Applied Intelligence (2023) 53:4733–4747
https://doi.org/10.1007/s10489-022-03730-x

Blockchain based Chaotic Deep GAN Encryption scheme for securing medical images in a cloud environment

K. L. Neela[1] · V. Kavitha[2]

Accepted: 6 May 2022 / Published online: 15 June 2022

Abstract

Enormous reports, like scanned medical pictures and electronic patient records, are generated by modern healthcare equipment, and these reports must be secured for future access. Conventional storage technologies are incapable of storing large amounts of data. Cloud storage systems, fortunately, satisfy because of their features, like availability and scalability. Although medical images can be saved in the cloud, majority of cloud service providers store them in plain text format. Because of hackers and the increasing computer power, current image encryption methods are vulnerable to attacks. Therefore, a Block chain based Chaotic Deep Generative Adversarial Network (GAN) Encryption Scheme is proposed. The BCDGE uses blockchain technology to protect personal information and verify the authenticity of the data. Secondly, the Chaotic Deep GAN Encryption method uses confusion, substitution, and diffusion principles to encrypt the medical image. The Deep GAN creates image-specific secret keys to improve the resilience against hackers, produced keys utilized as an input for confusion and diffusion phases. The sender transfers the encrypted medical picture to the cloud server, signs the ciphertext ID, and saves it to block chain. The security and performance analysis of the proposed scheme gives better results than the existing schemes.

Keywords Cloud computing · Medical image · Data security · Encryption · Block chain · Cryptography

1 Introduction

Today, cloud computing has achieved a great growth in industrial technologies like Amazon, HP, Apple, IBM, and Oracle [1–3]. The user in the cloud can use both hardware and software resources via internet. Resources such as services, storage, server's data, and software can be shared with multiple users by an Internet-based computing model named cloud computing. The three common cloud models are private, public, and hybrid. The private cloud is considered as the best option for businesses that have set-up as it offers advanced levels of security than common cloud. The public cloud, the management infrastructure and the cloud provider offer their possessions to the public user using virtualization. This permits numerous users to share the similar set-up and pay for their use. The hybrid cloud combines the characteristics of both private and public cloud models [4–9]. Since, cloud computing has a distributed nature, intruders may attack the cloud systems to exploit vulnerabilities. Intrusion occurs when an unauthorized user tries to access the information from the cloud, while intrusion detection is a monitoring and controlling measures that is performed on network. Network intrusion detection and prevention in the cloud is a foremost safety benefits among researchers [10, 11].

However, information stored in the cloud has various security threats due to the expose in Internet to public communications or third-party risk [12]. Also, the security issue rises on cloud computing due to identity and risk management, integrity control, auditing, logging, infrastructure, dependent risk and data access control. In cloud computing, an organization can provide their sensitive data to the cloud service provider. The possibility although, the sensitive data expands the cloud administrations that can be easily open and accessible for all, while getting possible failures [13–15]. By verifying the identity of the sender, personal e-mail and contact lists can be secured from being read by the intruders who try to get the data [16]. Several methods were employed to solve the cloud communication on privacy preserving and security solutions but there exist some limitations. Furthermore, the sensitive data storing on cloud computing is a main challenge and

✉ K. L. Neela
phd.klneela003@gmail.com

[1] Department of Computer Science and Engineering, University College of Engineering, Thirukkuvalai, Tamil Nadu, India

[2] Department of Computer Science and Engineering, University College of Engineering, Kancheepuram, Tamil Nadu, India

requires to incorporate the security problem rises on cloud computing. Currently, typical algorithms have been enormously high to provide security in healthcare industry, but in the meantime, they involve high costs and increased use of computing resources [17].

Telemedicine is a rapidly expanding area that involves giving medical care to patients who are not physically present in the same place as the physician. Artificial intelligence guide surgical procedures in real-time [18]. Moreover, confidential medical records of patients, like medical scans, are exchanged via the Internet. Since medical data is commonly available in a picture format, stringent security is essential to protect it. A sophisticated medical system necessitates technology capable of storing medical records in such a way that they can be accessed by authenticated users in any geographical location [19, 20].

Since cloud storage systems provide a solution to this, medical data sharing and data storage strategies in cloud databases have recently provided numerous benefits to patients and doctors in the medical field. Despite the benefits, like medical data sharing and storage in the cloud, data security remains a significant concern for the healthcare industry. When data is kept locally, the data owner is responsible for providing and maintaining the data. However, if the data owner wants to store it in a database system, the data is transferred to a third-party, like a Cloud Service Provider (CSP). CSP is susceptible to cyber-attacks, especially if it does not meet the security requirements. Therefore, storing the medical image while maintaining integrity is important. Data integrity refers to the confidence that the medical image has not been tampered with by unauthorized parties. An unauthorized user has the ability to alter medical images. As a consequence, the data change or loss in the image will cause misdiagnosis of disease. Therefore, a robust and reliable method is needed for securely transmitting sensitive medical data across public networks.

1.1 Contributions

The following are the contributions made in this work:

- The BCDGE uses blockchain technology to protect personal medical information and authenticates the medical data. As a result, security and authenticity increased.
- The chaotic Deep GAN based encryption scheme produces private key with large key space, randomness, complexity, and high entropy. This prevents the attackers from getting the actual medical data.
- The performance is determined by comparing the results with the existing encryption techniques.

The rest of the paper is arranged as follows. Section 2 specifies related work. Section 3 presents the proposed blockchain based chaotic Deep GAN encryption method. In section 4, performance analysis is conducted. Section 5 discusses the conclusion and future work.

2 Related works

Mondal and Goswami [21] proposed an effective honeypot algorithm for data security in the cloud computing. Initially, the dataset is preprocessed by a normalization method that replaces and removes unwanted missing values. After that, the GLCM algorithm for feature selection and CNN classifier predict and classify the attack types. In this privacy scheme, a Honeypot cryptography algorithm used for encryption. The cloud server is accountable for key generation and verification of the key with the user for authentication. Then the information from the cloud server is requested by the data owner. After providing the key, the files are decrypted using the Honeypot algorithm retrieved by the user. This provides effective security, but the cost is high.

Liu P [22] proposed public key encryption (PKE) scheme to protect data security in cloud computing. In this method, the RRA secure PKE against arbitrary function that give two structures, namely Related Randomness Attack-plaintext attack (RRA-PA) encryption scheme and Indistinguish-cipher attack (IND-CA). Here, they combined hardcode IND-CA with PKE for information security. This method provides security against the arbitrary function but, the performance is not effective.

Pavani V et al. [23] proposed enhanced cryptography mechanisms for secure data storage in cloud computing based on user game theory. Initially, the user store data on the cloud computing server then, the encrypted text and key are separated for avoiding multiple shareholder servers. The shareholder servers store numerous user keys and finally, decrypt the data using a cryptography algorithm. However, this process protects the Byzantine errors but it is complicated to use multiple domains in the same attribute.

Ke et al. [24] proposed a high-capacity reversible data hiding method based on the most significant bit of medical image encryption. Initially, they concealed the MSB value in order to predict it correctly during the decoding phase. Next, maximum capacity and complete reversibility are proposed for better image encryption. Here, repair and the data encryption encapsulation methods accomplish reversible data encryption recovery. Then, all the pixels in the encrypted image are named as 1bit messages in the encapsulation phase. Finally, secret information extract and recreate the original image. However, this method needs improvement in the effectiveness of the repair process and in its error prediction.

Ali et al. [25] presented a medical picture encryption technique based on a tent logistic tent structure and a Henon chaotic map. The user generates a shared secret encryption key using the public key of the medical center in this encryption scheme. The suggested chaos-based medical picture encryption system then employs a secret key for encryption. The user then signs the cypher picture and transmits it to the administrators, along with the digital signature and validation parameters. Finally, the health-care authority generates the shared secret key using a cypher image, a digital signature, and his own private key. The authority then performs decryption on the image using that secret key, verifies it using an authentication parameter, and accepts it if it is authenticated.

Afzal et al. [26] proposed biplane and chaotic image encryption to secure medical images in the cloud. Initially, the medical image is encrypted by a Biplane and Chaotic Encryption (BCE). Then, combining the two keys acquired using the logistic map and the linear feedback shift register yields the chaotic key sequence key used for encryption. Finally, the biplane is used to recover the original medical image after successfully obtaining the encrypted image from the cloud. The computational cost of this method is high.

Masood et al. [27] proposed Lightweight Cryptosystem (LC) dependent on Brownian motion, Henon chaotic map, and Chen's chaotic system to improve the security of encrypting medical images. Initially, the plain text image with rows and columns originally represented the complete dimension of a grayscale medical image. Then, each medical image was resized. After that, the resized image, which contains pixels, separated into smaller segments. Next, a 2D Henon chaotic map is utilized by the pixels to shuffle into each produced block. This process is called an intra-block shuffling process. After that, the image blocks are shuffled. Next, 3D-Brownian motion, which defines particles' position and impulses, changes each other. The proposed method occurs rate higher security in terms of unified average changing intensity, homogeneity, and peak to signal noise ratio, contrast, mean square error, energy, and number of pixels changing at a compared to existing method. It uses fewer computational resources while processing data quickly.

Lakshmi et al. [28] proposed the Hopfield Neural Network Images Encryption Scheme (HNN-IES) for MI cloud storage. This structure comprises of five stages. The underlying stage portrays a versatile key generation utilizing the back propagation neural network. Next, the stage presents an image of an explicit irregular arrangement generation utilizing HNN, from that point forward. Then, the confusion and dispersion measures are individually arranged in stages 3 and 4. At long last, shows availability foundation between the cloud and cryptosystem.

3 Proposed Blockchain based Chaotic Deep GAN Encryption (BCDGE) scheme

The key generation center, CSP, data user, blockchain, and data sender are the five major entities in the BCDGE model. Figure 1 depicts the BCDGE's system model. The data sender initially applies GLCM to extract associated data from the image. The sender then encrypts the medical image and signs it with his or her signature. After completing the encryption operation, the data sender transfers the encrypted medical image to the cloud server and saves the signature on the blockchain. Whenever a data user gets medical data, the signature recorded on the blockchain may be used to check the ciphertext's authenticity.

- **Data user:** The data user (physicians) can request the medical data from the cloud server to get the encrypted message. Moreover, the authenticity of the ciphertext is also verified by the users.
- **Data sender:** The data sender (which might include patients) encrypts and transfers the extracted medical data to the CSP. The ciphertext is also signed by the data sender and then stored on the blockchain network.
- **Cloud server:** CS has following aims: one is to store huge quantities of medical image data, and the other is to search and deliver the relevant ciphertext upon getting the DU's request.
- **Blockchain:** To establish a digital signature, we first built a hash of the picture using the SHA-1 algorithm. The blockchain checks the saved signature upon getting a request to confirm the validity of ciphertext. It returns 1 if it is true; else, it returns 0.
- **KGC:** The KGC generates the secret key and distributes to the data sender and user. The key generation process is explained in section 3.1.
- **Encrypt/decrypt:** Before sending the data to the cloud server, the medical image is initially encrypted by using the chaotic Deep GAN model as shown in Fig. 2.

3.1 Key generation phase

As illustrated in Fig.3, GAN creates a shared secret that is distinct for each image. GAN is trained by utilizing image features as inputs and secret keys as the output. The normalized significant features in the medical image extracted using GLCM [29], used as input for training the GAN. Mode collapse in GAN is overcome by employing the Multi Generator orthogonal GAN (MGO-GAN) method [30]. This method learns various pieces of information in an efficient and complementary manner by using multiple generators. Moreover, catastrophic forgetting is a significant issue in GAN, which is linked to mode collapse and non-convergence. This problem is overcome by the gradient penalty strategy [31]. By preserving and applying knowledge from prior tasks to the current task using the gradient penalty strategy, the non-convergence problem is eliminated.

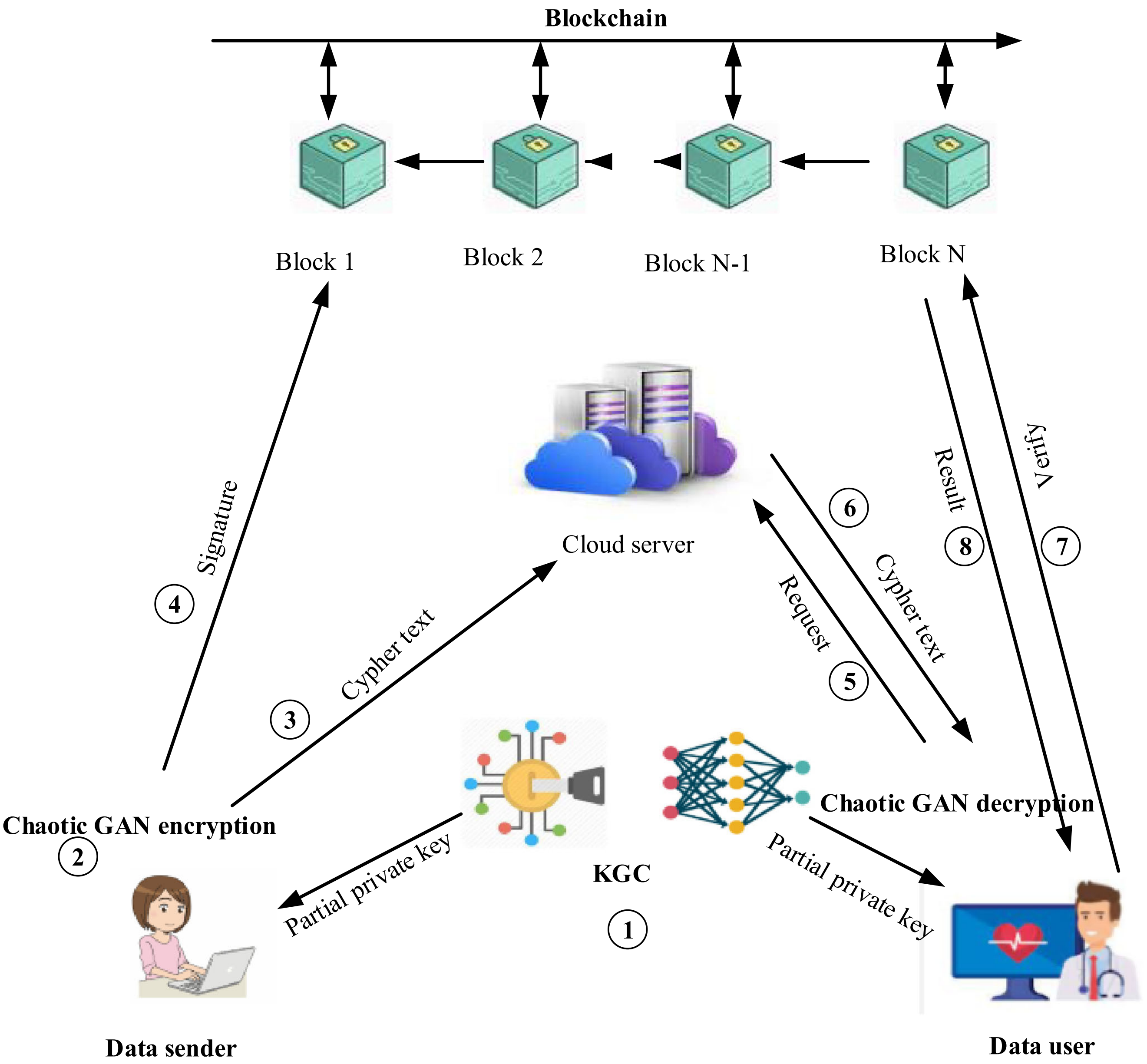


Fig. 1 Architecture of the proposed BCDGE scheme

(i) Generator Network (GN)

The original picture in the source domain will be transferred to the transformation domain via GN. The source domain contains the initial picture used to generate the secret key whereas, the transform domain has the target picture. The secret key is the outcome of the generator network GN. Transposed convolutional layers, residual blocks, one regular Convolutional Layer (CL), and down sample layers, make up the GN. The feature vectors sent through transposed convolutional layers in order to return it to the low-dimension. Subsequently, a normal convolution used to transform the low-level features into a picture. The resulting picture is used as the private key.

The normalization method is used for every CL in order to increase the quality of the produced picture, speeding up convergence and minimizing gradient explosion. The GN's loss function is given in Eq. (1).

$$GN_{LF} = \min_{GN}\left(E_{q\sim \mathrm{pinf}(q)}\log(1-DN(q))\right) \tag{1}$$

where E signifies expectation value, p inf(q) signifies the probability distribution of q belonging to original information, the discriminator network is represented by DN, the original picture is represented by q and GN signifies the generator network.

(ii) Discriminator Network (DN)

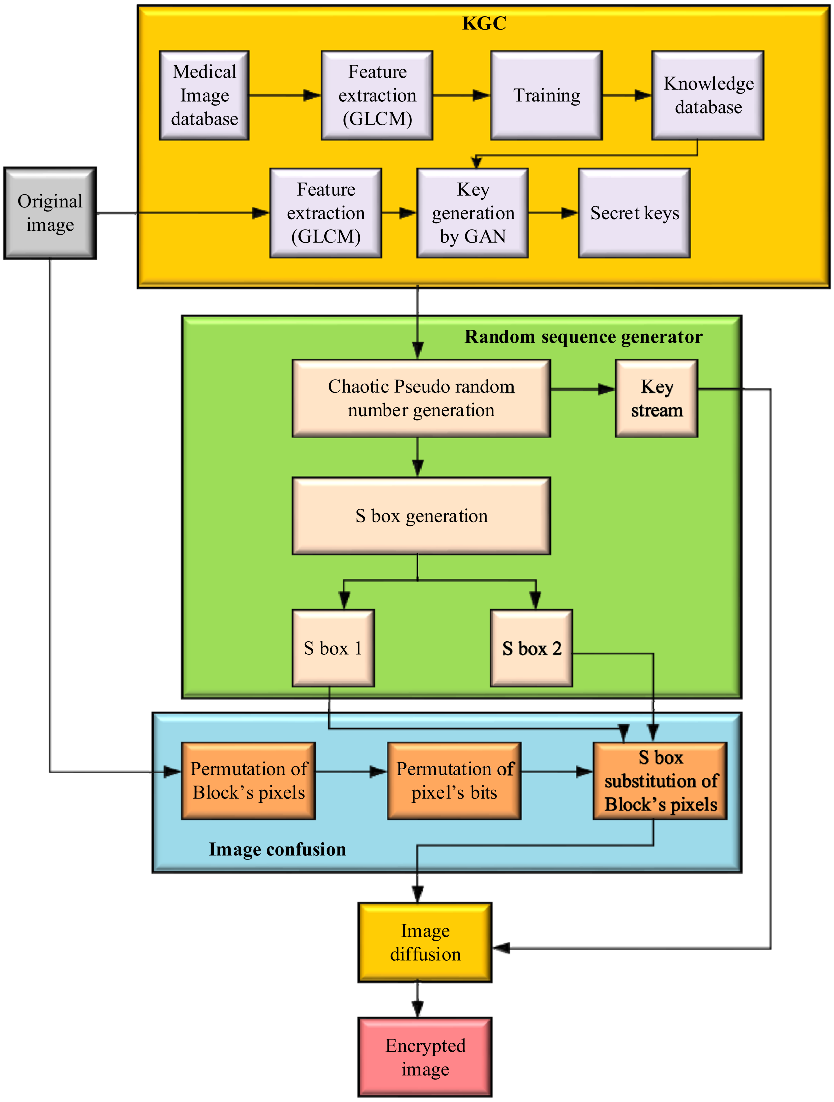

KGC
Medical Image database
Feature extraction (GLCM)
Training
Knowledge database
Original image
Feature extraction (GLCM)
Key generation by GAN
Secret keys
Random sequence generator
Chaotic Pseudo random number generation
Key stream
S box generation
S box 1
S box 2
Permutation of Block's pixels
Permutation of pixel's bits
S box substitution of Block's pixels
Image confusion
Image diffusion
Encrypted image

Fig. 2 Chaotic Deep GAN encryption model

To check if the produced picture (secret key) corresponds to the transform domain (target picture), the DN is employed. Five convolutional layers make up this network. Four convolutional layers process the input picture. They are then

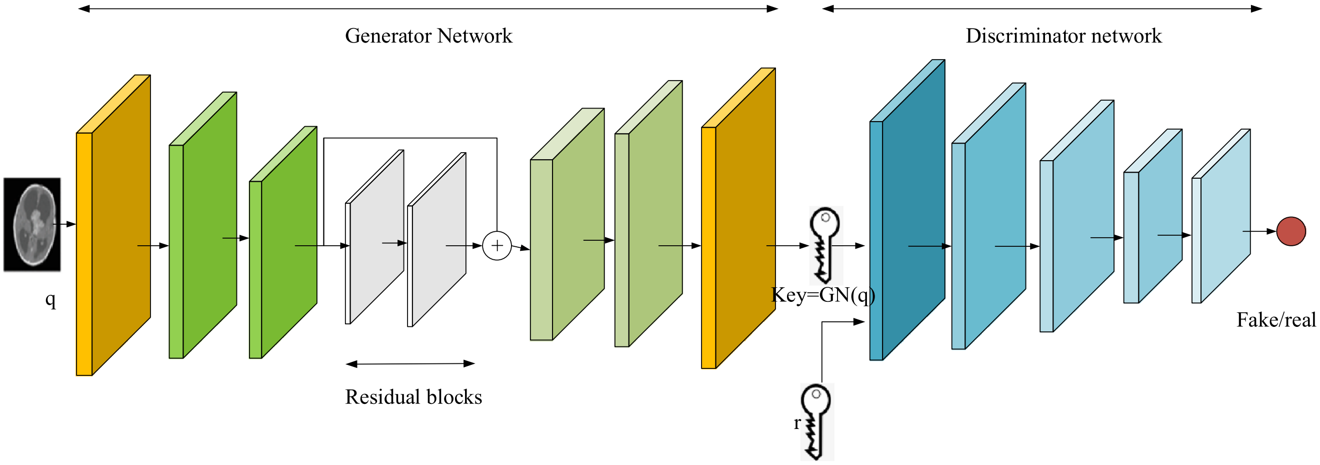


Fig. 3 Architecture of Chaotic Deep GAN

handled by the final layer, which produces a one-dimensional outcome. The discriminator's loss function is given in Eq. (2).

$$DN_{LF} = \max_{DN}\left(E_{q\sim \mathrm{pinf}(q)}\log(DN(q)) + E_{q\sim \mathrm{pinf}(q)}\log(1-DN(GN(r)))\right) \tag{2}$$

where the discriminator network is represented by DN, the original picture is represented by q, and GN signifies the generator network, and r signifies the data obtained from the transform domain. The loss function DN_{LF} improves DN's classification accuracy. Since the produced key is identical to that in the transformed domain, the DG is unable to easily distinguish them apart.

(iii) Image Private Key

Deep GAN generates a private key, which is the stream cypher and a picture. Each picture is composed of a series of pixels. The private key may be characterized as a combination of pixels, as shown in Eq. (3).

$$P_{key} = [N_1, N_2, \ldots N_i, \ldots N_n] \tag{3}$$

where N_i denotes image pixel. Also every N_i is made up of four quadruples as given in Eq.(4).

$$N_i = [h_x, v_y, c_h, p_i] \tag{4}$$

where v_y indicates value of vertical coordinate, h_x indicates value of horizontal coordinate. c_h denotes the color channel data, andp_i signifies the pixel value. p_i,v_y, and h_x have value ranges of 0–255, whereas c_h has a range of 0 to 2. The key (that comprises knowledge of pixel value as well as 3-dimensional space location data) guarantees that the secret key is complicated and the key space is vast enough, therefore greatly increasing the key's level of security.

(iv) Key Generation Phase

This is basically a network training procedure. Here, the parameters of the convolutional layers are initialized randomly before training as given in Eq. (5).

$$C_n = ran[c_{d,1}, c_{d,2}, \ldots, c_{d,l}] \tag{5}$$

where l^{th} parameter of the chaotic Deep GAN's d^{th} CL is represented by $c_{d,\ l}$. All the C parameters expressed in Eq. (6).

$$C = ran[C_1, C_2, \ldots, C_n] \tag{6}$$

Chaotic Deep GAN is made up of a GN and a DN, with the generator generating the secret key and is defined in Eq. (7).

$$P_{key(GN)} = GN(C; q) \tag{7}$$

where GN() signifies the generator's Convolutional Neural network (CNN), q denotes the original image, and C denotes the entire network parameters. The private key is created during the forward propagation phase, and the developed key is then utilized to compute the overall lossT_{loss}. This helps to determine the difference between the target and the produced private key in the Transformation Domain (TD). The gradient descent technique estimates the weights in each layer to get enhanced performance. This is defined in Eq. (8).

$$C^k_{d,l} = C^{k-1}_{d,l} - \gamma \nabla K\left(C^k_{d,l}\right) \tag{8}$$

where $C^k_{d,l}$ signifies the values of $c_{d,\ l}$ in the k^{th} round during training, $\nabla K\left(C^k_{d,l}\right)$ signifies the gradient loss that is sent to the d^{th} CL during the k^{th} round, and γ signifies the learning rate. The gradient descent approach may be used to adjust the network's parameters in order to learn the mapping relations effectively. The DN and the GN are trained in different ways. When the training epoch reaches the specified number or the loss stabilizes, a key is created similar to the target key in the transformation domain.

3.2 Encryption phase

The input medical image with size $P = J \times K \times L$ is encrypted where the number of layers is L and J and K signifies the dimension of the image.

3.2.1 Chaotic Pseudo random number generation (PRNG)

To initialize the model, the PRNG needs a 256-bit extrinsic secret key. The Deep GAN from the previous phase helps to generate this secret key. Therefore, the PRNG's initial state is generated using the outcome from GAN. The PRNG produce two S-boxes. The same concept used to create PMs will create S-boxes. Two distinct S-boxes are created using pseudorandom numbers.

3.2.2 Image confusion

(i) Pixel bits permutation

Based on the position parity of the pixel, the pixels in the medical image are permuted by cycling right or left shift. The procedure is depicted in Fig. 4. If the pixel is in a pair position, the bits of the pixel are permuted by cycling right 2 bits.

If the pixel's location is impaired, the bits of the image pixels are permuted by cycling two-bit left right:

$$P_{pair} = pixel << 2 \tag{9}$$

(ii) Permutation of pixel position

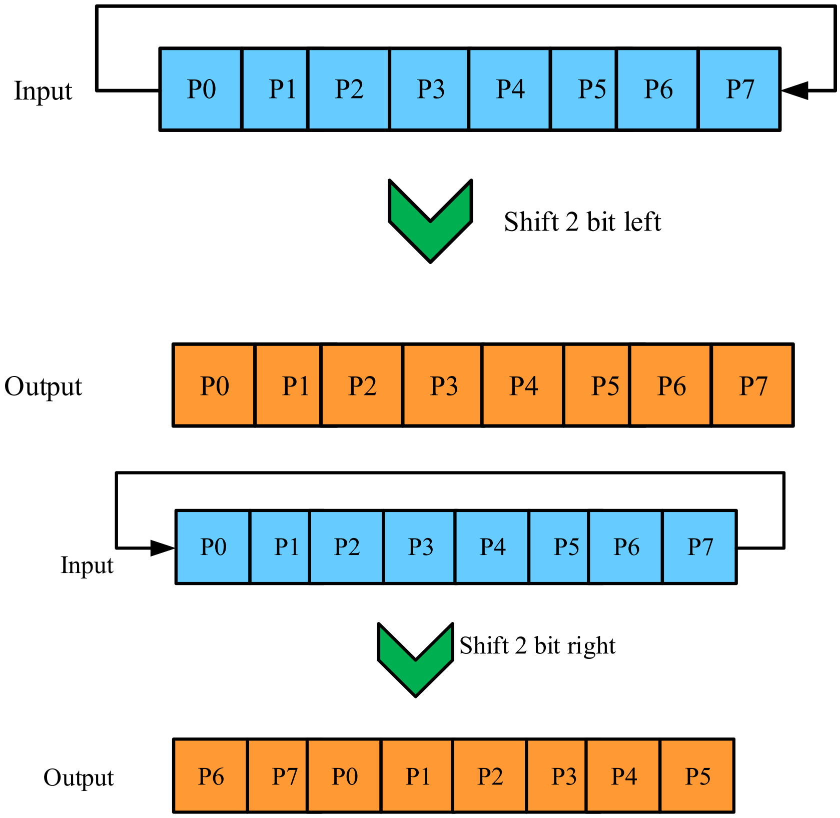


Fig. 4 Permutation of pixel bits

The PRNG is used to produce a permutation matrix PR_{mat} having size $J \times K$. The matrix is made up of randomized indices which is applied in order to permute the position of the medical image's pixels. Figure 5 shows the procedure for creating the matrix PR_{mat}. The PRNG is first iterated to generate a series of $J \times K$ random numbers. The numbers are then arranged in descending order while the index of each random number is preserved. To get the MP matrix, rearrange the series of indices into $J \times K$cases. Eventually, the pixels in the picture are permuted using the PM indices. Figure 6 shows how this technique works for 4 × 4 pixels. If the pixel's location is impaired, the bits of the image pixels are permuted by cycling two-bit left shift:

$$P_{imp} = pixel >> 2 \tag{10}$$

(iii) S-box substitution

Arranging the pixels in descending order after PSNR gives S-box-1 whereas arranging the pixels in ascending order after PSNR gives S-box-2. Following that, a condition is utilized for the permutation of pixels, allowing the S-box1 or S-box2 to be employed depending on the permuted pixel acquired in the previous step:

If the permuted pixel's location is paired, the medical image pixel is replaced by the value of S-box1:

$$PL_{rep} = Sbox1\left(P_{pair}\right) \tag{11}$$

If the permuted pixel's location is impaired, the medical image pixel is replaced by the value of S-box2:

$$PL_{rep} = Sbox2\left(P_{imp}\right) \tag{12}$$

A 256-case substitution list is referred to as the S-box. Consider the medical image P as an 8-bit coded picture for each pixel, with U and V being binary values derived from each pixel, as shown in Eq. (13).

$$P_{ab} = P0P1P2P3P4P5P6P7 \tag{13}$$

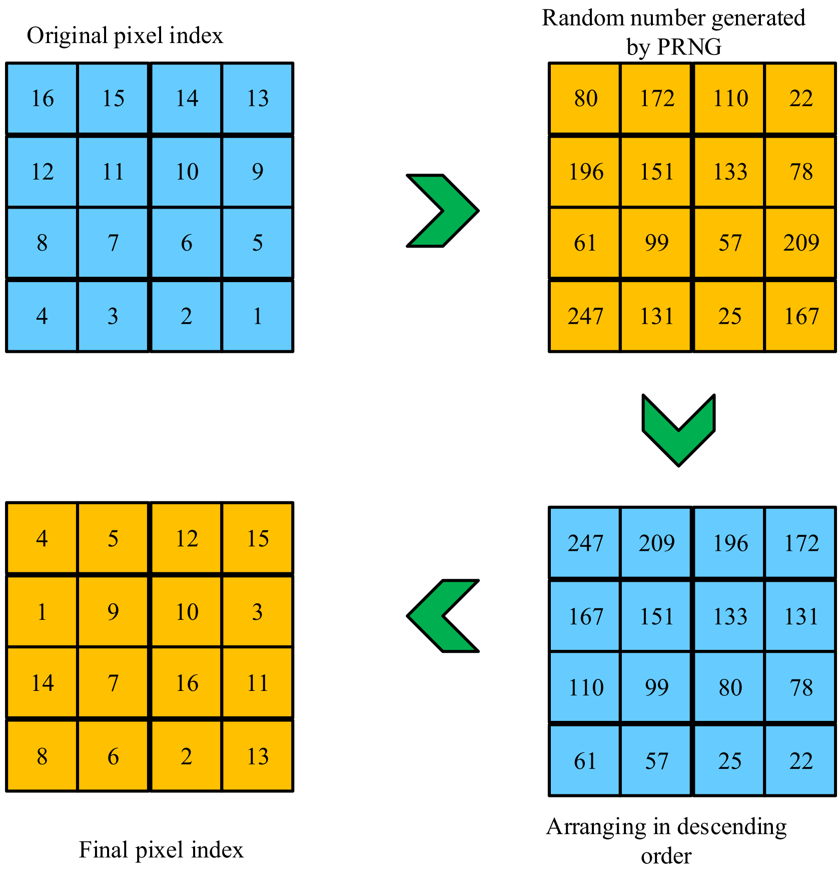
where $U = P4P5P6P7$ and $V = P0P1P2P3$.

The table value that meets the junction of U and V in the S-box is replaced for each pixel of the block.

3.2.3 Image diffusion

In order to produce a key stream, the PRNG is iterated $J \times K$ times. The picture obtained in the previous phase is then diffused simply by XORing the pixels of the image with the key stream. An encrypted image is obtained as a result of this procedure. The encrypted images then stored in the cloud by the data sender.

Fig. 5 Creation of permutation matrix for 4 × 4 pixels



Original pixel index

16	15	14	13
12	11	10	9
8	7	6	5
4	3	2	1

Random number generated by PRNG

80	172	110	22
196	151	133	78
61	99	57	209
247	131	25	167

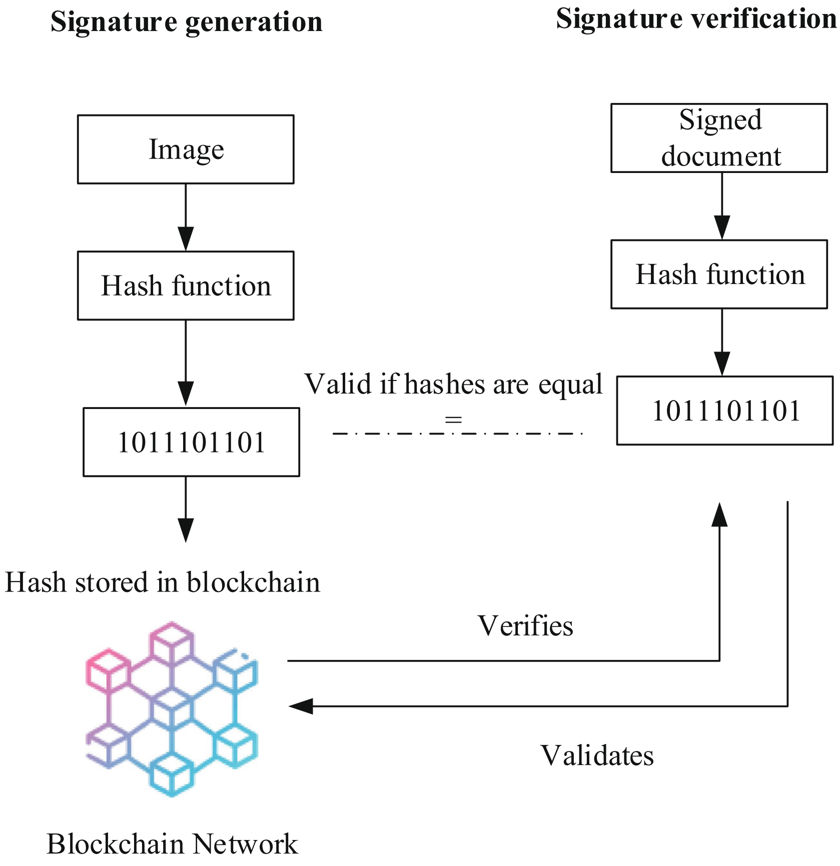
Arranging in descending order

247	209	196	172
167	151	133	131
110	99	80	78
61	57	25	22

Final pixel index

4	5	12	15
1	9	10	3
14	7	16	11
8	6	2	13

Fig. 6 General architecture of digital signature generation and verification process



3.3 Signature generation

Once the encryption process is completed, the data sender transfers the encrypted medical image to the cloud server and saves the hash on the blockchain by using a hashing function. This may be to check the ciphertext's authenticity.

3.4 Decryption and signature verification

To retrieve the original image, the reverse process of the encryption algorithm is applied by the data user. Finally, the signature recorded on the blockchain is verified by the data user to check the ciphertext's authenticity. In order to verify the authenticity, the blockchain checks if the hash function generated by the data user matches the hash stored on the blockchain by the data sender. If it matches, then the data is valid else invalid. Figure 6 shows the signature verification process.

4 Experimental results and comparative analysis

The experiments carried out on a UBUNTU 16.04 desktop with Intel(R) Core(TM) i7–6700 @ 3.40 GHz specs. Python 3.10 is used to run the simulation. A private blockchain created using the Geth Ethereum client to replicate the proposed system. Ethereum is one of the most widely used blockchain platforms, and its performance has been investigated by researchers and developers. The software used for implementation are provided in Table 1.

The BraTS18 dataset [32], the Montgomery County chest X-ray dataset [33], and the Ultrasonic Brachial Plexus dataset (https://www.kaggle.com/c/ultrasound-nerve-segmentation/data/?select=sample_submission.csv) used in the assessment since they reflect three distinct anatomical areas. The input image's resolution is 256 × 256 throughout training and the network's weight parameters are randomly initialized. To adjust the loss function, Adam optimizer is used. The learning rate is set at 0.0002 and the training epoch has been set to 20,000 to improve performance.

Experiments with various training epochs, batch sizes, and learning rates are examined to determine alternative hyper-parameters for network training. Furthermore, this dataset is divided into two halves, with 90% of the dataset serving as the training dataset and the remaining 10% serving as the validation set. Here, the mean Information Entropy (IE) for the produced secret key analyze the various hyper-parameters in the network. It has been observed that when the learning rate decreases, the performance of the network continues to improve. The highest IE of the produced secret key is more than 7.9 when the learning rate is reduced to 0.0002. Also, with 0.0002 learning rate and a batch size 1, the performance of the network is always superior to the outcomes acquired with a batch size of 6 or 10. Therefore, the best results are obtained with training epoch 20,000, 0.0002 learning rate, and batch size of 1. The hyper parameters of GAN are provided in Table 2.

4.1 Security analysis

(i) Key Space (KS) analysis

The resilience to exhaustive assaults is determined by the KS size. The produced secret key using Deep GAN is obtained in a picture format. The dimension of the picture is 256 × 256, where each pixel's value ranges from 0 to 255. Thus, the KS of the produced key becomes $(2^8)^{65536}$. As a result, attackers will have a much harder time guessing the secret key, and also the KS will be big enough to withstand exhaustive assaults.

(ii) IE

This measure has been utilized to measure the randomness or unpredictability of a private key and to show the uncertainty level. It is determined using Eq. (14).

Table 1 Parameter setting

Software	Use	Version/value
Windows	Operating system	8.1
Mist	Ethereum wallet	0.9.2
Geth	Command line Ethereum client	1.7.2
Claymore pool	Claymore's Dual Ethereum AMD GPU Miner	10
Cloud	Average RAM	512 MB
	Number of virtual machines	34
	Number of users	100
	Average bandwidth	1000,000 MB

Table 2 Hyper-parameters of GAN

Parameters	Value
Learning Rate	0.0002
Epoch	20,000
Batch Size	1
Optimizer	Adam
Dropout Rate	0.5

$$IE = \sum_{i=0}^{t-1} P(r_i)\log_2 \frac{1}{P(r_i)} \tag{14}$$

where, the probability of r_i pixel is represented by $P(r_i)$, and t signifies the number of pixels. For grayscale pictures, the maximum entropy value is 8. The entropy of the produced private key as shown in Table 3 is around 7.98, indicating that the obtained private key has a good level of unpredictability.

(iii) Histogram Analysis

From Table 4 it is observed that the pixel variations of the ciphertext pictures are very consistent, as opposed to the histogram obtained for the plaintext pictures. The variations are remarkably similar to the histogram dispersion of white noise, indicating that the encrypted picture effectively secures the plaintext images' statistical data. Furthermore, it has been confirmed that the produced private key aids to encrypt the images in a random manner. As a result, it becomes extremely difficult for hackers to extract meaningful information from the encrypted pictures.

(iv) Sensitivity Analysis

During the analysis, the original image's one-pixel value is modified at random. The two pictures are then given as inputs to the Deep GAN to produce two private keys, one before and one after altering one of the pixel values. Now, using these two measures, the discrepancies between two secret keys are computed to determine the sensitivity. To measure the divergences between the secret keys, two metrics are used: The Unified Average Changing Intensity (UACI) and the Number of Pixel Change Rate (NPCR).

The pixel rate of change, abbreviated as NPCR, is a measure of the ratio of values of pixels at the very similar position in two different images. It is determined using Eq. (15).

$$NPCR = \frac{\sum_{i=0}^{J}\sum_{i=0}^{K} F(i,j)}{I_s} \times 100\% \tag{15}$$

where

$$F(i,j) = \begin{cases} 1, & if\, D_1(i,j) \neq D_2(i,j) \\ 0, & if\, D_1(i,j) = D_2(i,j) \end{cases} \tag{16}$$

UACI measures the altered average intensity of two pictures. It is determined using Eq. (17).

$$UACI = \frac{\sum_{i=0}^{J}\sum_{j=0}^{K} \left| D_1(i,j) - D_2\left(i,j\right) \right|}{I_s} {*}100\% \tag{17}$$

Where,$D_1$and D_2signifies the pixel value in the location (i, j) and I_s signifies the size of the image.

The experiment is conducted with 8 different input images and its average NPCR and UACI values are found out. The findings from Table 5 show that a little modification to the original picture (a single pixel value change) caused over 99.6% variations between two produced private keys, with average intensity variations above 33%. This demonstrates that the private key produced is sensitive to the input picture, and therefore satisfies both uncertainty and randomness.

4.2 Performance evaluation of image encryption and decryption

The Structural Similarity Index (SSIM) and Mean Square Error (MSE) are the evaluation measures to determine the effectiveness of the decryption system. Table 6 provides the encryption and decryption results.

(i) Similarity Analysis

When using a high-security encryption system, the resemblance between the pictures prior and after encryption should be very minimal. The MSE is determined using Eq. (18).

$$MSE = \frac{1}{I_d}\sum_{i=1}^{N}\sum_{j=1}^{M} \left(u(i,j) - v(i,j)\right)^2 \tag{18}$$

Table 3 Entropy results for various methods

Methods/ Key ID	1	2	3	4	5	6	7	8
BCE	7.902	7.897	7.956	7.787	7.802	7.804	7.866	7.876
HNN-IES	7.914	7.855	7.982	7.799	7.924	7.816	7.922	7.899
LC	7.921	7.875	7.798	7.802	7.965	7.881	7.973	7.901
Chaotic-Deep GAN	7.980	7.954	7.984	7.894	7.972	7.895	7.961	7.931

Table 4 Histogram analysis.

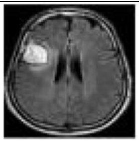
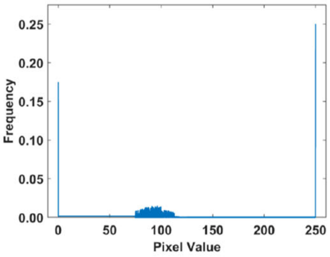
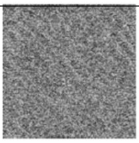
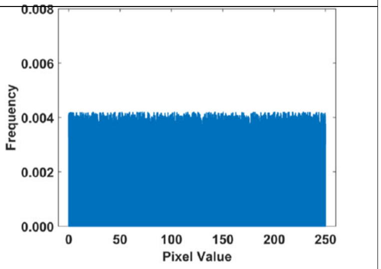
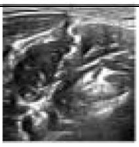
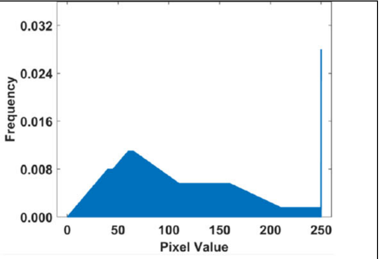
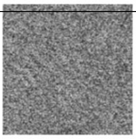
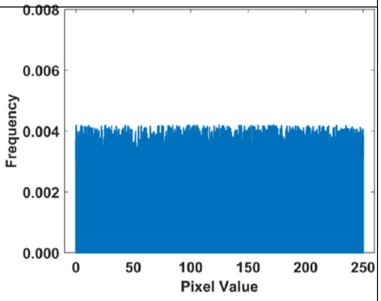
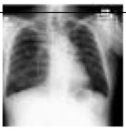
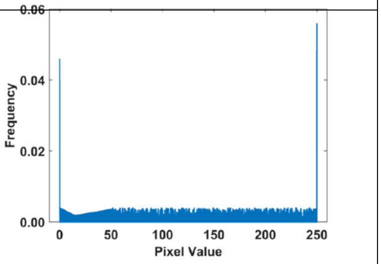
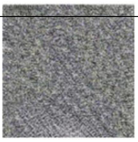
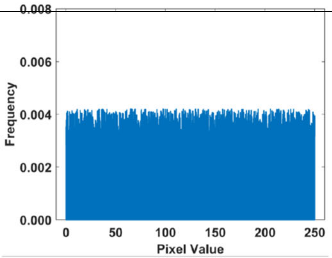
Sl.no	Plain/ciphertext image	Histogram
Plain text Image 1		
Ciphertext image 1		
Plain text Image 2		
Ciphertext image 2		
Plain text Image 3		
Ciphertext image 3		

Table 5 Average NPCR and UACI results for various methods

Method	Average NPCR	Average UACI
BCE	97.48	32.63
HNN-IES	97.41	32.79
LC	98.54	32.91
Proposed Chaotic-Deep GAN	99.60	33.86

where I_d signifies the dimension of the image, v denotes the ciphertext image, u denotes the plaintext image. The higher the MSE score, the lower the resemblance between the two pictures. SSIM is determined using Eq. (19).

$$SSIM(I_1, I_2) = \frac{(2\mu_{I_1}\mu_{I_2} + g_1)(2\sigma_{I_1I_2} + g_2)}{({\mu_{I_1}}^2{\mu_{I_2}}^2 + g_1)({\sigma_{I_1}}^2{\sigma_{I_2}}^2 + g_2)} \tag{19}$$

where I_1 and I_2 are two pictures, $2\sigma_{I_1I_2}$ represents the covariance of I_1 and I_2, ${\sigma_{I_2}}^2$ represents the variance of I_2, ${\sigma_{I_1}}^2$ represents the variance of I_1, μ_{I_2} represents the mean value of I_2, I_1 represents the mean value of I_1, and g_1 and g_2 are the constants needed to ensure stability. A higher SSIM value implies a high degree of resemblance between two pictures, with a value ranging between 0 to 1. The proposed BCDGE encryption technique is tested with 8 different input images and its average MSE and SSIM values are determined in Table 7. The results in Table 5 shows that the difference in MSE between ciphertext picture and plaintext pictures is quite high, but the SSIM values are near to zero. It shows that the ciphertext picture and plaintext pictures are not identical. Thus, the suggested technique provides good encryption performance.

(ii) Time complexity

In Fig. 7, the runtime of the encryption and decryption on medical pictures with various resolutions is analyzed to determine the effectiveness of the proposed network. The proposed network can encrypt or decrypt 14.31 medical images per second at 256 × 256resolution, while pictures with 512 × 512resolution can be encrypted or decrypted at 4 images per second. On both 512 × 512 and 256 × 256 resolution pictures, our technique has been proven to provide the fastest encryption speeds than other existing methods like BCE, HNN-IES, and LC.

(iii) Normalized Correlation (NC) Analysis

For attackers to exploit encryption technique, the significant correlation between neighboring pixels might offer a substantial quantity of statistical information. An effective encryption technique should be capable of reducing the correlation between neighboring pixels in a picture.

Table 6 Encryption and decryption results for proposed Chaotic Deep GAN scheme.

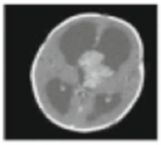
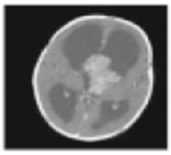
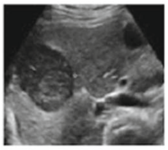
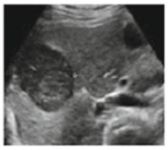
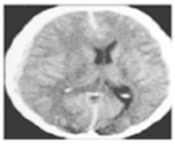
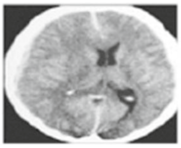
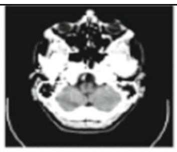
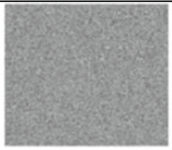
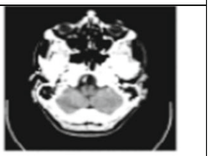
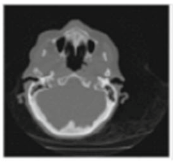
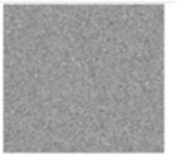
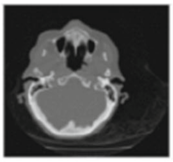
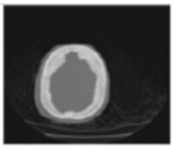
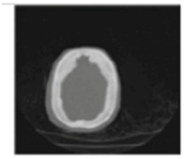
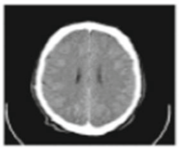
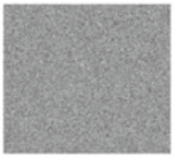
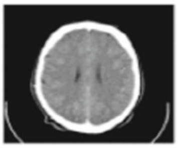
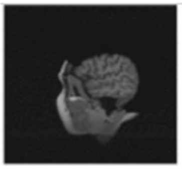
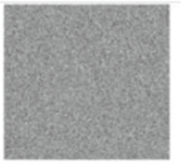
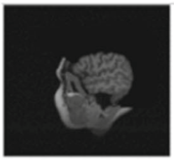
Sl.no	Original images	Encrypted image	Decrypted images
1			
2			
3			
4			
			
5			
6			
7			
8			

Table 7 Average MSE and SSIM results for various methods

Method	Average MSE	Average SSIM
BCE	92.89	0.089
HNN-IES	89.24	0.870
LC	95.72	0.285
Proposed Chaotic-Deep GAN	124.60	0.018

$$NC = \frac{\sum O(i,j)D(i,j)}{\sqrt{\sum O(i,j)^2 \sqrt{\left(\sum D(i,j)\right)^2}}} \tag{20}$$

Where $O(i, j)$ and $D(i, j)$ are the original and the decrypted images respectively. The larger is the NC value, the better is the image restoration quality. Table 8 provides the NC results obtained for the proposed and the existing techniques. The results show that the proposed technique has higher NC value (0.987) than the existing techniques such as BCE, HNN-IES, and LC.

(iv) Differential attack analysis

Differential attack is a type of attack in which an attacker attempts to crack an encrypted image without using secret keys. To do this, the attacker selects certain plain images adaptively, accesses the encryption machine to generate the matching cypher images, and then compares the obtained cypher images to retrieve the unknown data. A differential attack is introduced to determine the influence of a single pixel value change in the plain input image on the cypher image. As a result, the higher the NPCR value, the more challenging it is for adversaries to understand the link between cipher-text and plaintext. The results from Table 9 shows that the proposed encryption scheme has higher NPCR value than the existing techniques which shows that the proposed technique is good at resisting differential attacks.

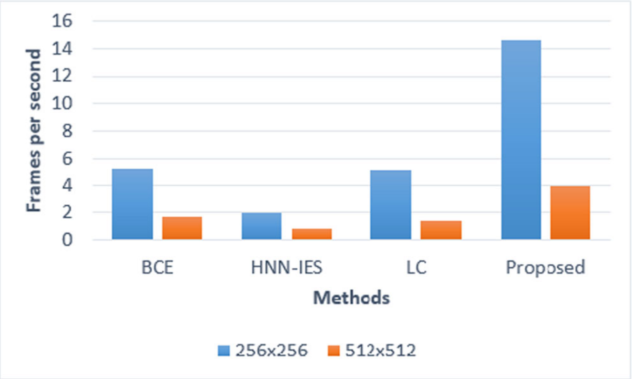


Fig. 7 Time complexity analysis

Table 8 NC Results

Method	NC
BCE	0.875
HNN-IES	0.849
LC	0.931
Proposed Chaotic-Deep GAN	0.987

(v) Confidentiality and Integrity analysis

The confidentiality of medical data saved in the cloud is a major security concern. Because of the significant costs of reputational damage, cloud providers have taken precautions to secure the confidentiality of their data. Data confidentiality in cloud cannot be easily maintained and secured due to the presence of attackers. Therefore, it is essential to analyze the data confidentiality and integrity of the proposed system. Before transmitting the medical image to the cloud server, the proposed BCDGE scheme encrypts it with his/her private key. The ciphertext cannot be decoded without the private key. Furthermore, only the institution/person that has received the consent of data owner is permitted to access the data stored on the cloud server. As a result, the ciphertext can only be decrypted by the authorized personnel, ensuring data confidentiality. Moreover, each block's signatures ensure data integrity. The blockchain network distinguishes between various nodes/users and their validity. By analyzing if the user has the authority to decrypt the ciphertext, the blockchain mechanism assures that only the authorized user decodes the ciphertext using the private key.

5 Conclusion

Cloud storage infrastructures are prone to a variety of security risks because of their openness. While using the cloud to create a medical picture repository, it is crucial to make sure the security measures are effective. Therefore, a blockchain-based secure architecture called BCDGE is proposed in this paper. Here, the medical images to be stored in the cloud are encrypted by using the Chaotic Deep GAN scheme. It

Table 9 NPCR Results

Method	NPCR
BCE	97.58
HNN-IES	99.33
LC	98.08
Proposed Chaotic-Deep GAN	99.49

produces the secret key, followed by confusion and a diffusion process. The XOR method decrypt the picture using the created secret key. The sender then stores the encrypted image on the cloud, signs the ciphertext ID, and saves it on the blockchain. The signature can later be used to verify the authenticity of the ciphertext image. Experimental findings and security analyses demonstrate that the proposed Chaotic Deep GAN method has pseudo-randomness, a wide key space, and is extremely sensitive to modification. Thus, the BCDCD architecture offers a high degree of security when compared to other existing techniques.

Authors contributions All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

Declarations

This article does not contain any studies with human participants and/or animals performed by any of the authors.

Conflict of interest Authors declares that they have no conflict of interest.

Consent to participate There is no informed consent for this study.

Consent for publication Not Applicable.

References

1. Altowaijri SM (2020) An architecture to improve the security of cloud computing in the healthcare sector. In Smart Infrastructure and Applications 249–266
2. Dutta A, Misra C, Barik RK, Mishra S (2021) Enhancing mist assisted cloud computing toward secure and scalable architecture for smart healthcare. InAdvances in communication and computational technology springer, Singapore. 1515-1526
3. Sri Vigna Hema V, Kesavan R (2019) ECC based secure sharing of healthcare data in the health cloud environment. Wirel Pers Commun 108(2):1021–1035
4. Bokhari MU, Makki Q, Tamandani YK (2018) A survey on cloud computing. In Big Data Analytics 149–164
5. Singh G, Garg S (2020) Fuzzy elliptic curve cryptography based cipher text policy attribute based encryption for cloud security. In 2020 international conference on intelligent engineering and management (ICIEM) 327-330
6. Mishra S, Sharma SK, Alowaidi MA (2020) Analysis of security issues of cloud-based web applications. Journal of Ambient Intelligence and Humanized Computing. 1-2
7. Benssalah M, Rhaskali Y, Drouiche K (2021) An efficient image encryption scheme for TMIS based on elliptic curve integrated encryption and linear cryptography. Multimedia Tools and Applications 2:2081–2107
8. Xia Z, Jiang L, Liu D, Lu L, Jeon B (2019). BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing. IEEE Trans Serv Comput
9. Chai X, Zhang J, Gan Z, Zhang Y (2019) Medical image encryption algorithm based on Latin square and memristive chaotic system. Multimed Tools Appl 78(24):35419–35453
10. Vurukonda N, TrinadhBasu M, Velde V, Enumula K (2020) Revocable storage identity based encryption for protected shared data in cloud computing. Materials Today: Proceedings
11. Gupta BB, Yamaguchi S, Agrawal DP (2018) Advances in security and privacy of multimedia big data in mobile and cloud computing. Multimed Tools Appl 77(7):9203–9208
12. Li J, Wang S, Li Y, Wang H, Wang H, Wang H, Chen J, You Z (2019) An efficient attribute-based encryption scheme with policy update and file update in cloud computing. IEEE Transactions on Industrial Informatics 15(12):6500–6509
13. El Makkaoui K, Ezzati A, Beni-Hssane A, Ouhmad S (2019) Fast cloud–Paillier homomorphic schemes for protecting confidentiality of sensitive data in cloud computing. Journal of Ambient Intelligence and Humanized Computing. 1-0
14. Zhang L, Cui Y, Mu Y (2019) Improving security and privacy attribute based data sharing in cloud computing. IEEE Syst J 14(1):387–397
15. Vijitha S, Unnithan SN (2021) Secure medical image transmission using modified leading diagonal sorting with probabilistic visual cryptography. Materials Today: Proceedings
16. Zhang Q, Yang LT, Castiglione A, Chen Z, Li P (2019) Secure weighted possibilistic c-means algorithm on cloud for clustering big data. Inf Sci 479:515–525
17. Sivaram M, Kaliappan M, Shobana SJ, Prakash MV, Porkodi V, Vijayalakshmi K, Vimal S, Suresh A (2020) Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud. Journal of Ambient Intelligence and Humanized Computing
18. Kamal ST, Hosny KM, Elgindy TM, Darwish MM, Fouda MM (2021) A new image encryption algorithm for Grey and color medical images. IEEE Access. 9:37855–37865
19. Kari AP, Navin AH, Bidgoli AM, Mirnia M (2021) A new image encryption scheme based on hybrid chaotic maps. Multimed Tools Appl 80(2):2753–2772

20. Vaseghi B, Mobayen S, Hashemi SS, Fekih A (2021) Fast reaching finite time synchronization approach for chaotic systems with application in medical image encryption 9:25911–25
21. Mondal A, Goswami RT (2020) Enhanced honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security. Microprocessors and Microsystems 81:103719
22. Liu P (2020) Public-Eey Encryption secure against related randomness attacks for improved end-to-end security of cloud/edge computing 8:16750–9
23. Pavani V, Krishna PS, Gopi AP, Narayana VL (2020) Secure data storage and accessing in cloud computing using enhanced group based cryptography mechanism. Materials Today: Proceedings
24. Ke G, Wang H, Zhou S, Zhang H (2019) Encryption of medical image with most significant bit and high capacity in piecewise linear chaos graphics. Measurement. 135:385–391
25. Ali TS, Ali R (2020) A novel medical image signcryption scheme using TLTS and Henon chaotic map. IEEE Access 8:71974–71992
26. Afzal I, Parah SA, Hurrah NN, Song OY (2020) Secure patient data transmission on resource constrained platform. Multimedia Tools and Applications. 1-26
27. Masood F, Driss M, Boulila W, Ahmad J, Rehman SU, Jan SU, Qayyum A, Buchanan WJ (2021) A lightweight Chaos-based medical image encryption scheme using random shuffling and XOR operations. Wireless personal communications. 1-29
28. Lakshmi C, Thenmozhi K, Rayappan JB, Rajagopalan S, Amirtharajan R, Chidambaram N (2020) Neural-assisted image-dependent encryption scheme for medical image cloud storage. Neural Computing and Applications. 1-4
29. Mall PK, Singh PK, Yadav D (2019) Glcm based feature extraction and medical x-ray image classification using machine learning techniques. In 2019 IEEE Conference on Information and Communication Technology. 1-6
30. Li W, Fan L, Wang Z, Ma C, Cui X (2021) Tackling mode collapse in multi-generator GANs with orthogonal vectors. Pattern Recogn 110:107646
31. Thanh-Tung H, Tran T (2020) Catastrophic forgetting and mode collapse in GANs. In2020 International Joint Conference on Neural Networks (IJCNN) IEEE. 1-10
32. Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R, Lanczi L (2014) The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans Med Imaging 34(10):1993–2024
33. Jaeger S, Candemir S, Antani S, Wáng YX, Lu PX, Thoma G (2014) Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. Quantitative imaging in medicine and surgery 4(6):475–477

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

K. L. Neela received her B.E. degree in Computer Science and Engineering in 2006 from the Oxford Engineering College, Trichy and Master's degree in Computer Science and Engineering in 2008 from J.J. college of Engineering and Technology, Trichy. She received Ph.D degree in Computer Science and Engineering from Anna University, Chennai. Presently, she is working as Assistant Professor in University College of Engineering, Thirukkuvalai, Tamil Nadu, India. She has more than 10 years of teaching experience. She has published more than 10 International journals in areas such as Network Security and Cloud Computing. She has attended 5 different FDP to enrich her career.

V. Kavitha obtained her B.E degree in Computer Science and Engineering in 1996 from the Norrul Islam College of and ME degree in Computer Science and Engineering in 2000 from Mepco Schlenk Engineering College. She received PhD degree in Computer Science and Engineering from Anna University, Chennai in the year 2009. Right from 1996, she is in the Department of Computer Science & Engineering under various designations. Presently she is working as Associate Prof in the Department of CSE at University College of Engineering, Kancheepuram, Tamil Nadu, India. Currently, under her guidance twelve research scholars are pursuing PhD as full time and part time. Her research interests are Wireless networks, Mobile Computing, Network Security, Wireless Sensor Networks, Image Processing; Cloud Computing. She has published 5 National journal and 30 International journals in areas such as Network security, Mobile Computing, wireless network security, and Cloud Computing.