# A Cryptographic Security Mechanism for Dynamic Groups for Public Cloud Environments

Submitted: 8th February 2022; accepted 17th March 2022

*Sheenal Malviya, Sourabh Dave, Kailash Chandra Bandhu, Ratnesh Litoriya**

DOI: 10.14313/JAMRIS/2-2022/15

**Abstract:**
*Cloud computing has emerged as a significant technology domain, primarily due to the emergence of big data, machine learning, and quantum computing applications. While earlier, cloud computing services were focused mainly on providing storage and some infrastructures/platforms for applications, the need to advance computational power analysis of massive datasets. It has made cloud computing almost inevitable from most client-based applications, mobile applications, or web applications. The allied challenge to protect data shared from and to cloud-based platforms has cropped up with the necessity to access public clouds. While conventional cryptographic algorithms have been used for securing and authenticating cloud data, advancements in cryptanalysis and access to faster computation have led to possible threats to the traditional security of cloud mechanisms. This has led to extensive research in homomorphic encryption pertaining to cloud security. In this paper, a security mechanism is designed targeted towards dynamic groups using public clouds. Cloud security mechanisms generally face a significant challenge in terms of overhead, throughput, and execution time to encrypt data from dynamic groups with frequent member addition and removal. A two-stage homomorphic encryption process is proposed for data security in this paper. 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 shows that the proposed system outperforms them regarding the cryptographic performance metrics.*

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

## I. 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 to access platforms with highly 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.

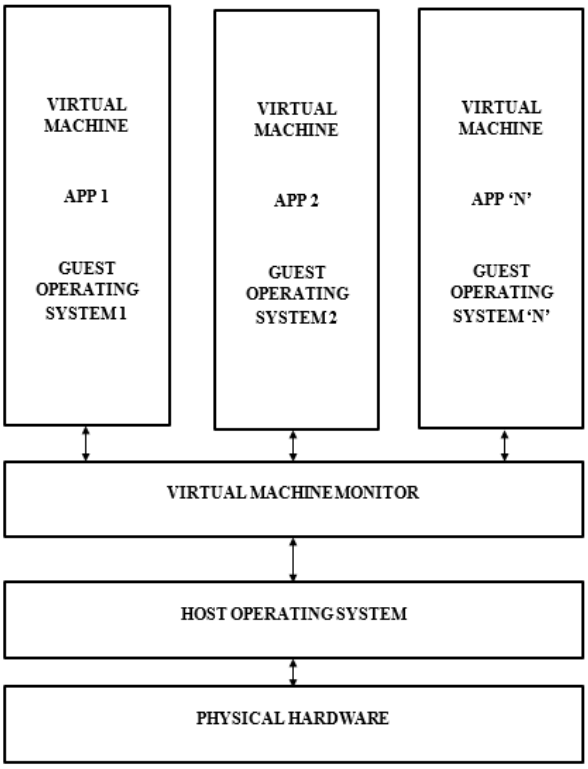


**Fig. 1.** Typical Illustration of a Cloud Platform Interacting with a Remote Client

In figure 1, multiple applications are being run by the cloud platform, 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. If user groups are accessing the cloud platform, 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 due to group members' frequent addition and deletion. In general, several cloud service providers may or may not have access to the internal functionality of the cloud architecture, which often leads to loophole cloud security that employ conventional cryptographic techniques for cloud security.

Cloud-based security groups describe areas where different security measures can be applied. Correctly implemented cloud-based security groups help limit unlawful access to IT resources in the event of a security infringe. The motivation behind this proposed work is to provide better security mechanism for the public cloud, which is an essential requirement in today's scenario. Cryptography is derived from the Greek word Krypto's, which means hidden. It is the practice and study of secure communications techniques that enables only the sender and intended recipient of a data/message to view its contents. Secure Communication is observed as a scenario where the data or message shared amid two parties can't be accessed by an adversary or hacker. The core principals of modern cryptographic algorithms are Data Integrity, Authentication, Data Confidentiality and Non-repudiation. Cryptographic algorithms are classified into two major categories: Symmetric Key Cryptography and Asymmetric Key Cryptography (popularly known as public key cryptography).

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 that can upload data to the cloud-based server in less time and high security to operate between the remote client and the server seamlessly. This work proposes a key generation approach to authenticate users in the group, and homographic encryption algorithms are employed to encrypt data to the cloud server.

The organization of this paper is as follows: Sect. 2 presents the system model for dynamic groups; Sect. 3 describes the proposed approach for homomorphic encryption. This section elaborates additive and multiplicative homomorphic encryption technique. The simulation setup and results are discussed in Sect. 4. The last part is Sect. 5, which concludes the research work done and experiences gained. The declarations are also mentioned in the last

## II. System Model For Dynamic Groups

Workgroups are generally dynamic, with members changing and/or members migrating to other groups [3]. Hence it is necessary to authenticate the members joining a group that has 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 severe challenge. Most of the intrusions generally occur at this level due to the ease of breakthrough into the network. However, the key should be renewed intermittently to decrease the chances of fraudulent intercepts. This is generally implemented based on a public key infrastructure (PKI) system to authenticate users. The PKI infrastructure is usually 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]. Blockchain technology is also found to be a trustworthy implementation for securing transactions in many domains [24-30]. Song Li et al. [31] proposed a public auditing scheme with the blockchain technology to resist the malicious auditors. 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.

## III. Proposed Approach For Homomorphic Encryption

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

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

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

### A. Additive Homomorphic Encryption:

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

$$n = p.q \tag{1}$$

Next, a key generator $G$ is designed such that:

$$G \in I^{+} \tag{2}$$

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 ciphertext (c) given by:

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

Here,
$E$ is the encryption algorithm.
$f$ stands for a function of.
$r$ is a chosen random number.

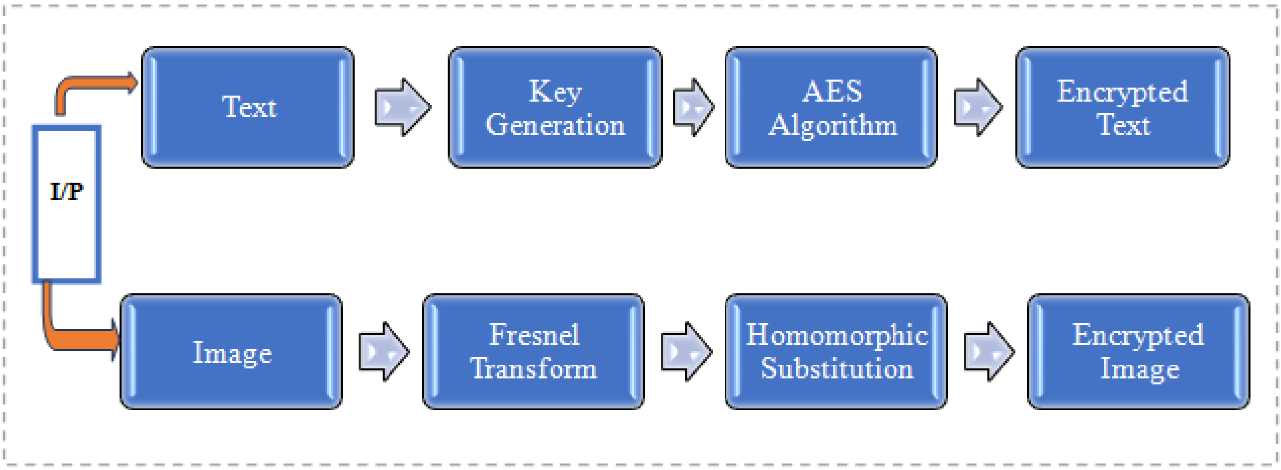


**Fig. 2.** Process Flow

Then, the following relations hold true:

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

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

$$\text{Or } k(p1,r1)k(p2,r2) = g^{p1+p2} r1r2^n modn^2 \quad (6)$$

$$\text{Or } k(p1,r1)k(p2,r2) = k(p_1 + p_2, r_1 r_2) \quad (7)$$

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).

## B. Multiplicative Homomorphic Encryption:

In this case, the following relations hold true:

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

$$\text{Or } (p_1 p_{2)})^b mod(n) = k(p1p2) \quad (9)$$

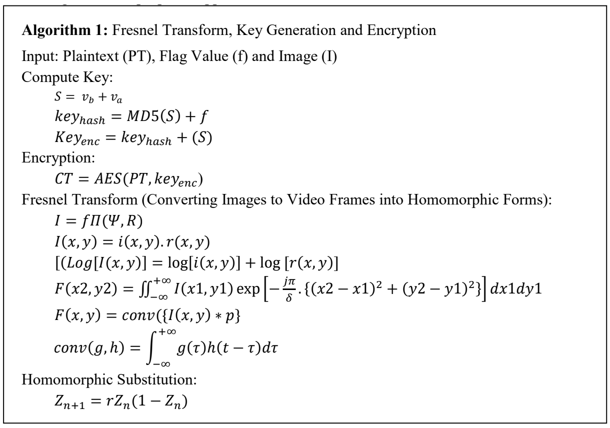
Equation (9) reveals that the separate plain texts can be obtained as a product of the individual plain texts with the same key (k).

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

## C. Proposed Technique

In Figure 2 the input is taken as a text as well as an image. After this a hash key is generated using MD5 and Flag Value and the text input is encrypted using AES. The image input is converted through Fresnel Transform into Homomorphic form and homomorphic Substitution algorithm. As an output encrypted text and image is generated.

The algorithm for proposed approach is used as mentioned below:



```
**Algorithm 1:** Fresnel Transform, Key Generation and Encryption
Input: Plaintext (PT), Flag Value (f) and Image (I)
Compute Key:
    $S = v_b + v_a$
    $key_{hash} = MD5(S) + f$
    $Key_{enc} = key_{hash} + (S)$
Encryption:
    $CT = AES(PT, key_{enc})$
Fresnel Transform (Converting Images to Video Frames into Homomorphic Forms):
    $I = f\Pi(\Psi, R)$
    $I(x,y) = i(x,y).r(x,y)$
    $[(Log[I(x,y)] = \log[i(x,y)] + \log[r(x,y)]$
    $F(x2,y2) = \iint_{-\infty}^{+\infty} I(x1,y1) \exp\left[-\frac{j\pi}{\delta}.\{(x2-x1)^2 + (y2-y1)^2\}\right] dx1dy1$
    $F(x,y) = conv(\{I(x,y) * p\}$
    $conv(g,h) = \int_{-\infty}^{+\infty} g(\tau)h(t-\tau)d\tau$
Homomorphic Substitution:
    $Z_{n+1} = rZ_n(1 - Z_n)$
```

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:

$$S = v_b + v_a \quad (10)$$

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

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

Here,
$v_b$ & $v_a$ are chosen byte file and attribute file, respectively.
$S$ is used to store the chosen byte file and attribute file.
$f$ 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:

$$CT = AES(PT, key_{enc}) \quad (13)$$

Here,
$CT$ is Cyphertext.
$PT$ is Plaintext.

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

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

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 3.

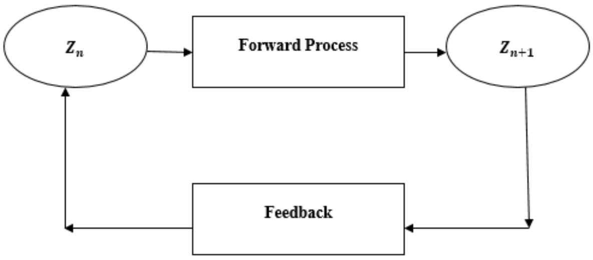


**Fig. 3.** Generation of The Random States Based on the Recursive Two-State Machine

Only the authentic user has the value of r and initial value , which uniquely determines the state .

This type of generator can generate random values using deterministic machines [8]. For converting images to video frames into homomorphic forms, the Fresnel Transform is used, and the process can be represented as [9]:

$$I = f\Pi(\Psi,R) \tag{15}$$

Here,
$I$ is the original image.
$\Psi$ is the illumination.
$R$ is the reflectance.
$\Pi$ represents the constant product operator.
$f$ represents a function of.

Typically, the constant illumination component and the high pass components can be separated using filters. A low pass filter is used to separate the illumination component, and a high pass filter separates the reflectance component [10].

The image intensity of such an image is given by:

$$I(x,y) = i\,(x,y).r\,(x,y) \tag{16}$$

Here,
$I$ is the image intensity which is a function of the coordinates (x, y).
$(x,y)$ are the pixel coordinates.
$i$ is the illumination function.
$r$ is the reflectance function.

Taking log on both sides:

$$[(Log[I(x,y)] = \log[i(x,y)] + [r(x,y)] \tag{17}$$

In general, the illumination component is similar in value for most images and generally has a lot of redundancy or redundant data [11].

The reflectance, however, varies significantly for different images. Thus, to avoid redundancy in the encrypted image, save space and reduce the size of the image, only the reflected component can be encrypted [12]. The illumination co-efficient can be embedded in the LSB positions of the encrypted data. The embedded data can be extracted from the LSB locations, and the complete image can be recreated [13].

The essential facts about this approach are:
1) Only the reflectance co-efficient is encrypted.
2) The reflectance co-efficient is again split into the Most Significant Bits (MSBs) and Least Significant Bits (LSBs).
3) The illumination co-efficient can be hidden in the LSB locations of the encrypted reflectance component.
4) Other secret images or data can also be hidden in the LSB locations to implement steganography within the encrypted image.
5) The essence of data hiding in LSB locations makes the hidden data extremely imperceptible, thereby giving no clue to attackers about hidden data.
6) The primary benefit of the techniques is the extraction of hidden data directly from the encrypted domain, not needing the decryption of the cover image.

The technique to convert standard images to homomorphic images is the Fresnel Transform which is mathematically given by [14]:

For an image $I(x,y)$,

$$F(x2, y2) = \iint_{-\infty}^{+\infty} I(x1, y1) \exp\left[-\frac{j\pi}{\delta}.\{(x2 - x1)^2 + (y2 - y1)^2\}\right] dx1dy1 \tag{18}$$

Here,
$F$ is the image in the Fresnel Domain.
$x$,y are the coordinates.
$I$ is the original image.
$\delta$ is the Transform parameter given by:

$$\delta = \lambda d \tag{19}$$

Here,
$\lambda$ is the wavelength.
$d$ is the separation between the image and the Fresnel plane.

The Fresnel transform is also given by the convolution integral of the image $I(x1,y1)$ and the term

$$\exp\left[-\frac{j\pi}{\delta}.\{(x2 - x1)^2 + (y2 - y1)^2\}\right]$$

which is also called the propagator function (p) [15].

Thus, the Fresnel transform can thus be computed as:

$$F(x,y) = conv(\{I(x,y)*p\}) \tag{20}$$

Here,
conv represents the convolution operation.
* represents the convolution operator.

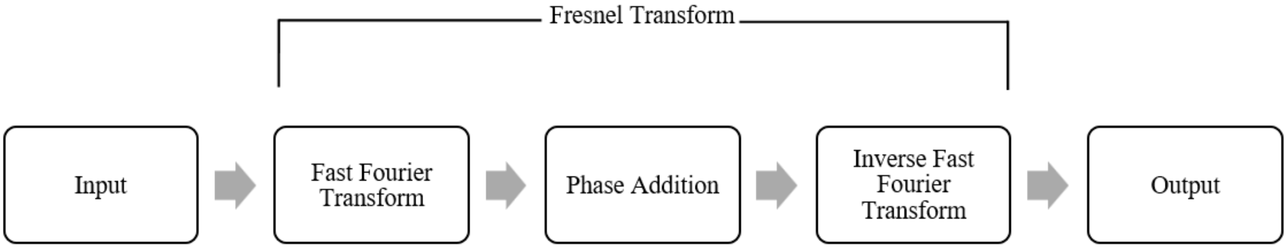


**Fig. 5.** Computation of Fresnel Transform Using the Fourier Transform [23]

```
Algorithm 2: Encryption for Image or Video Data
Input: $X \to g(i,j,k)$ // Where g denotes the function describing the dependence of the original image of
    $(i,j,k)$
Key Generation is Based as a Function Given By:
    $K_E = g_e(i,j,k)$ // Here, $g_e$ represents the encryption key generation algorithm.
The Encrypted Image is Generated Based on a Random and Adaptive Key and Image Parameters Given
    By:
    $Y \to z(I,\ K_E)$
The image degradation function is denoted as $f_D$.
The image restoration function is denoted as $f_R$.
The relation between $f_D$ and $f_R$ is Given By:
     $f_D = f_R^{-1}$
For Text Data, the Plain Text X is a Function of Only One Independent Variable to Argument, Given By:
    $X \to g(i)$
```

Without loss of generality, the convolution of any two functions g and h is given by:

$$conv(g,h) = \int_{-\infty}^{+\infty} g(\tau)h(t-\tau)d\tau \qquad (21)$$

Here,
$\tau$ is called the translator variable.

The physical visualization of Fresnel Transform is given in figure 3. The actual computational equivalent of the Fresnel Transform is shown in figure 4 [16]. While the Fresnel Transform can be computed directly but is complex on hardware, the Fresnel Transform is calculated indirectly using the modified version of the Fourier Method, which is computationally less complex from a hardware point of view [17].

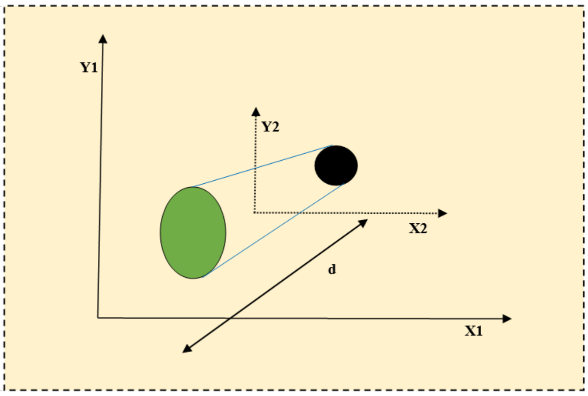


**Fig. 4.** Visualization of the Fresnel Transform [23]

The fundamental idea behind the working of the algorithms is implementing chaos and making it infeasible for attackers to decrypt the data through brute force [18]. This is possible if the encryption mechanism is one way (trapdoor). It is easy to compute in one direction but infeasible to compute its inverse by applying brute force to cryptanalysis [20]. The one-way function (trapdoor) for encrypting the data is expressed as:

$$Y = f(X) : \text{Easy to Compute.}$$

$$X = f^{-1}(Y) : \text{Infeasible to Compute by Brute Force.}$$

## D. Evaluation Parameters

The evaluation of the proposed system is based on the following metrics [21], [22]:

$$Throughput = \frac{Data\ Size\ Processed}{Time\ of\ Execution} \tag{22}$$

$$Avalanche\ Effect = \frac{No.of\ bits\ changed\ in\ Cipher\ Text}{No.of\ bits\ changed\ in\ Plain\ Text} \tag{23}$$

$$Mean\ Square\ Error\ (mse) = \frac{1}{n}\sum_{i=1}^{i=n}(P_{ia} - P_{id})^2 \tag{24}$$

Here,

$n$ is the number of bits.

$P_{ia}$ is the actual plain text.

$P_{id}$ is the deciphered plain text after decryption.

## IV. Simulation Results and Discussions

The simulation set up is presented through figure 6, which clearly shows that the user uploads the file to cryptography server and the server in turn encrypt the file using secure key and send back to user as well as upload it on cloud storage. Other users can also download the file with the help of secure key, provided to them securely. The cryptographic algorithm is implemented using Dotnet Framework on cryptographic server and user account details are stored on Microsoft SQL Server Database and Encrypted Image is uploaded on Microsoft Azure Cloud Platform.

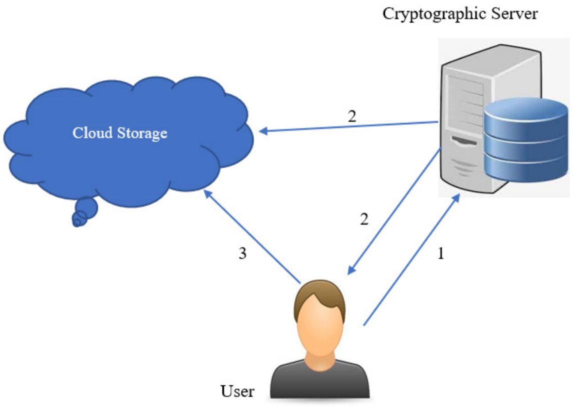


**Fig. 6.** Simulation Setup

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

The implementation of the chaotic network can be visualized using the Mandelbrot set, which exhibits a random chaotic nature. This complies with the one-way (trapdoor) function requirements in which the one-way transform computation is feasible, but the inverse calculation is infeasible using brute force. The generator set of the chaotic network (often termed as the Mandelbrot set) is depicted in figure 10.

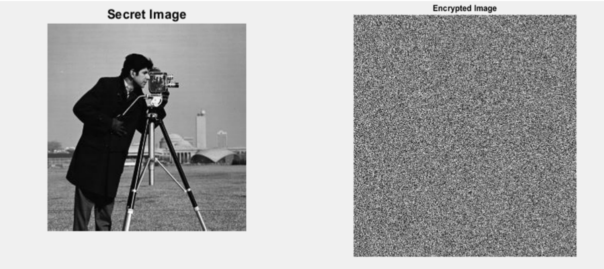


**Fig. 7.** Plain Text Image Vs Encrypted Image Data

Figure 7 depicts the plain text (image data), which is a standard test image (cameraman.jpg), its encrypted counterpart. For simulation, standard .jpg images with attributes of has been chosen. The encrypted image is shown to hide the original information in the image. Moreover, data can also be hidden in the encrypted image's least significant bits (LSB) locations. The data can be later extracted directly from the encrypted version of the image without the hard necessity to decrypt the image in the first place due to the homomorphic nature of the encryption. The essential facts about this approach are:

1) Only the reflectance co-efficient is encrypted.
2) The reflectance co-efficient is again split into the Most Significant Bits (MSBs) and Least Significant Bits (LSBs).
3) The illumination co-efficient can be hidden in the LSB locations of the encrypted reflectance component.
4) Other secret images or data can also be hidden in the LSB locations to implement data hiding within the encrypted image.
5) The essence of data hiding in LSB locations makes the hidden data extremely imperceptible, thereby giving no clue to attackers about hidden data.
6) The major benefit of the techniques is the extraction of hidden data directly from the encrypted domain, not needing the decryption of the cover image.

Thus, the proposed approach supports data hiding.

The performance of the proposed system is evaluated in terms of the throughput, execution time, and the avalanche effect. Multiple runs of the proposed algorithm for varying data sizes have been depicted in figure 8.

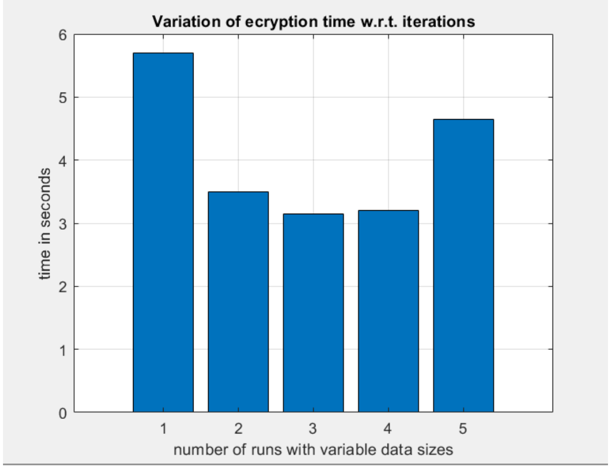


**Fig. 8.** Variation of Encryption Time w.r.t. Runs and Data Size

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.

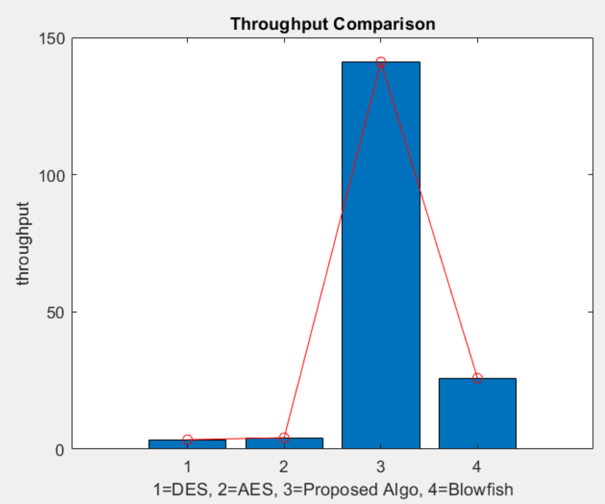


**Fig. 9.** Comparison of Throughput w.r.t. Standard Encryption Algorithms

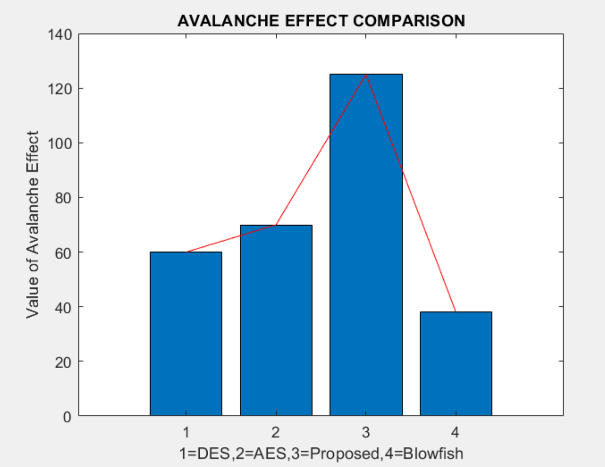


**Fig. 10.** Comparison of Avalanche Effect w.r.t. Standard Encryption Algorithms

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 throughput and avalanche effect values indicate that the proposed algorithm is fast and highly random. 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.

**Tab. 1.** Summary of Results

| Parameter | DES | AES | Blowfish | Proposed Approach |
|---|---|---|---|---|
| Throughput (Bits/Second) | 12 | 14 | 18 | 137 |
| Avalanche Effect (On 1-bit change) | 60 | 72 | 38 | 122 |

It can be observed that the proposed algorithm outperforms the existing approaches in terms of the throughput and avalanche effect criteria. Increased avalanche effect implies that the algorithm is more sensitive to changes in initial conditions, i.e., the plain text data. The increased throughput suggests that the proposed algorithm is NOT computationally complex to be implemented practically on hardware, making it suitable for a wide range of applications on the cloud. High-speed broadband internet access and the ubiquity of mobile devices make cloud computing a viable option for many domains everywhere. several factors are driving cloud adoption. The proposed technique will benefit the cloud environment directly or indirectly in multiple ways. The most important benefit is the safety and security of the cloud. The proposed security mechanism enables the dynamic groups of the public cloud to exchange the data efficiently and securely. Through the two-stage homomorphic encryption technique, the transactions within the cloud will become seamless and secure.

## V. Conclusions

The proposed work presents data security mechanisms for cloud environments 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 ciphertext is generated using the key and the AES algorithm. Homomorphic encryption is then implemented using the chaotic logistic map. The Mandelbrot set obtained also shown in the figure. The performance of the proposed algorithm is evaluated in terms of the avalanche effect, throughput, and execution time. The proposed approach is novel in terms of the efficient utilization of MD5 and AES to provide additional security for cloud environments. Also the work carried out in this paper contributes to the existing body of literature in the field of security and privacy. The obtained results indicate the superiority of the proposed algorithm terms of throughput, which is significantly high (137) and avalanche effects (122) as compare to the existing cryptographic techniques. The measures taken are very crucial and represent 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.

## VI. Declarations

**Funding:** Not applicable

**Conflicts of interest/Competing interests:** The authors declare that there is no conflict of interest.

**Availability of data and material:** Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

**Authors' contributions**: Sheenal Malviya, Sourabh Dave contributed to the design and implementation of the research. Kailash Chandra Bandhu and Ratnesh Litoriya performed the analysis of the results and to the writing of the manuscript. All authors discussed the results and contributed in the manuscript preparation.

## AUTHORS

**Sheenal Malviya** – Medi-Caps University, Indore, (M.P.) India, email: malviyasheenal7@gmail.com

**Sourabh Dave** – Medi-Caps University, Indore, (M.P.) India, email: sourabh.dave@gmail.com

**Kailash Chandra Bandhu** – Medi-Caps University, Indore, (M.P.) India, email: kailashchandra.bandhu@gmail.com

**Ratnesh Litoriya*** – Medi-Caps University, Indore, (M.P.) India, email: litoriya.ratnesh@gmail.com

## 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 Society 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] M.P. Babitha, K.R.R. Babu, "Secure cloud storage using AES encryption", *Proceedings in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 859–864.

[8] E. Tanyildizi and F. Özkaynak, "A New Chaotic S-Box Generation Method Using Parameter Optimization of One Dimensional Chaotic Maps," in *IEEE Access,* vol. 7, pp. 117829–117838, 2019.

[9] M. Alloghani, M.M. Alani, D. Al-Jumeily, T. Baker, "A systematic review on the status and progress of homomorphic encryption technologies", *Journal of Information Security and Applications*, Elsevier 2019, Volume 48, 102362.

[10] L. Liu, F. Nie, A. William, Z. Li, T. Zhang, and B. C. Lovell, "Multi-Modal Joint Clustering With Application for Unsupervised Attribute Discovery," in *IEEE Transactions on Image Processing* 2018, vol. 27, no. 9, pp. 4345–4356.

[11] F. Wang, J. Wang, R. Ni, Z. Zhu, and Y. Hu, "Resolution Adaptative Network for Cryptanalysis of Asymmetric Cryptosystems," in *IEEE Access*, vol. 8, pp. 187419–187430, 2020.

[12] Z. Shao, X. Liu, Q. Yao, N. Qi, Y. Shang, J. Zhang, "multiple-image encryption based on chaotic phase mask and equal modulus decomposition in quaternion gyrator domain", *Signal Processing and Communication*, Elsevier 2020, Volume 80, 115662.

[13] G. Luan, A. Li, D. Zhang, and D. Wang, "Asymmetric Image Encryption and Authentication Based on Equal Modulus Decomposition in the Fresnel Transform Domain," in *IEEE Photonics Journal*, vol. 11, no. 1, pp. 1–7, Feb. 2019.

[14] Y. Kim, M. Sim, I. Moon, and B. Javidi, "Secure Random Phase Key Exchange Schemes for Image Cryptography," in *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10855–10861.

[15] D. Blinder, C. Schretter, H. Ottevaere, A. Munteanu, and P. Schelkens, "Unitary Transforms Using Time-Frequency Warping for Digital Holograms of Deep Scenes," in *IEEE Transactions on Compu-*

*tational Imaging*, vol. 4, no. 2, pp. 206–218, June 2018.

[16] M. Khurana and H. Singh, "Asymmetric optical image encryption using random Hilbert mask based on fast Walsh Hadamard transform," 2017 *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 2017, pp. 374–377.

[17] L. Zhao, J. T. Sheridan, and J. J. Healy, "Unitary Algorithm for Nonseparable Linear Canonical Transforms Applied to Iterative Phase Retrieval," in *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 814–817, June 2017.

[18] P.S. Goswami, T. Chakraborty, "Design of a Quantum One-Way Trapdoor Function", *Emerging Technology in Modelling and Graphics*, Springer 2020, vol. 937, pp. 547–555.

[19] N. Döttling, S. Garg, Y. Ishai, G. Malavolta, T. Mour, "Trapdoor hash functions and their applications", *Advances in Cryptology*, Springer 2019. *Lecture Notes in Computer Science*, vol. 11694, pp. 3–31.

[20] G. Luan, Z. Chen, C. Huang, "Silhouette-free multiple-image encryption using coherent superposition and Fresnel transform", *Optic*, Elsevier 2020, Vol. 224, 165498.

[21] 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.

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

[23] L. M. Bernardo and O. D. D. Soares, "Fractional Fourier transforms and optical systems," *Optics Communications*, Vol. 110, no. 5–6, pp. 517–522, 1994.

[24] P. Pandey and R. Litoriya, "Securing E-health networks from counterfeit medicine penetration using Blockchain," *Wireless Personal Communications*, Vol. 117, issue 1, pp. 7–25, 2020.

[25] P. Prateek and L. Ratnesh, "Promoting Trustless Computation through Blockchain Technology," *National Academy Science Letters*, Vol. 44, pp. 225–231, 2020.

[26] P. Prateek and L. Ratnesh, "Securing and authenticating healthcare records through blockchain technology," *Cryptologia*, Vol. 44, no. 4, pp. 341–356, 2020.

[27] P. Pandey and R. Litoriya, "Implementing healthcare services on a large scale: challenges and remedies based on blockchain technology," *Health Policy and Technology*, Vol. 9, no. 1, pp. 69–78, Jan. 2020.

[28] S. Soner, R. Litoriya, and P. Pandey, "Exploring Blockchain and Smart Contract Technology for Reliable and Secure Land Registration and Record Management," *Wireless Personal Communications*, Vol. 121, Issue 1, pp. 2495–2509 Aug. 2021.

[29] P. Pandey and R. Litoriya, "Ensuring elderly well being during COVID-19 by using IoT," *Disaster Medicine and Public Health Preparedness*, pp. 1–10, Oct. 2020.

[30] P. Pandey and R. Litoriya, "Technology intervention for preventing COVID-19 outbreak," *Information Technology & People*, vol. 34, no. 4, pp. 1233–1251, May 2021.

[31] Song Li, Jian Liu, Guannan Yang, Jinguang Han, "A Blockchain-Based Public Auditing Scheme for Cloud Storage Environment without Trusted Auditors", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8841711, 13 pages, 2020. https://doi.org/10.1155/2020/8841711