A Novel Scheme for Encryption and Decryption of 3D Point and Mesh Cloud Data in Cloud Computing

M.P. Rajakumar*, J. Ramya[*], B. Uma Maheswari*, R. Sonia**

**Department of Computer Science and Engineering,*
St. Joseph's College of Engineering, Chennai 600 119,
Tamilnadu, India. (e-mail: rajranjhu@gmail.com)
***Department of Research and Development,*
Airgate Technologies, OMR, Chennai 600 119,
Tamilnadu, India (e-mail: sonia.j25@gmail.com)

Abstract: With the rapid evolution of Internet technology, cloud computing has taken a major role in managing large amounts of data. The major concerns in this domain are security and privacy. Therefore, attaining a reliable level of confidentiality in the cloud computing environment is a pivotal task. Among different types of data stored in the cloud, the 3D point and mesh cloud data are increasingly popular in recent days, due to the growth of 3D modelling and 3D printing technologies. Hence, in this research, we propose a novel scheme for preserving the privacy of 3D point and mesh cloud data. Chaotic map-based data encryption is a recently trending research area due to its unique properties like pseudo-randomness, deterministic nature, sensitivity to initial conditions, ergodicity, etc. Hence, chaotic systems can be regarded as systems of deterministic randomness. We exploit these properties of chaotic systems to encrypt the 3D point and mesh cloud data. To boost encryption efficiency significantly, in this work, we propose a novel chaotic map. The sequence generated by this map is used to transform the coordinates of the cloud data. The improved range of the proposed map is depicted using bifurcation analysis. The quality of the proposed chaotic map is also analyzed using metrics like Lyapunov exponent and approximate entropy. We also demonstrate the performance of the proposed encryption framework using attacks like brute-force attack and statistical attack. The experimental results clearly depict that the proposed framework produces the best results compared to the previous works in the literature.

Keywords: Chaotic map, cloud computing, encryption, 3D point cloud, 3D mesh.

1. INTRODUCTION

With the increasing popularity of internet usage, 3D point (Yuan et al., 2017) and 3D mesh (Guo et al., 2015) data representations are widely being used for the representation of objects. Applications like Autodesk123D Catch capture the photograph of objects from different angles and transmit them to remote cloud-based servers. This data is then reconstructed to form 3D models of the objects and are transmitted to the users. There are also numerous desktop applications for editing the 3d point and mesh cloud data. Recently, the Virtual Reality (VR) technology enables the users to experience the virtual reality 3D environment. The 3D models are of two types namely, the 3D solid model and the 3D shell model. The 3D solid model represents the volume of the object whereas the 3D shell model represents the boundary region or the surface of the object. The 3D shells include 3D point and 3D mesh models. These models are widely used in 3D graphics. Most of the computer games make use of these models. Thus, it is evident that the 3D point and mesh data have taken a role in our day-to-day lives.

However, the main issue faced by these data is the privacy issue since they are stored in the cloud. Thus, encryption of these data is a vital task. These 3D data are massive and multi-dimensional. Also, they have a high correlation among the neighbouring points. Thus, traditional encryption algorithms like Rivest, Shamir, and Adleman (RSA) (Andalib et al., 2014), Advanced Encryption Standard (AES) (Dworkin 2001), Data Encryption Standard (DES)(Grabbe, 1992) and blowfish (Valmik and Kshirsagar, 2014), Twofish (Solomon and Be, 2016), Elliptic Curve Cryptography (ECC) (Fernandas, 1999), ElGamal encryption (Schnorr and Jakobsson, 2000), Diffie-Hellman key exchange (Steiner et al., 1996) etc., may not be sufficient to meet the security issues of 3D data.

Hence researchers have recently developed numerous encryption schemes based on chaotic maps. The deterministic random nature of chaotic maps (Turcotte, 2012) has made them popular among various encryption frameworks that include image and video encryptions. Also, the properties of chaos (Huang and Nien, 2019; Lian et al., 2005) like sensitivity to initial conditions ergodicity and random nature have made them produce good encryption results compared to conventional cryptographic schemes. A system for encryption of 3D solid models was presented in (Del Rey, 2015). In this research, we present a novel scheme for the encryption of 3D cloud point and 3D mesh data.

The overall contributions of this paper are threefold:

a) A novel chaotic map to produce chaotic sequence for encryption.
b) A novel two-level encryption framework for the encryption of 3D point cloud and 3 D mesh cloud data.
c) Evaluation of the proposed encryption scheme and comparison with the state-of-the-art frameworks.

The rest of the paper is organized as follows. Section 2 includes a detailed literature survey of the previous works in the literature. Section 3 explains the generation of chaotic sequences using the proposed chaotic map. Section 4 describes the proposed encryption methodology. The results and discussion are performed in Section 5. Section 6 concludes the paper.

2. RELATED EARLIER WORKS

Extensive research has been carried out in the field of cloud encryption schemes in the literature. In this section, we discuss some of these research works.

A review of various schemes for securing user data in cloud computing based on encryption algorithms was proposed in (Arora and Parashar, 2013). In this research, the security issues faced by cloud computing, mechanisms used, the challenges faced are reviewed in detail. In addition, various security algorithms like RSA, AES, DES and blowfish algorithms were implemented and analyzed in this paper. A framework for encrypting health records in cloud computing was proposed in (Li et al., 2013). In this work, a patient-centric scheme was proposed in which attribute-based encryption was performed. This system achieved a high degree of security by utilizing multi-authority encryption. A proxy-based encryption scheme for cloud storage was proposed in (Zhang et al., 2019). In this scheme, a proxy is authorized by the sender for data encryption. This encrypted data is uploaded to the cloud. This framework is based on lattice-based cryptography. The system was proved to achieve security against the misbehaved cloud servers.

Cloud-assisted encryption was presented in (Zhang et al., 2019). This system was designed for the industrial internet of things applications. A high level of security against quantum attacks was achieved by this encryption scheme. Also, this encryption algorithm used a low communication overhead and had low computational cost. A new system for the protection of cloud storage against guessing keyword attack was proposed in (Y Zhang et al., 2019). Since key words possess low entropy, they are prone to be attacked by the keyword guessing attack. Hence, in this work, the authors proposed a system in which the key words were encrypted using dedicated key servers. The keys in the key servers are periodically updated. This framework was capable of resisting online keyword guessing attacks with good security. To enhance the security of cloud images, a scheme using biometric authentication was presented in (Kakkad et al., 2019). In this paper, encryption is done on two levels. In the first level, compression is done using discrete wavelet transform. At the second level, encryption is done by using a hybrid combination of a secure hash algorithm and blowfish algorithm.

Cloud encryption framework using attribute-based encryption was proposed in (Li et al., 2019). In this paper, collusion prevention was achieved. The update parameters are generated by the data owner. This new file update scheme helps to achieve reduced storage cost and communication expenses. An edge-based framework for privacy protection was introduced in (Wang et al., 2020). Here, instead of transmitting the entire data to the cloud, a part of the data is first stored in the edge servers. This helps to protect the data from the cloud leakage issue since the original data cannot be retrieved from the cloud data. This differential storage technique is aided in achieving good encryption performance.

Homomorphic encryption was utilized in (Alabdulatif et al., 2020) for achieving the security of big data stored in the cloud. In this work, different cloud notes were enabled and segregated for performing computational analysis of different parts of the data. These nodes were made to operate independently. Thus, the performance of this system was found to be better than the encryption using a single cloud computing node. Encryption schemes based on chaos theory are popularly used in recent days. A new encryption scheme using chaos theory and a single round dictionary was proposed in (Liu et al., 2019). In this work, the compressive sensing theory was utilized to achieve simultaneous compression and encryption. The measurement matrix used for encryption is a chaotic logistic sequence. Quantification was achieved using a sigmoid function. A single round dictionary was used as a substitute for discrete cosine transform (DCT) basis function. Thus, a different unique dictionary was generated for each image. This helped to achieve good encryption performance.

Encryption of selective data based on steganography using chaos theory was proposed in (Xiang et al., 2015). Here, the secret image to be encrypted was embedded into another image and transferred to the cloud. In the cloud, this image was encrypted using chaos theory. The encrypted image was then sent back to the user. The user then extracts the secret image and decrypts it. In this way, the secret data is not revealed to the cloud. A new encryption scheme based on three chaotic maps was presented in (Mishra et al., 2014). In this paper, initially, the image is encrypted using a two-dimensional Cat map. Then, at the second level, the image is encrypted using two-dimensional logistic map. Finally, at the third level, the image is encrypted using a one-dimensional logistic map. For all three levels, the same keys were utilized for encryption.

Logistic map was utilized for encryption in (Pareek et al., 2006). In this work, an 80-bit secret key and a pair of logistic maps were employed for image encryption. Here, eight different groups of operations were framed, and the outcome of the logistic map was used to select one of the eight groups. In addition, after every encryption, the key is modified to achieve a high level of security. Encryption scheme using chaotic tent map was proposed in (Som et al., 2019). Here, every bit plane of the image is categorized into two groups namely, the significant and the non-significant group. Thus, the computational cost was greatly lowered in this work. It

was found that this scheme achieved around 35% reduction in computational complexity.

Encryption using hyperchaotic maps were proposed in (Patro et al., 2019). Two types of permutations were performed. Initially, the data was encrypted using block permutations followed by the bit permutations. Finally shuffling was performed at the bit level to enhance the security. Similarly, during the diffusion process, the bit level diffusion was performed first. Then the pixel diffusion was performed. A hash value having a length of 256 bits was used for encryption. A new scheme for the encryption of 3D point cloud data was proposed in (Jin et al., 2016). In this work, two types of encryptions were proposed, analyzed and compared. The first scheme was based on the generation of random sequences using logistic chaotic mapping. The second scheme was based on the projection of the coordinates of the 3D cloud data points using a transformation matrix. This matrix was generated using a rotation matrix and a translate vector.

3D cloud data encryption using the sequences generated from the Cat map was proposed in (Jia et al., 2019). Here, the sequence generated by Cat chaotic maps was used for two types of encryptions. The first framework was based on sorting the sequences and shuffling the locations of the 3D data based on the sorted sequences. The second framework was based on the transformation of the 3D location using a transform matrix multiplication. These schemes were analyzed using security analysis tests like key space analysis, statistical analysis, etc. Chaotic Henon map was utilized for encryption of medical data in (Huang et al., 2013). In this work, the Haar wavelet transform was used to embed the cover images. The images in the spatial domain were converted to the frequency domain. In this domain, the data was modified using the random sequences generated by Henon map. This technique helped to achieve reversible data hiding with a good level of security.

A new encryption scheme for encryption of 3D mesh data was proposed in (Liang et al., 2019). Each mesh data contains many facets and each facet contains three vertices. And each vertex contains three coordinates. Using the coordinates of the mesh data, discrete cosine transform was applied. Then the float values were normalized. The public key encryption was performed using the RSA algorithm. Asymmetric encryption system was used for 3D mesh encryption. Double encryption was performed using the message digest to achieve high encryption performance. In addition, two objectives were focused namely, the shape error function and the computational efficiency function. These two objectives were optimized using the proposed asymmetric encryption framework. A scheme for fast encryption of 3D data was proposed in (Wang et al., 2019). In this paper, random points were used to achieve the confusion of the 3D coordinates. Next, diffusion was achieved using XOR operation. Elgamal encryption algorithm is proposed to provide multiagent security (*) The proposed framework evaluates the performances of cryptographic algorithms based on threshold values. [*]

3. CHAOTIC SEQUENCE GENERATION

3.1 Chaos Theory

Randomness generated by a deterministic system is mathematically defined as chaos. Chaos is a universal phenomenon that is based on nonlinearity. The sequence generated by these maps can be predicted only if the initial and control parameters are known. Else, the system appears to be completely random. Even a very small change in these values produces a completely different sequence. Therefore, in the field of encryption, chaos theory is popularly being used.

Chaotic maps are designed based on chaos theory. They are categorized into two: 1D and higher dimensional maps. The ID maps are simple single-dimensional sequences that are generated based on the two control parameters. Commonly used ID maps are logistic map was proposed in (R. M. May 1976) and logistic sine map was proposed in (J. Feng et al., 2016). In this work, we propose a new 1D chaotic map for the encryption of cloud 3D point and mesh data. Since the sequence generated by these maps completely depends on only two parameters, the physical realization of these maps through electric circuits is feasible. Since chaotic maps generate random sequences in a deterministic manner, we have employed them for encryption in our work.

The properties of a chaotic system are as follows:

3.1.1. Ergodicity

Chaotic system produces sequences that have the same properties over time as averaged over the space of all the system states.

3.1.2. Randomness

The sequence generated by a chaotic map is completely random and has random properties similar to that of random matrices like the Gaussian random matrix, Bernoulli random matrix, etc.

3.1.3. Sensitivity to parameters

Chaos system is extremely sensitive to its parameters like the initial value and control parameter. A very small change in these values generates a completely different random sequence.

3.2 Metrics to validate chaotic nature

The two commonly used metrics to validate the chaotic nature of these maps are the Lyapunov exponent and the approximate entropy.

3.2.1. Lyapunov exponent (LE)

Lyapunov exponent (Wolf et al., 1985) is a popularly used metric for quantification of the chaos in a chaotic map. It evaluates the average divergence between two trajectories that are obtained with two different initial values that are close to each other. It is mathematically defined as,

$$LE = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \log \left| \frac{dx_{n+1}}{dx_n} \right| \quad (1)$$

A positive value of Lyapunov exponent indicates that the two trajectories generated by the map will diverge exponentially with respect to time, whereas a negative value of Lyapunov exponent indicates that the two trajectories will overlap at some point of time. Also, the larger the value of LE, the greater is the chaotic nature of the sequence produced by a map.

3.2.2. Approximate entropy (AE)

Approximate entropy (Pincus, 1995) is also used for the quantitative representation of the chaotic nature of chaotic maps. Higher values of AE indicate that the complexity of the chaotic sequence is very high.

3.3 Logistic Map

The logistic map is defined as:

$$x_{n+1} = \mu x_n(1 - x_n) \quad (2)$$

where $\mu \in [0,4]$ is the control parameter and $x_0 \in [0,1]$ is the initial condition. From Fig 1, we find that the proposed map is chaotic in the range$\mu \in [3.57,4]$.

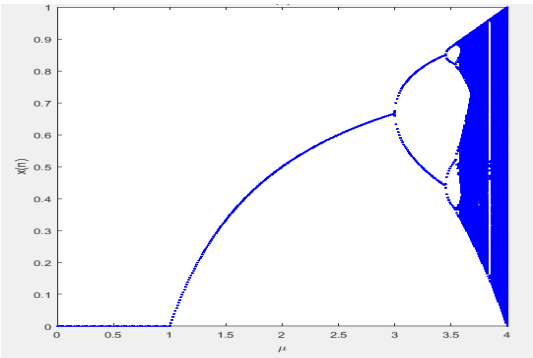

Fig. 1. Bifurcation of logistic map.

To further depict the chaotic nature of the logistic map, the Lyapunov exponent of the logistic map is shown in Fig 2. In the Lyapunov exponent graph, the region which is positive refers to the chaotic region. From Fig2, it is evident that the logistic map is chaotic in the region $\mu \in [3.57,4]$.

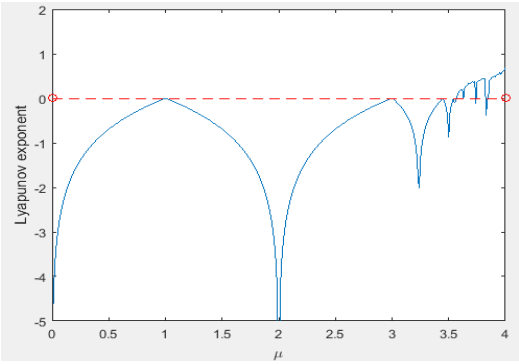

Fig. 2. Lyapunov exponent of logistic map.

3.4 Logistic Sine Map

The logistic sine map is defined as:

$$x_{n+1} = \mu x_n(1 - x_n) + \frac{(4-\mu)}{4} sin(\pi x_n) \quad (3)$$

where $\mu \in [0,4]$ is the control parameter and $x_0 \in [0,1]$ is the initial condition. From Fig 3, we find that the logistic sine map is fully chaotic in the range $\mu \in [0,4]$.

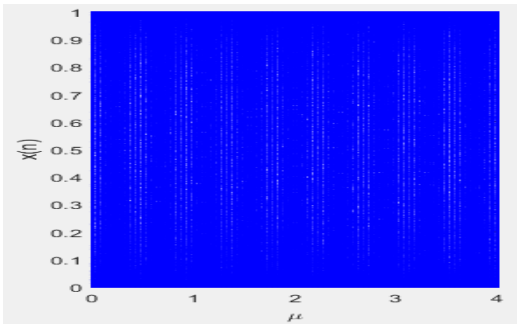

Fig. 3. Bifurcation of logistic sine map.

Similar to the logistic map, the Lyapunov exponent of the logistic sine map is plotted in Fig 4. From Fig 4, we see that the logistic sine map is completely chaotic in the region $\mu \in [0,4]$. However, the highest value of LE attained is only 0.6933 when $\mu = 4$.

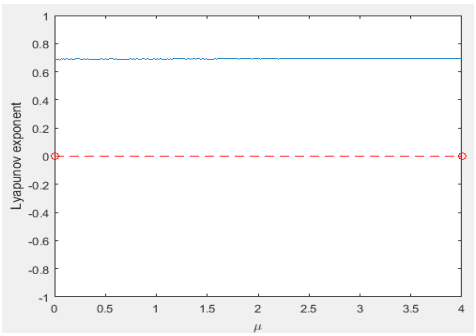

Fig. 4. Lyapunov exponent of logistic sine map.

3.5 Proposed Chaotic Map

The proposed chaotic map is defined as:

$$x_{n+1} = ((7000 - \mu)/7000) * sin(8\pi x_n) \quad (4)$$

where $\mu \in [0,75]$ is the control parameter and $x_0 \in [0,1]$ is the initial condition. From figure 5, we find that the proposed map is chaotic in the range$\mu \in [0,75]$. This range is very much greater than that of the logistic map and logistic sine map. The sequences generated by the proposed map is used for encryption and decryption of the point cloud and mesh cloud data. In addition, the initial 1000 values generated using aparticular key $K = \{x_{0_i}, \mu_i\}$ are ignored to avoid the transient effect.

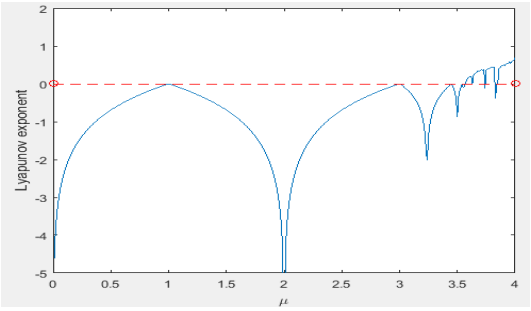

Fig. 5. Bifurcation of proposed chaotic map.

Similar to the logistic map and logistic sine maps, the Lyapunov exponent of the proposed map is plotted in Fig 6. From Fig 6, we see that the proposed map is completely chaotic in the region$\mu \in [0,75]$. However, the highest value of LE attained is 1.3881 when $\mu = 75$.

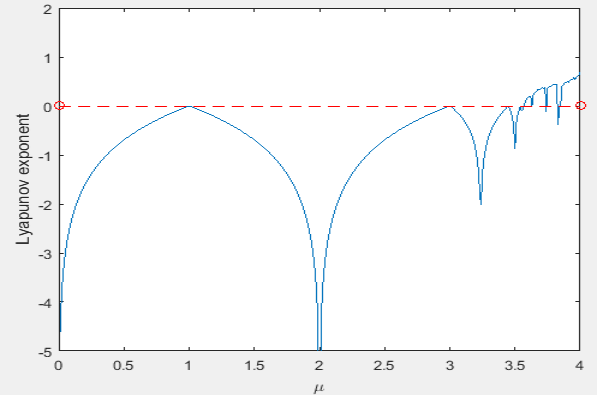

Fig. 6. Lyapunov exponent of proposed chaotic map.

Also, to prove the chaotic properties of the proposed map quantitatively Lyapunov exponent and approximate entropy are evaluated and shown in Table 1. Table 1 show the LE and AE values of the proposed map and other existing maps. It can be inferred from the tabulated values that the LE and AE values of the proposed map are high compared to other maps indicating that the proposed map has better chaotic behaviour compared to other existing maps.

Table 1. Comparison of proposed chaotic maps with logistic sine maps.

Map	Map Equation	LE	AE
Logistic	$x_{n+1} = \mu x_n(1 - x_n)$	0.6933	0.7142
Logistic Sine Map	$x_{n+1} = \mu x_n(1 - x_n) + \frac{(4-\mu)}{4} sin(\pi x_n)$	0.6933	0.7142
Proposed Map	$x_{n+1} = (7000 - \mu)/7000\, sin(8\pi x_n)$	**1.3881**	**1.8762**

4. PROPOSED ENCRYPTION METHODOLOGY

This section describes how cloud encryption is performed using the sequence generated by the proposed chaotic map. The keys used for encryption and decryption are securely transmitted between the sender and the receiver. In addition, this technique utilized very few sets of keys hence the problem of data leakage is minimized. The security achieved by the proposed scheme is too high since we have utilized a two-level encryption algorithm. In the first level, the sequences generated by chaotic maps are sorted in ascending order to shuffle the coordinates of the cloud data. In the second level, the sequences generated by chaotic maps are sorted in descending order to further shuffle the coordinates of the cloud data. Thus, double encryption is achieved using ascending sort (AS) and descending sort (DS).

4.1. 3D Point Cloud Model

The 3D point cloud model data comprises a 3-dimensional coordinate system. That is, each point consists of 3 coordinates. Also, the proposed scheme consists of a double encryption methodology. Hence, to encrypt this data, we generate six different random sequences from the proposed chaotic map. These six sequences are generated using 6 chaotic keys referred as$PK_1, PK_2, \ldots, PK_6$. Here, each key refers to a pair of key parameters which are the initial value and the control parameter.

4.1.1. Encryption of 3D Point Cloud Model

The steps involved in the encryption of the 3D point cloud model are shown in Algorithm 1. Initially using the first three keys, three sequences are generated. The sequences are sorted using ascending order. The details of the original and the new locations of the sequences are then stored. Using these stored locations, the locations of the point cloud data $P_1, P_2, \ldots, P_n$ are swapped to obtain intermediate point cloud data$IP_1, IP_2, \ldots, IP_n$. Using the next three chaotic keysPK_4, PK_5, PK_6, again three new sequences are generated. Using these sequences once again sorting is performed using descending order. The details of the original and new locations are then stored. Now the intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$ is again swapped based on this location information to obtain encrypted point cloud data$EP_1, EP_2, \ldots, EP_n$. The entire process is depicted in Fig7.

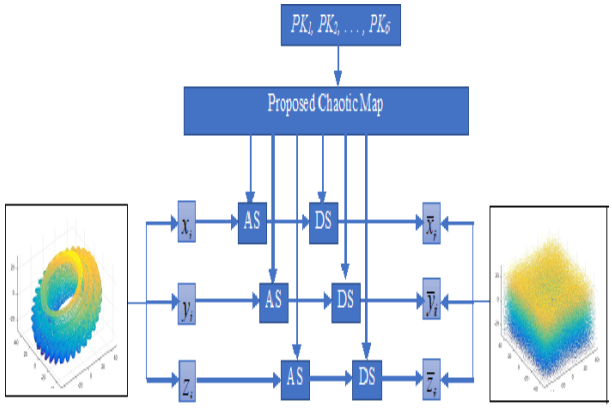

Fig. 7. Proposed two-level encryption scheme for point cloud data.

Algorithm 1: Encryption of 3D Point Cloud

Input:
Original point cloud $P_1, P_2, \ldots, P_n$ where $P_i = \{x_i, y_i, z_i\}$
Chaotic keys $PK_1, PK_2, \ldots, PK_6$ where $PK_i = \{x_{0_i}, \mu_i\}$.

Output:
Encrypted point cloud $EP_1, EP_2, \ldots, EP_n$ where $EP_i = \{\bar{x}_i, \bar{y}_i, \bar{z}_i\}$.

Steps:

1. Using the chaotic keys PK_1, PK_2, PK_3 generate three chaotic sequences S_1, S_2, S_3.
2. Sort the chaotic sequences in ascending order and store the new location of each value.
3. Using the stored locations, swap the locations of the point cloud data $P_1, P_2, \ldots, P_n$ to obtain intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$.

4. Now, using the chaotic keys PK_4, PK_5, PK_6 generate three new chaotic sequences S_4, S_5, S_6.
5. Sort the new chaotic sequences in descending order and store the new location of each value.
6. Using the stored locations swap the locations of the intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$ to obtain encrypted point cloud data $EP_1, EP_2, \ldots, EP_n$.

4.1.2 Decryption of 3D Point Cloud Model

The decryption of 3D point cloud is done to reverse the effect of encryption and to get back the original data $P_1, P_2, \ldots, P_n$ from the encrypted data $EP_1, EP_2, \ldots, EP_n$. This is given in Algorithm 2. Here, using the chaotic keys PK_4, PK_5, PK_6, three sequences namely S_4, S_5, S_6 are generated. Then, these sequences are sorted in descending order and their location details are stored. Based on this information, the encrypted data is swapped to obtain the intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$. Now, using the chaotic keys PK_1, PK_2, PK_3 three new chaotic sequences S_1, S_2, S_3 are once again generated. The new chaotic sequences are again sorted in ascending order and the new location of each value is then stored. Using this information, the intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$ is swapped to get the original data $P_1, P_2, \ldots, P_n$. This process is illustrated in Fig 8.

Algorithm 2: Decryption of 3D Point Cloud

Input:
Encrypted point cloud $EP_1, EP_2, \ldots, EP_n$.
Chaotic keys $PK_1, PK_2, \ldots, PK_6$.
Output:
Original point cloud $P_1, P_2, \ldots, P_n$.
Steps:
1. Using the chaotic keys PK_4, PK_5, PK_6 generate three chaotic sequences S_4, S_5, S_6.
2. Sort the chaotic sequences in descending order and store the new location of each value.
3. Using the stored locations, swap the locations of the encrypted point cloud data $EP_1, EP_2, \ldots, EP_n$ to obtain intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$.
4. Now, using the chaotic keys PK_1, PK_2, PK_3 generate three new chaotic sequences S_1, S_2, S_3.
5. Sort the new chaotic sequences in ascending order and store the new location of each value.
6. Using the stored locations swap the locations of the intermediate point cloud data $IP_1, IP_2, \ldots, IP_n$ to obtain original point cloud data $P_1, P_2, \ldots, P_n$.

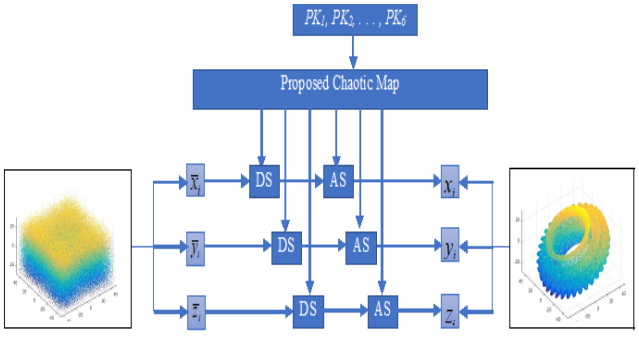

Fig. 8. Proposed two-level decryption scheme for point cloud data.

4.2 3D Mesh Cloud Model

The 3D mesh cloud model data comprises of many facets $M_1, M_2, \ldots, M_n$. Each facet has three vertices $M_i = \{V_1^i, V_2^i, V_3^i\}$ and each vertex is represented by a 3-dimensional coordinate system. That is, each point consists of 3 coordinates. It is denoted as $V_1^i = \{x_1^i, y_1^i, z_1^i\}$, $V_2^i = \{x_2^i, y_2^i, z_2^i\}$ and $V_3^i = \{x_3^i, y_3^i, z_3^i\}$. Also, the proposed scheme consists of a double encryption methodology. Hence, to encrypt this data, we generate 18 different random sequences from the proposed chaotic map. These 18 sequences are generated using 18 chaotic keys referred as $MK_1, MK_2, \ldots, MK_{18}$. Here, again each key refers to a pair of key parameters which are the initial value and the control parameter.

4.2.1. Encryption of 3D Mesh Cloud Model

The steps involved in the encryption of 3D mesh cloud model are shown in Algorithm 3. Initially using the nine chaotic keys $MK_1, MK_2, \ldots, MK_9$, nine sequences are generated. The sequences are sorted using ascending order. The details of the original and the new locations of the sequences are then stored. Using these stored locations, the locations of the mesh cloud data $M_1, M_2, \ldots, M_n$ are swapped to obtain intermediate point cloud data $IM_1, IM_2, \ldots, IM_n$. Using the next nine chaotic keys $MK_{10}, MK_2, \ldots, MK_{18}$, again nine new sequences are generated. Using these sequences once again sorting is performed using descending order. The details of the original and new locations are then stored. Now the intermediate point cloud data $IM_1, IM_2, \ldots, IM_n$ is again swapped based on this location information to obtain the encrypted mesh cloud data $EM_1, EM_2, \ldots, EM_n$. The entire process is depicted in Fig 9.

Algorithm 3: Encryption of 3D Mesh Cloud

Input:
Original mesh cloud $M_1, M_2, \ldots, M_n$ where $M_i = \{V_1^i, V_2^i, V_3^i\}$.
Here, $V_1^i = \{x_1^i, y_1^i, z_1^i\}$, $V_2^i = \{x_2^i, y_2^i, z_2^i\}$ and $V_3^i = \{x_3^i, y_3^i, z_3^i\}$.
Chaotic keys $MK_1, MK_2, \ldots, MK_{18}$ where $MK_i = \{x_{0_i}, \mu_i\}$
Output:
Encrypted mesh cloud $EM_1, EM_2, \ldots, EM_n$ where $EM_i = \{\bar{V}_1^i, \bar{V}_2^i, \bar{V}_3^i\}$.
Here, $\bar{V}_1^i = \{\bar{x}_1^i, \bar{y}_1^i, \bar{z}_1^i\}$, $\bar{V}_2^i = \{\bar{x}_2^i, \bar{y}_2^i, \bar{z}_2^i\}$ and $\bar{V}_3^i = \{\bar{x}_3^i, \bar{y}_3^i, \bar{z}_3^i\}$.
Steps:
1. Using chaotic keys $MK_1, MK_2, \ldots, MK_9$ generate nine chaotic sequences $S_1, S_2, \ldots, S_9$.
2. Sort the chaotic sequences in ascending order and store the new location of each value.
3. Using the stored locations, swap the locations of the mesh cloud data $M_1, M_2, \ldots, M_n$ to obtain intermediate point cloud data $IM_1, IM_2, \ldots, IM_n$.
4. Now, using the chaotic keys $MK_{10}, MK_{11}, \ldots, MK_{18}$ generate nine new chaotic sequences $S_{10}, S_{11}, \ldots, S_{18}$.
5. Sort the new chaotic sequences in descending order and store the new location of each value.

6. Using the stored locations swap the locations of the intermediate mesh cloud data $IM_1, IM_2, \ldots, IM_n$ to obtain encrypted mesh cloud data $EM_1, EM_2, \ldots, EM_n$.

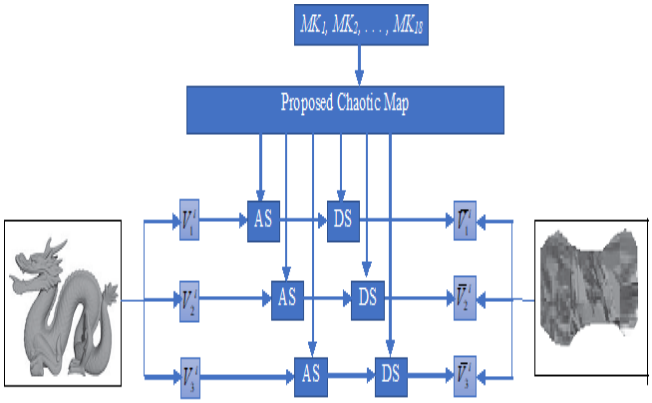

Fig. 9. Proposed two-level encryption scheme for mesh cloud data.

4.2.2. Decryption of 3D Mesh Cloud Model

The decryption of 3D mesh cloud is done to reverse the effect of encryption and to get back the original data $M_1, M_2, \ldots, M_n$ from the encrypted data $EM_1, EM_2, \ldots, EM_n$. This is given in Algorithm 4. Here, using the chaotic keys $MK_{10}, MK_{11}, \ldots, MK_{18}$, nine sequences namely $S_{10}, S_{11}, \ldots, S_{18}$ are generated. Then, these sequences are sorted in descending order and their location details are stored. Based on this information, the encrypted data is swapped to obtain the intermediate mesh cloud data $IM_1, IM_2, \ldots, IM_n$. Now, using the chaotic keys $MK_1, MK_2, \ldots, MK_9$ nine new chaotic sequences $S_1, S_2, \ldots, S_9$ are once again generated. The new chaotic sequences are again sorted in ascending order and the new location of each value is then stored. Using this information, the intermediate mesh cloud data $IM_1, IM_2, \ldots, IM_n$ is swapped to get the original data $M_1, M_2, \ldots, M_n$. This process is illustrated in Fig 10.

Algorithm 4: Decryption of 3D Mesh Cloud

Input:
Encrypted mesh cloud $EM_1, EM_2, \ldots, EM_n$.
Chaotic keys $MK_1, MK_2, \ldots, MK_{18}$.
Output:
Original mesh cloud $M_1, M_2, \ldots, M_n$.
Steps:
1. Using chaotic keys $MK_{10}, MK_{11}, \ldots, MK_{18}$ generate nine chaotic sequences $S_{10}, S_{11}, \ldots, S_{18}$.
2. Sort the chaotic sequences in descending order and store the new location of each value.
3. Using the stored locations, swap the locations of the encrypted mesh cloud data $EM_1, EM_2, \ldots, EM_n$ to obtain intermediate point cloud data $IM_1, IM_2, \ldots, IM_n$.
4. Now, using the chaotic keys $MK_1, MK_2, \ldots, MK_9$ generate nine new chaotic sequences $S_1, S_2, \ldots, S_9$.
5. Sort the new chaotic sequences in ascending order and store the new location of each value.
6. Using the stored locations swap the locations of the intermediate mesh cloud data $IM_1, IM_2, \ldots, IM_n$ to obtain the original mesh cloud data $M_1, M_2, \ldots, M_n$.

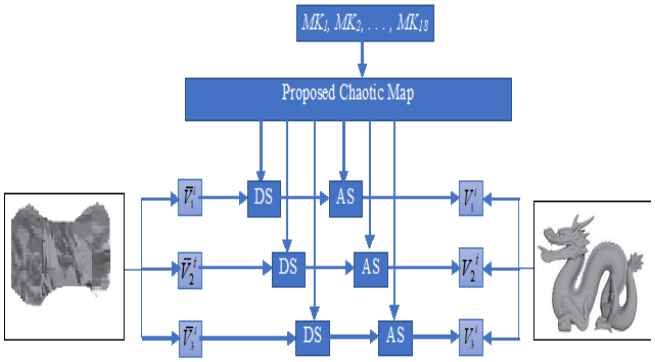

Fig. 10. Proposed two-level decryption scheme for mesh cloud data

5. RESULT AND DISCUSSIONS

The analysis of 3D point cloud and 3D mesh cloud data was performed using data in the Artec 3D and Stanford 3D scanning repository datasets respectively.

5.1 Security Analysis of 3D Point Cloud

The security analysis of 3D point cloud data is performed using secret key space analysis, secret key sensitivity analysis and speed of encryption analysis.

5.1.1 Secret Key Space Analysis

The key space of any encryption scheme should be very large so that the key cannot be predicted using brute force attack. Else, in a particular span of time, using an exhaustive search technique the key can be predicted and the data can be decrypted. The secret key for encryption of point cloud data is $PK_i = \{x_{0_i}, \mu_i\}$ where $i = 1,2,\ldots,6$. We know that in the proposed encryption scheme, $0.0000\ldots \leq x_{0_i} \leq 1.0000\ldots$ and $0.0000\ldots \leq \mu_i \leq 75.0000\ldots$ We also know that the precision value of a 64-bit double data is 10-15. Thus, the range is 1015. Therefore, the key space of the proposed scheme for point encryption is (1015)12 = (10)180 = (2)450. The key space of AES algorithm is (2)256. Thus, it is clear that the encryption capability of the proposed scheme is better than that of AES algorithm.

5.1.2 Secret Key Sensitivity Analysis

Since the proposed encryption scheme is based on chaos theory, it is highly sensitive to key parameters which are the initial condition and the control parameter. Thus, to test the secret key sensitivity we change the secret keys by a very small Δ=10-15 value. The decryption is then performed using new set of keys where $PK_i = \{x_{0_i} + \Delta, \mu_i + \Delta\}$ and $i = 1,2,\ldots,6$. Fig 11 shows the original data, encrypted data and the result obtained after decryption using the new set of keys. From the Fig 11, it is evident that the data cannot be retrieved back even if there is a small change in the key parameter values. Thus, our proposed framework possesses very high secret key sensitivity. The sensitivity level is in the order of 10-15 which a is very low value.

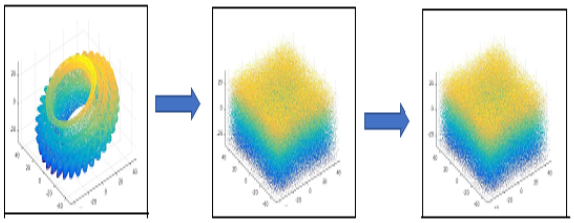

Fig. 11. Results obtained using secret key sensitivity analysis of point cloud data.

5.1.3 Speed of Encryption Analysis

The proposed system was simulated using MATLAB R2016b running on windows Intel i3 core processor with 6GB RAM. The time complexity of the proposed system is O(6N). The time taken by the proposed framework for encryption is compared with the previously proposed state-of-the-art algorithms and is shown in Table 2. From Table 2, we infer that the time taken by the proposed system is very less compared to the state-of-the-art works in the literature. It was compared with previously proposed techniques like random variable (RV), random transformation matrix (RTM) by (X. Zin et. al 2016) and random reversible matrix (Z.Wu et. al 2016).

Table 2. Encryption time analysis of point cloud data.

Point Cloud data	Size	Time (s)			
		RV	RTM	RRM	Proposed
Smart car	8097	0.000533	0.000698	0.000817	0.0000133
Gear	11061	0.000491	0.000428	0.000724	0.000121
Classic chair	33791	0.001331	0.003903	0.008754	0.000841
Copper key	2332	0.003413	0.004346	0.006324	0.000234
Dragon	51341	0.007342	0.008432	0.004235	0.000148

5.2 Security Analysis of 3D Mesh Cloud

The security analysis of 3D mesh cloud data is performed using secret key space analysis, secret key sensitivity analysis and entropy analysis.

5.2.1 Secret Key Space Analysis

The secret key for encryption of mesh cloud data is $MK_i = \{x_{0_i}, \mu_i\}$ where $i = 1,2,\ldots,18$. We know that in the proposed encryption scheme the values of the secret key parameters range between $0.0000\ldots \leq x_{0_i} \leq 1.0000\ldots$ and $0.0000\ldots \leq \mu_i \leq 75.0000\ldots$ Therefore, the key space of the proposed scheme for point encryption is (1015)36 = (10)540 = (2)1350. The key space of MD5 algorithm is (2)512. Thus, the encryption capability of the proposed scheme for mesh encryption is better than that of MD5 algorithm.

5.2.2 Secret Key Sensitivity Analysis

Since the proposed encryption scheme is based on chaos theory, it is highly sensitive to key parameters which are the initial condition and the control parameter. Thus, to test the secret key sensitivity for mesh data we change the secret keys again by a very small Δ=10-15 value. The decryption is then performed using new set of keys where $PK_i = \{x_{0_i} + \Delta, \mu_i + \Delta\}$ and $i = 1,2,\ldots,18$. Fig 12 shows the original data, encrypted data and the result obtained after decryption using the new set of keys.

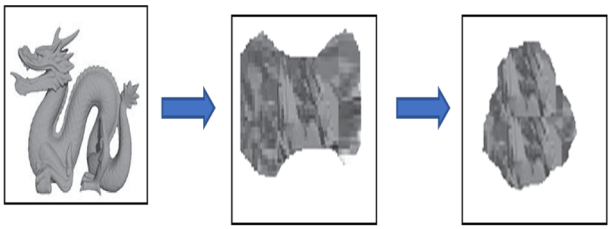

Fig. 12. Results obtained using secret key sensitivity analysis of mesh cloud data.

5.2.3 Entropy Analysis

The best way to quantize the security of mesh encryption is by means of entropy analysis. This is because entropy gives the uncertainty of an information source. It can be also defined as a measure of confusion. Thus, the amount of difficulty to retrieve the original mesh data without the use of the secret key is given by entropy. It is calculated in (Y. Liang et. al (2019). We compare the proposed scheme with state-of-the-art works like by (G.N. Pham's et. al 2017; M. Eluard et. al 2013; Y. Liang et al 2019).

Table 3. Entropy analysis of mesh cloud data.

Mesh Cloud data	Entropy			
	Pham's	Marc's	Liang's	Proposed
Bunny	39551243	2690986	3748975	89476927
Happy Budda	3452313	1343517	4351453	7431440
Dragon	56937564	38465937	48563847	94638496
Armadillo	462856	659367	873647	1275960
Thai statue	748364	659376	985647	1648397

6. CONCLUSIONS

In this research, we have presented a novel scheme for encryption of point and mesh cloud data. The proposed scheme utilizes the sequences generated by a novel chaotic map for encryption. The chaotic properties of the proposed chaotic map were proved using bifurcation analysis, Lyapunov exponent and approximate entropy. Through quantitative analysis, it was shown that the randomness of the proposed map was greater than that produced by commonly used logistic and sine logistic maps. In addition, the proposed double encryption scheme produced excellent results in terms of security analysis. For real-time implementation, the main aspect is encryption time. The proposed point cloud encryption scheme utilized minimum encryption time compared to state-of-the-art works in the literature. Further, the entropy of the encrypted mesh data was also computed and compared with that achieved by state-of-the-art works. It

was observed that our system produces the best results. To minimize the problem of data leakage there must be fewer keys. Since our framework is based on chaotic maps, very few keys were utilized and thus data leakage problem was also eradicated.

Conflict of interest: The authors declare that they have no conflict of interest

REFERENCES

C. Yuan, X. Yu, and Z. Luo (2017). 3D point cloud matching based on principal component analysis and iterative closest point algorithm,*ICALIP 2016 - 2016 International Conference on Audio, Language and Image Processing - Proceedings*, pp. 404–408.

K.Guo, D. Zou, and X. Chen(2015). 3D mesh labeling via deep convolutional neural networks,*ACM Transactions on Graphics*., vol. 35, no. 1.

S.Andalib and S. Azad(2014). The RSA algorithm,Practical Cryptography: Algorithms and Implementations Using C++.

M.J.Dworkin(2001). FIPS 197, Advanced Encryption Standard (AES),*Network Security, National Institute of Standard and Technology*., vol. 197, no. 12, pp. 6028.

J. Grabbe(1992). The DES algorithm illustrated, *Laissez Faire City Times*, pp. 1–15.

M. N. Valmik and P. V. K. Kshirsagar(2014). Blowfish Algorithm,*IOSR Journal of Computer Engineering.*, vol. 16, no. 2, pp. 80–83.

J. Solomon and I. V Be(2016). A Study of Twofish Algorithm,*International Journal of Engineering Development and Research*., vol. 4, no. 2, pp. 2321–9939.

A.D.Fernandas(1999).Elliptic-curve cryptography*Dr. Dobb's Journal.,* vol. 24, no. 12, pp. 56–63.

C. P. Schnorr and M. Jakobsson(2000). Security of signed ElGamal encryption, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 1976, pp. 73–89.

M. Steiner, G. Tsudik, and M. Waidner(1996). Diffie-Hellman key distribution extended to group communication,*Proceedings of the ACM Conference on Computer and Communications Security*, pp. 31–37.

D. L. Turcotte(2012). Logistic map,*Fractals and Chaos in Geology and Geophysics*, pp. 231–244.

C. K. Huang and H. H. Nien(2009), Multi chaotic systems based pixel shuffle for image encryption, *Optical Communications*., vol. 282, no. 11, pp. 2123–2127.

S. Lian, J. Sun, and Z. Wang(2005). A block cipher based on a suitable use of the chaotic standard map, *Chaos, Solitons and Fractals*, vol. 26, no. 1, pp. 117–129.

A.M. Del Rey (2015).A method to encrypt 3D solid objects based on three-dimensional cellular automata,*Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 9121, pp. 427–438.

R.Arora and A. Parashar(2013). Secure User Data in Cloud Computing Using Encryption Algorithms,*International Journal of Engineering Research and. Applications.,* vol. 3, no. 4, pp. 1922–1926.

M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou(2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, *IEEE Transactions on Parallel Distributed. System.,* vol. 24, no. 1, pp. 131–143.

X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng(2019). Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage,*Inf. Sci. (Ny).*, vol. 494, pp. 193–207.

X.Zhang, C. Xu, H. Wang, Y. Zhang, and S.Wang(2019), FS-PEKS: Lattice-based Forward Secure Public-key Encryption with Keyword Search for Cloud-assisted Industrial Internet of Things,*IEEE Transactions on Dependable and Secure Computing.*, pp. 1–10.

Y. Zhang, C. Xu, J. Ni, H. Li, and X. S. Shen(2019). Blockchain-assisted Public-key Encryption with Keyword Search against Keyword Guessing Attacks for Cloud Storage,*IEEE Transactions on Cloud Computing.*, pp. 1–14.

V. Kakkad, M. Patel, and M. Shah(2019). Biometric authentication and image encryption for image security in cloud framework,*Multiscale and Multidisciplinary Modelling, Experiments and Design.,* vol. 2, no. 4, pp. 233–248.

J. Li et al.(2019). An Efficient Attribute-Based Encryption Scheme with Policy Update and File Update in Cloud Computing,*IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6500–6509.

T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie(2020). Edge-based differential privacy computing for sensor–cloud systems, *Journal of Parallel and Distributed Computing*., vol. 136, pp. 75–85.

A. Alabdulatif, I. Khalil, and X. Yi(2020). Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption, *Journal of Parallel and Distributed Computing.,* vol. 137, pp. 192–204.

Y. Li, H. Yu, B. Song, and J. Chen(2019). Image encryption based on a single-round dictionary and chaotic sequences in cloud computing, *Computer Science; Concurrency and Computation: Practice and Experience.,* vol. 5182, pp. 1–15.

T. Xiang, J. Hu, and J. Sun(2015). Outsourcing chaotic selective image encryption to the cloud with steganography,*Digital Signal Processing.,* vol. 43, pp. 28–37.

M. Mishra, P. Singh, and C. Garg(2014). A New Algorithm of Encryption and Decryption of Images,*International Journal of Computer Science and Engineering.,* vol. 4, no. 7, pp. 741–746.

N. K. Pareek, V. Patidar, and K. K. Sud(2006). Image encryption using chaotic logistic map, *Image and Vision Computing.,* vol. 24, no. 9, pp. 926–934.

S. Som, A. Mitra, S. Palit, and B. B. Chaudhuri(2019). A selective bitplane image encryption scheme using chaotic maps,*Multimedia Tools and Applications.,* vol. 78, no. 8, pp. 10373–10400.

K. A. K. Patro, B. Acharya, and V. Nath(2019). Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps, *Microsystem Technologies.,* vol. 25, no. 12, pp. 4593–4607.

X. Jin, Z. Wu, C. Song, C. Zhang, and X. Li(2016). 3D point cloud encryption through chaotic mapping, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),*vol. 9916 LNCS, pp. 119–129.

C. Jia, T. Yang, C. Wang, B. Fan, and F. He(2019).Encryption of 3D Point Cloud Using Chaotic Cat Mapping,*3D Research.*,vol. 10(1).

L. Huang, M. Hwang, and L. Tseng(2013). Reversible Data Hiding for Medical Images in Cloud Computing Environments Based on Chaotic Hénon Map, *Journal of Electronics and Technology*, vol. 11, no. 2, pp. 230–236.

Y. Liang, F. He, and H. Li(2019), An asymmetric and optimized encryption method to protect the confidentiality of 3D mesh model,*Advanced Engineering Informatics,* vol. 42.

X. Wang, M. Xu, and Y. Li(2019). Fast encryption scheme for 3D models based on chaos system, *Multimedia Tools and Applications.*, vol. 78, no. 23, pp. 33865–33884.

R. M. May(1976). Simple mathematical models with very complicated dynamics, *Nature*, vol. 261, no. 5560, pp. 459–467.

J. Feng, J. Zhang, and X. Zhu(2016). A novel chaos optimization algorithm, *Multimedia Tools and Applications.*

A. Wolf, J. B. Swift, H. L. Swinney, and J. A.Vastano(1985).Determining Lyapunov Exponents From A Time Series, *Physica D:Nonlinear Phenomena. 16D*, pp. 285–317.

S. Pincus and S. Pincus(1995). Approximate entropy (ApEn) as a complexity measure Approximate entropy (ApEn) as a complexity measure.*An Interdisciplinary Journal of Nonlinear Science.*, vol. 5, no. 1, pp. 110–117.

Z. Wu, X. Jin, C. Song, C. Zhang, and X. Li(2016). Random reversible matrix based point cloud encryption, *Xitong Fangzhen Xuebao / Journal of. System and Simulation.*, vol. 28, no. 10, pp. 2455–2459.

G. N. Pham, K. R. Kwon, E. J. Lee, and S. H. Lee(2017). Selective encryption algorithm for 3D printing model based on clustering and DCT domain, *Journal of Compute Science and Engineering.*, vol. 11, no. 4, pp. 152–159.

M. Éluard, Y. Maetz, G. Doërr, R. Technicolor, and D. France(2013). Geometry-preserving Encryption for 3D Meshes,*Proceedings of COmpression et REprésentation des Signaux Audiovisuels*, Nov, pp. 7–12.