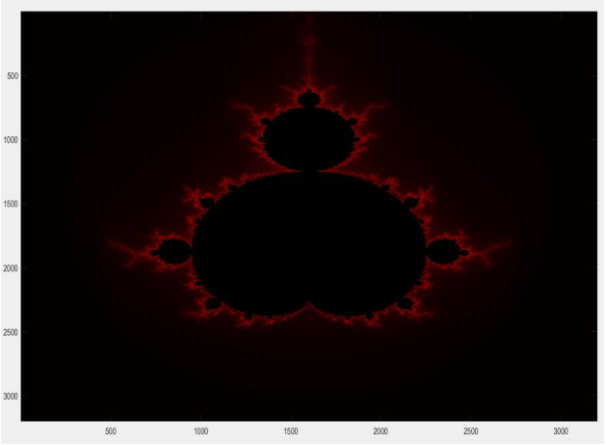


Fig. 7 Visual representation of the Mandelbrot set generated based on the chaotic network.

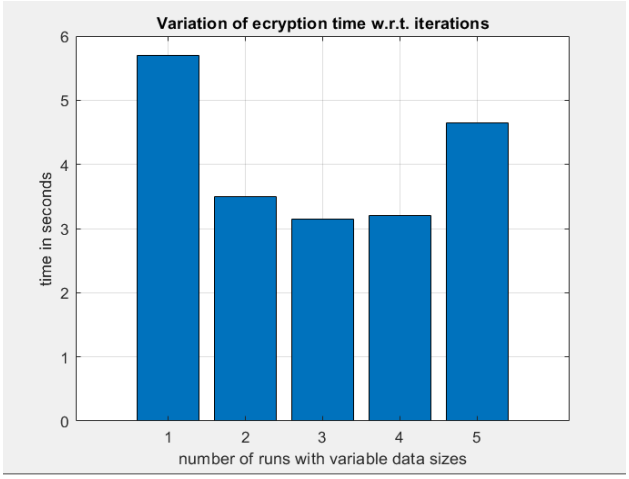


Fig. 8 Variation of Encryption Time w.r.t. runs and data size

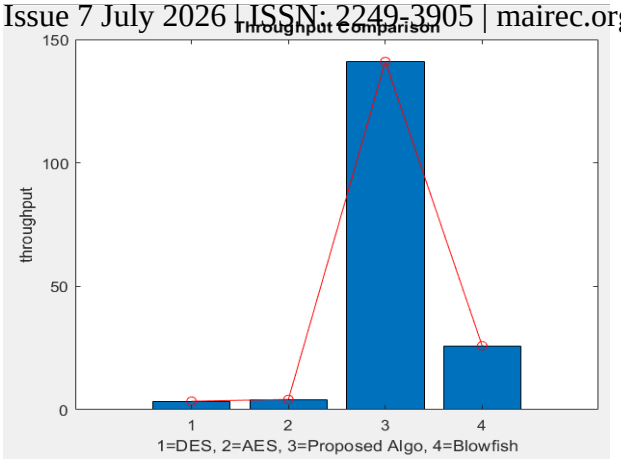


Fig. 9 Comparison of throughput w.r.t. standard encryption algorithms

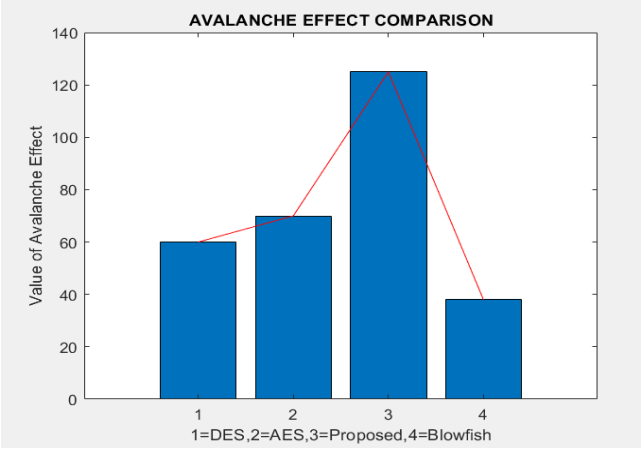


Fig. 10 Comparison of Avalanche Effect w.r.t. standard encryption algorithms

While throughput is a measure of the speed of processing of the algorithm, the avalanche effect is a measure of the randomness of the algorithm. High values of throughput and avalanche effect clearly indicate that the proposed algorithm is fast as well as highly random in nature.

5. Conclusions

The proposed work presents a data security mechanisms for cloud environments which are accessed by dynamic groups. The security mechanism developed is a two way approach in which the key is generated using the MD5 algorithm while the cipher text is generated using the key and the AES algorithm. Homomorphic encryption is then implemented using the chaotic logistic map. The Mandelbrot set obtained is also shown in the figure. The performance of the proposed algorithm is evaluated in terms of the avalanche effect, throughput and execution time. The results indicate that the proposed algorithm outperforms the existing cryptographic algorithms in terms of throughput and avalanche effects which are the inference metrics of speed and randomness of an algorithm. Thus it can be concluded that the proposed approach is an effective technique for securing cloud data especially for public clouds.

Acknowledgment

The authors acknowledge the support and constructive criticism of the faculty members of the department of Computer Science, Medicaps University, Indore.

REFERENCES

[1] M. Sookhak, A. Gani, K. M. Khan and R. Buyya, "Dynamic remote data auditing for securing big data storage in cloud computing," Information Sciences, Vol- 380, pp: 101-116, 2017.

[2] P. A. Pimpalkar and H. A. Hingoliwala, "A Secure Cloud Storage System with Secure Data Forwarding," *International Journal of Scientific & Engineering Research*, Vol- 4, Issue-6, pp: 3002-3010, 2013

[3] H.T.Wu, Y.M.Cheung, Z.Yang, S.Tang, "A high-capacity reversible data hiding method for homomorphic encrypted images*", Journal of Visual Communication and Image Representation*, *Elsevier* 2019,Vol-62, pp:87-96.

[4] A Bakhshandeh, Z Eslami "An authenticated image encryption scheme based on chaotic maps and memory cellular automata", *Journal of Optics and Lasers in Engineering, Elsevier* 2013, Vol-51, Issue-6, pp:665-673.

[5] M.Rani, V.Kumar, "Superior mandelbrot set", *Journal of Korea Socety of Mathematical Education,*2004, Vol-8, Issue-4, pp:279-291.

[6] M Tebaa, S El Hajji, A El Ghazi, "Homomorphic encryption applied to the cloud computing security", *Proceedings of the World Congress on Engineering,* 2012 Vol-1, pp:1-4

[7] MP Babitha, KRR Babu, "Secure cloud storage using AES encryption", *Proceedings in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp: 859-864,

[8] N Shimbre, P Deshpande, "Enhancing distributed data storage security for cloud computing using TPA and AES algorithm", Proceedings in *2015 International Conference on Computing Communication Control and Automation*, pp: 35-39.

[9] L Coppolino, S D'Antonio, G Mazzeo, "Cloud security: Emerging threats and current solutions", *Journal of Computers and Electrical Engineering", Elsevier* 2017, Volume 59, pp: 126-140