cryptography

MDPI

Article

Neural Crypto-Coding Based Approach to Enhance the Security of Images over the Untrusted Cloud Environment

Pallavi Kulkarni [1,*], Rajashri Khanai [2], Dattaprasad Torse [1], Nalini Iyer [3] and Gururaj Bindagi [4]

[1] Department of Electronics and Communication Engineering, KLE Dr. MSSCET, Belgaum 590008, India; datorse@klescet.ac.in
[2] Department of Computer Science Engineering, KLE Dr. MSSCET, Belgaum 590008, India; rajashrikhanai@klescet.ac.in
[3] Department of Electronics and Communication Engineering, KLE Technological University, Hubli 580031, India; nalinic@kletech.ac.in
[4] Platform Architect, Novartis, Hyderabad 500081, India; gururajnb@yahoo.com
* Correspondence: pallavik15@gmail.com

Abstract: The cloud provides on-demand, high-quality services to its users without the burden of managing hardware and software. Though the users benefit from the remote services provided by the cloud, they do not have their personal data in their physical possession. This certainly poses new security threats for personal and confidential data, bringing the focus back on trusting the use of the cloud for sensitive data. The benefits of the cloud outweigh the concerns raised earlier, and with an increase in cloud usage, it becomes more important for security services to evolve in order to address the ever-changing threat landscape. Advanced encryption standard (AES), being one of the most widely used encryption techniques, has inherent disadvantages related to the secret key that is shared, and predictable patterns in subkey generation. In addition, since cloud storage involves data transfer over a wireless channel, it is important to address the effect of noise and multipath propagation on the transmitted data. Catering to this problem, we propose a new approach—the secure and reliable neural cryptcoding (SARNC) technique—which provides a superior algorithm, dealing with better encryption techniques combined with channel coding. A chain is as strong as the weakest link and, in the case of symmetric key encryption, the weakest link is the shared key. In order to overcome this limitation, we propose an approach wherein the key used for cryptographic purposes is different from the key shared between the sender and the receiver. The shared key is used to derive the secret private key, which is generated by the neural key exchange protocol. In addition, the proposed approach emphasizes strengthening the sub-key generation process and integrating advanced encryption standard (AES) with low-density parity check (LDPC) codes to provide end-to-end security and reliability over wireless channels. The proposed technique was tested against research done in related areas. A comparative study shows a significant improvement in PSNR, MSE, and the structural similarity index (SSIM). The key strength analysis was carried out to understand the strength and weaknesses of the keys generated.

Keywords: cloud computing; image; neural key exchange protocol; LDPC coding; cryptcoding; subkey; Khazad function; security; reliability

Citation: Kulkarni, P.; Khanai, R.; Torse, D.; Iyer, N.; Bindagi, G. Neural Crypto-Coding Based Approach to Enhance the Security of Images over the Untrusted Cloud Environment. *Cryptography* **2023**, *7*, 23. https://doi.org/10.3390/cryptography7020023

Academic Editor: Cheng-Chi Lee

Received: 2 November 2022
Revised: 7 April 2023
Accepted: 20 April 2023
Published: 4 May 2023

1. Introduction

Cloud computing has emerged as a promising technology that has changed the way we do things.

Cloud adoption by enterprises is increasing at a rate of more than 20%. There has been increased adoption of smart devices by the end users, partly due to the coronavirus pandemic. The current trend of exchanging information primarily involves multimedia data, especially images. The communication and transmission of images are not limited to the everyday life of a common person, but have wider applications in the field of industries,

the military, and medicine. Around 70% of internet traffic is dominated by images. As per the IBM Cost of Data Breach Report 2022, 45% of data breaches are cloud-based. The average time taken to find and prevent a data breach is 277 days. This reinforces the need to take a fresh look at security, especially involving sensitive information, since the security provided by cloud service providers (CSP) [1] has not stopped frequent security breaches adversely impacting organizations and users. Numerous theories and techniques have been proposed to tackle the problem of data security. Encryption, steganography, and watermarking techniques [2] are widely used to secure the images. Among these, encryption is the most efficient and commonly used method. One of the approaches to protect the privacy and integrity of outsourced data is to encrypt it before storing it in the cloud [3,4]. Users can upload an encrypted file that is foolproof from any tampering.

Security is always considered a shared responsibility between two parties. The focus is shifting towards users taking control of certain aspects of security, thus protecting sensitive personal identifiable information (PII), while CSP enables the platform to store and share the data. In line with that principle, we propose an approach that empowers users to take control of securing the images involving sensitive information. The approach extends the concepts of bring your own key (BYOK), in conjunction with bring your own algorithm (BYOA). It further adds additional controls, which ensure that the key provided by the users is different from the key used for encrypting the images. Our approach proposes that the encryption of sensitive information happens at edge devices and that the cloud is used for storage purposes only.

Our work is focused on securing the images, and AES is the preferred choice because it is the most popular and widely used symmetric key encryption algorithm [5]. Any person who possesses the key can encrypt or decrypt the message successfully. A major limitation of any symmetric key algorithm is that it is dependent on

1. Sharing the secret key between two parties.
2. Keeping the secret key secure from the intruder so that communication is not compromised.

The evolution of new technology led to the development of new cryptographic algorithms as well as new cryptanalysis methods. This will again give rise to the need for strong, secure cryptographic algorithms [6,7]. Although the AES algorithm has good confusion and diffusion properties, a weak key could make it vulnerable to attacks. Subkeys generated by the key schedule algorithm (KSA) in AES have a linear relationship with each other and the secret key. A weak KSA reduces the overall security of the cipher. There has been less emphasis on key generation and subkey generation compared to the strong encryption algorithm. In our proposed SARNC method, we use a neural key exchange protocol to generate the actual key of encryption/decryption, which is different from the shared secret key. In the process of key generation, tree parity machines (TPMs) are used on both sides. Final synchronized weights between TPMs are used as a key [8]. The suggested method also generates the subkeys by the Khazad function, which increases the complexity between the subkeys and the secret key.

Another issue addressed in our work is channel errors occurring during the transmission of an encrypted file. Communication via a wireless communication medium is mostly noisy and exposed to invaders. Attenuation, bandwidth limitations, multipath propagation, etc., all affect the efficiency of a wireless channel. To guard the data against channel errors, error correction codes (ECC) are used. We have combined AES with a low-density parity-check code (LDPC) [9], which is a linear block code with a performance close to Shannon's limit. It is highly popular and widely used because of its low decoding complexity and flexible structure. A pictorial representation of the proposed approach is given in Figure 1.

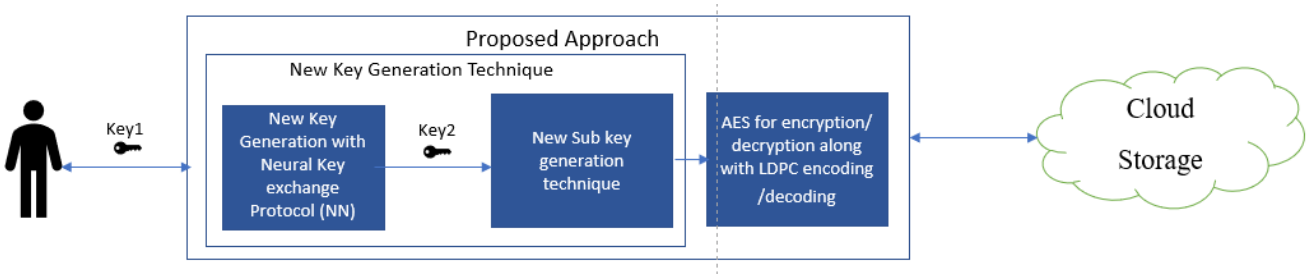

Figure 1. Proposed Approach.

The proposed work is summarized as follows:

1. A new key generation technique for securing the shared key of AES and a new subkey generation technique to strengthen the KSA.
2. Instead of treating encryption and encoding as two separate steps, our approach combines them into a single step. This helps to achieve the security and reliability of image data over the internet.
3. The new technique successfully passes the key strength analysis tests, such as frequency test, bit independence test, and bitwise uncorrelation test.

The paper is divided into the following sections: Section 2 explains the related work which helps us understand current trends and technology. In Section 3, we present the system architecture and model, and the projected goal. Then, complete explanation of the scheme is given in Section 4 under Methodology. Result analysis concerning security and performance is carried out in Section 5, followed by the conclusion of the paper.

2. Related Work, Research Gap, and Problem Formulation

Extensive research work was conducted to understand the strength and weaknesses of the various approaches. Some of the key papers are highlighted here.

Ramachandran et al., in reference [1] focused on providing security in an untrusted cloud environment. In this mechanism, a master key is used to generate public and private keys for encryption/decryption. The key distribution center (KDC) is responsible for key generation and distribution. An assessment of the proposed method is carried out using communication cost, and encryption/decryption time. The proposed method has seen improvement in communication costs and encryption/decryption time.

In order to overcome the disadvantages associated with AES subkey generation, Rahul Saha et al., in their paper [5], modified the traditional key expansion algorithm by using the symmetric random function generator (SRFG). This adds randomness to the generated key. The proposed technique is tested against related and fault tolerance attacks. As compared to original AES, the proposed method has better confusion properties and Avalanche effect. The limitation of the work is the time taken by the key generation module, which is the trade-off between security and time.

M. Zeghid et al. [6] proposed a modified AES algorithm to protect the confidentiality of image data from unauthorized access. The focus in this study is on images characterized by reduced entropy. The modified encryption scheme consists of a keystream generator consisting of an A5/1 keystream generator and a W7 keystream generator. The A5/1 is made up of three linear feedback shift registers (LFSRs). W7 consists of the control unit and a function unit. The function unit is accountable for the keystream generation. The authors implemented both AES and modified AES algorithms, and compared the results to show the superiority of the modified AES algorithm.

The authors Mayank Gupta et al., in reference [10], presented a new method to secure the secret sharing of an image between two entities. They employed Shamir's scheme to create the shares of an image. The key is generated by the tree parity machine. The synchronized weight between the two parties becomes the key to the encryption. This key is used for the encryption of shares. The results of the experiments, such as correlation,

RMSE, PSNR, time complexity analysis, and security analysis, show that the proposed algorithm can be used for secure image communications.

LI Ning et al. [11] proposed to combine AES and LDPC in a single step, to achieve security and reliability for satellite communication. A new round key generation technique is used to overcome the disadvantages of the traditional round key generation technique, for example, the linear relationship between the subkeys generated. The proposed round key generation algorithm is based on the modified Khazad function.

In their work, Mona F. M. Mursi et al. [12] combined hybrid chaotic encryption with LDPC. The image is transformed by applying FRFT and then encrypted using an Arnold cat map for confusion and a Hénon map for diffusion. This encrypted image is encoded using LDPC coding. In this paper, the analysis of results is carried out only concerning BER, FER, and PSNR. Emphasis is not given to analyzing the security aspect of the proposed algorithm.

The JSALE method proposed by Eran Pisek et al. [13] incorporates all the features of AES by interlacing some operations used in AES with the layers of a QC-LDPC code. This method provides high security with low BER and less hardware complexity.

A key strength analysis for different key generation and encryption techniques was carried out by Shazia Afzal et al. [14] in their work. Using different statistical tests, the authors identified the strengths of different key schedule algorithms.

The paper [15] presented by Shakir et al. proposed a novel technique that integrates the Haar wavelet transform with the AES. A chaotic logistic map is used for pixel shuffling. Distinct frequency domains of the image, i.e., estimate coefficient (LL) and detail confidence (LH, HL, and HH) are obtained using the Haar wavelet transform. AES encryption is applied to the lower frequency band (LL). The resulting image is scuffled by a chaotic logistic map to further improve the encryption strength. This makes malicious rebuilding very challenging. The proposed method performed well across multiple images and attained a better level of image security and a lower level of image degradation.

Jie Liu et al. [16] used a hyper-chaotic system with an LDPC code. This technique implements a pseudorandom sequence generator that is constructed using a hyperchaotic system for scrambling the plaintext. This is encoded by the LDPC encoder and then encrypted by the permutation box. This helps to improve security and reliability.

Alireza Arab et al., in paper [17], proposed an image encryption algorithm, which uses a chaos system to generate the key and the modified AES algorithm for encryption. In a chaotic system, a small change in input makes a major change in the output. The chaos system improves the safety of image encryption algorithms. An Arnold chaos system is used to generate the key. The AES algorithm is modified by replacing proposed propagation operations with the permutable operation and replacing the linear transformation operation with the column integration operation. The advantage of this method includes reduced time complexity and increased diffusion ability of the algorithm. The keyspace analysis shows that the proposed method successfully resists brute-force attacks.

The goal of the research presented in [18] by Ziaur Rahman et al. was to secure an IoT-based smart home. The generation of a key was performed with the help of chaos and logistic maps. More randomness and computational unpredictability was added by using this method.

The paper [19], presented by Lakshmi et al., focuses on securing medical images stored in the cloud. In this work, the dynamic keys are generated by the back propagation network (BPN), in which the distinctive features of an image are taken as input to the BPN. Thus, the keys generated are unique to images. The generated keys are used as an initial seed for confusion and diffusion sequence generation through a Hopfield neural network (HNN). The detailed security analysis carried out has confirmed the resiliency against the various attacks.

In the paper [20], Vishruti Kakkad et al. used biometric authentication with encryption to ensure the security of images in a cloud environment. Biometric authentication is required to upload and download the file to/from the cloud. Initially, the image is compressed using discrete wavelet transform (DWT), then the hash value is calculated by SHA

followed by the Blowfish algorithm for encryption. The encrypted file is further divided into three equal parts and its hash value is calculated. Those chunks, along with their hash values, form a hash table which is stored in the database. The proposed technique adds an extra level of security to the images stored in the cloud.

The above survey observations are tabulated in Table 1 as shown below:

Table 1. Survey observations.

Paper	Correlation Coefficient	Histogram Analysis	Entropy	NPCR	UACI	BER	PSNR	MSE	SSIM	Key Strength Analysis
[1]	Communication cost, encryption, and decryption time analysis									
[10]	✓						✓	✓		
[6]	✓		✓				✓			
[17]	✓	✓	✓	✓	✓					
[19]	✓	✓	✓							
[20]	Analysis to check Accuracy, cost and devices required									
[5]	Related key attack analysis, Fault injection analysis, Differential and Linear cryptanalysis									
[18]	Cryptanalysis, Calculation of key generation time analysis									
[15]	✓	✓	✓				✓	✓		
[12]		✓				✓	✓			
[16]			✓			✓				
[11]						✓	✓			
[13]						✓				
[14]										✓
Proposed Approach (SARNC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

From the above survey, it is concluded that:

- Most of the approaches involve chaos-based and/or hybrid techniques for key generation and encryption. However, of the approaches are limited to academic interest rather than real-world application, because of problems such as insufficient security analysis, flawed design methodology, and low efficiency.
- Few research papers focus on key generation with minimal or no key strength analysis.
- None of the researchers provided a holistic end-to-end solution ensuring the security and reliability of data at rest and during transmission.

To overcome these findings, we propose a comprehensive security solution comprised of the following key points:

- The scope of our work focuses on AES, since it is widely used in the industry because of its versatility and ease of use.
- Multi-layered architecture comprising key generation using neural key exchange protocol from the shared secret key.
- Improving nonlinearity of subkey using the Khazad function.
- Combining encryption and encoding in a single step to provide secure and reliable data transfer.
- Detailed analysis comprising statistical, differential, and key strength analysis.

3. System Architecture

The proposed work follows a cloud storage architecture that has three entities: data owner, cloud service provider (CSP), and authorized user [21,22]. Figure 2 illustrates the interaction between the three entities of the system.

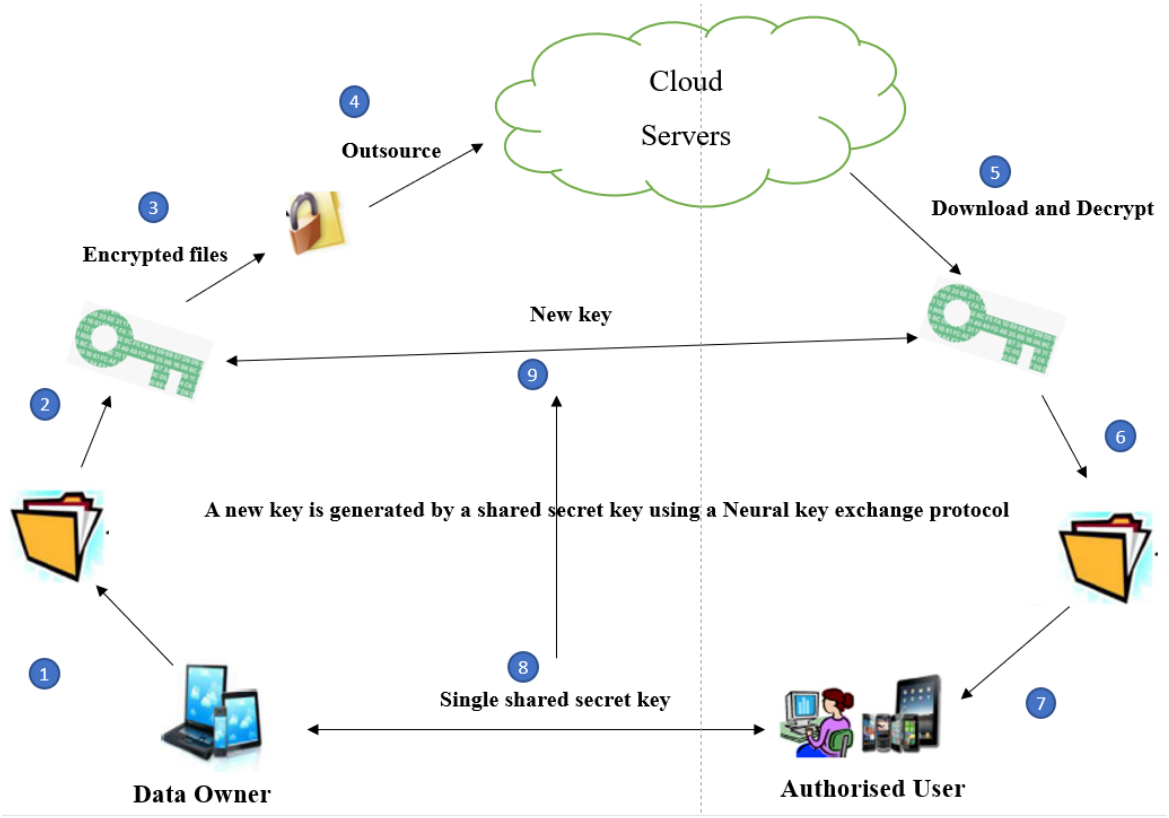

Figure 2. Proposed system architecture.

The data owner (DO) is the one who possesses data and intends to consume the cloud data services.

The cloud service provider (CSP) is an establishment that offers various services to its users, one of which is storage service.

The authorized user (AU) is authorized to use the files stored by the data owner. He/she can download the files from the cloud in an encrypted form and can decrypt them by generating the actual key using a shared secret key to get the original data.

The system architecture explains the interaction between the three entities of cloud system architecture.

(1) The data owner has sensitive data/files (personally identifiable information) that needs to be stored in the cloud in encrypted form (detailed in the proposed system architecture: steps 1 through 4).
(2) Data owners and authorized users share a single secret key. On both sides, a new key is generated for encryption/decryption using the neural key exchange protocol. The user can be an owner as well (detailed in the proposed system architecture: steps 8 and 9).
(3) When an authorized user wants to recover the original data/file, he/she downloads the encrypted files from the cloud, generates the key and executes the decryption algorithm, and gets back the corresponding original data/files (detailed in the proposed system architecture: steps 5, 6 and 7).

4. Methodology

In this section, we explain the key [10] and subkey generation [11], and cryptcoding using AES-128 with the LDPC coding technique. A unique kind of feed-forward neural network referred to as a tree parity machine (TPM) is used to generate the private key (Key2). The Khazad function is used to improve the AES subkey generation. The block diagrm shown in Figure 3 provides detailed steps involved in the end to end solution.

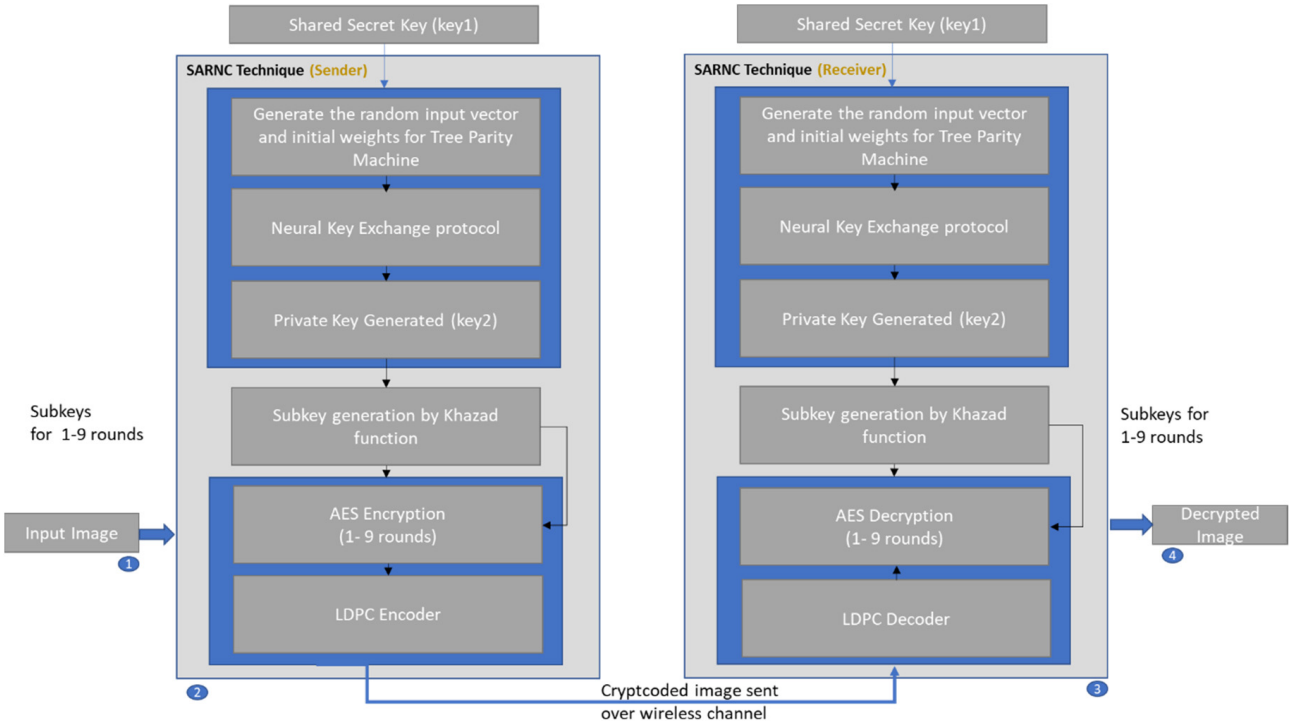

Figure 3. Detailed flow diagram.

Ddetails of internal working are described in the following sections.

4.1. Key Generation by Neural Key Exchange Protocol

1: Let key1 be the key shared between two parties and key2 be the new key generated by the proposed algorithm.

$$key2 \rightarrow func(key1)$$

Key2 is generated using a neural key exchange protocol. Random input vector and initial random weights for TPM are generated by key1, which means that the new key, key2, is a function of key1.

2: Define the structure of NN by choosing the number of neurons in the input and hidden layer.

The structure of the neural network is defined by the length of the key shared between two parties. For example, if we take a 16 byte key, i.e., a 128 bit key, we can have:

Input layer—64/32/16 neurons, hidden layer—2/4/8 neurons, and output layer—1 neuron.

3: Consider two tree parity machines (TPM), for instance, TPM-X and TPM-Y. Both TPMs need to agree on the structure of NN based on key1.

In the process of key generation, the tree parity machine is constructed by both parties. They take the common input vectors I, H, M, and L. Using Algorithm 1, a TPM is constructed, and outputs are generated. Synchronization of weights is carried out if output τ_1 is equal to τ_2. After the synchronization, the Hebbian learning rule is used to update the weights. This process is carried out until the weights are equal. Respective weights are used as keys after full synchronization.

Tree parity machine (TPM)

τ is the final output. The output τ_1 and τ_2 generated by TPM-X and TPM-Y by Algorithm 1, are compared for equality. If they are equal, then the synchronization of weights is carried out using Algorithm 2. Individual weights are used as keys only if synchronization is achieved between the two parity machines.

Synchronizing the weights

The Hebbian learning rule is used for updating weights and is given in Algorithm 2.

Algorithm 1: Tree Parity Machine

Given *I[n]*
I is the input vector of size n.
Given *H, M, L*
H—The number of hidden neurons.
M—The number of input neurons connected to hidden neurons.
L—Defines the range of each weight {−L, 0, +L}
Weights $W_{ij} = \{-L, \ldots .., 0, \ldots .., L\}$
$\sigma i = sgn \sum_{j=1}^{n} Wij * Iij$ (1)
is the activation function.
$\tau = \prod_{k}^{i=1} \sigma i$ (2)

Algorithm 2: Hebbian Rule

Input: $\{W, I, \sigma, \tau_1, \tau_2, _I\}$
for each (i, j) in W do,
$W_{i,j} \leftarrow W_{i,j} + X_{i,j} * \tau_1 * \theta\ (\sigma_i, \tau_1)\ *\ \theta\ (\tau_2\ , \tau_1)$ (3)
$W_{i,j} \leftarrow f_{clip}(W_{i,j})$ (4)
end for
where,
$\theta = \begin{cases} 1, & if\ \ \tau_1 = \tau_2 \\ 0, & otherwise \end{cases}$
$f_{clip} = \begin{cases} L, & if\ \ W_{i,j} > L \\ -L, & if\ \ W_{i,j} < -L \end{cases}$

4.2. Subkey Generation

The key schedule algorithm of AES has a predictable pattern of subkey generation that makes the algorithm prone to attacks. A robust KSA makes the cipher more resilient to linear and differential attacks. In our work, we have considered the idea of sub-key generation by the Khazad function, and the function is:

$$K_i = K_{i-8} \oplus H(S(K_{i-4})) \oplus C^{[(i-8)/4]} \quad (5)$$

$i = (8, 9, \ldots\ \ldots\ , 43)$

In which H is a linear diffusion function, S is a byte substitution function, and C is a round constant. The second-round keys are generated as:

$$K_8 = K_0 \oplus H(S(K_4)) \oplus C^0 \quad (6)$$

$$K_9 = K_1 \oplus H(S(K_5)) \oplus C^0 \quad (7)$$

$$K_{10} = K_2 \oplus H(S(K_6)) \oplus C^0 \quad (8)$$

$$K_{11} = K_3 \oplus H(S(K_7)) \oplus C^0 \quad (9)$$

The first-round subkey (K_0, K_1, K_2, K_3) *(K_0, K_1, K_2, K_3)* can only be generated from the original key. The next set of round keys are generated as follows:

$$K_4 = K_0 \oplus K_2 \quad (10)$$

$$K_5 = K_1 \oplus K_3 \quad (11)$$

$$K_6 = K_4 \oplus K_5 \quad (12)$$

$$K_7 = K_5 \oplus H(S(K_6)) \oplus C^0 \quad (13)$$

Assume that the attacker knows (K_4, K_5, K_6, K_7), but they still cannot construct (K_0, K_1, K_2, K_3), as K_7 only depends on K_5, and K_6, K_6 depends only on K_4 and K_5, and so on. The attacker needs to carry out 232 exhaustive attacks to get the first-round keys and 264 to guess the original key. Therefore, the proposed algorithm meets the safety requirements.

4.3. Block Diagram of AES–LDPC Cryptcoding

This method uses modified AES with LDPC coding that uses 128 bits of block length and key length. AES has a typical substitution and permutation network (SPN) architecture. Figure 4 depicts the block diagram of AES–LDPC coding (cryptcoding). The first 9 rounds are the same as the traditional AES with regard to proposed key and subkey generation. LDPC coding is embedded in the tenth round, after the substitution box (S-box). Therefore, we get the encryption control of the LDPC encoding round. The output of the tenth round is the desired ciphertext that needs to be stored in the cloud. Table 2 gives the details of parameters used for AES–LDPC cryptcoding [12,13].

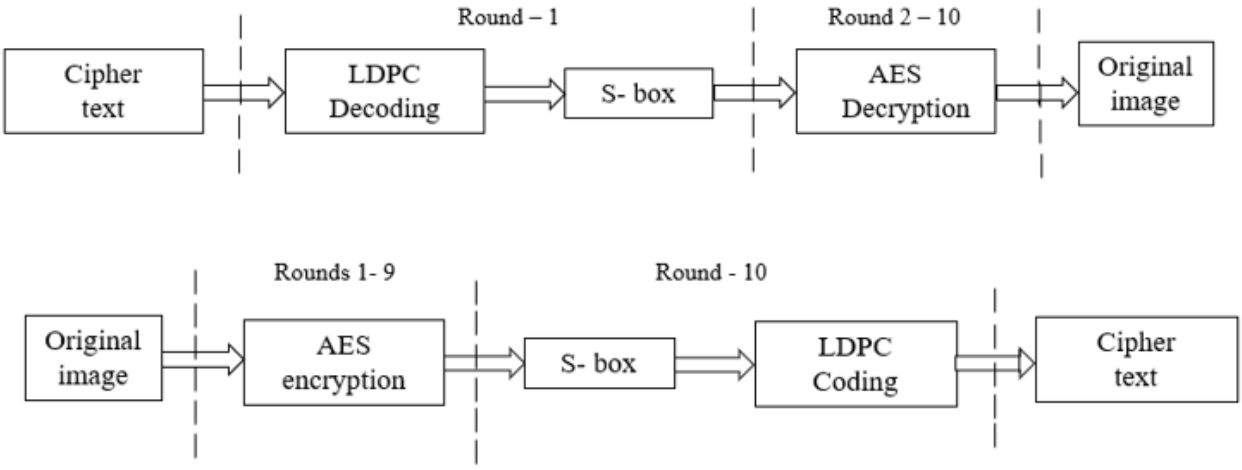

Figure 4. AES–LDPC coding.

Table 2. Particulars of the parameters used in the proposed scheme.

Modulation	BPSK
Coding	LDPC
Code rate	1/2
Frame size	64,800
No. of iterations	10
Channel	AWGN
Ciphering	AES-128 with LDPC coding

5. Security Analysis, Performance Evaluation, and Discussion

Simulation Environment

The proposed work was carried out on an AMD Ryzen 5 processor laptop with 8 GB memory and a Windows 10 operating system using MATLAB-R2020b software. A Dropbox open source cloud server was used for file storage. We used grayscale images for experimental purposes.

5.1. Key Strength Analysis of Proposed Key/Subkey Generation Technique

A key schedule algorithm (KSA) should have good confusion and diffusion properties. Any of the compromised subkeys should not reveal any information about other subkeys

or a secret key. In order to understand the key strength, three sets of statistical tests, namely, frequency, bit independence, and bitwise uncorrelation tests [14] were carried out.

5.1.1. Frequency Test

To find out the randomness in subkeys generated by the proposed KSA, we used the frequency test. This test is establishes the occurrence of the number of ones and zeroes in a random set of data. The KSA is said to pass the test if the proportion of zeroes and ones is close to 50%. Further testing is not required if the algorithm fails to clear the randomness (frequency) test.

Let a_0 and a_1 denote the number of zeroes and ones in an n-bit sequence and the statistics used is:

$$Z = (a_0 - a_1)^2/n \tag{14}$$

Results Discussion:

Figure 5 shows the results of the frequency test carried out for the proposed method and is compared with AES-128. From the figure, we can observe that randomness in the subkeys generated by the proposed method is more, which is about 90.2813% (average value of the nine rounds).

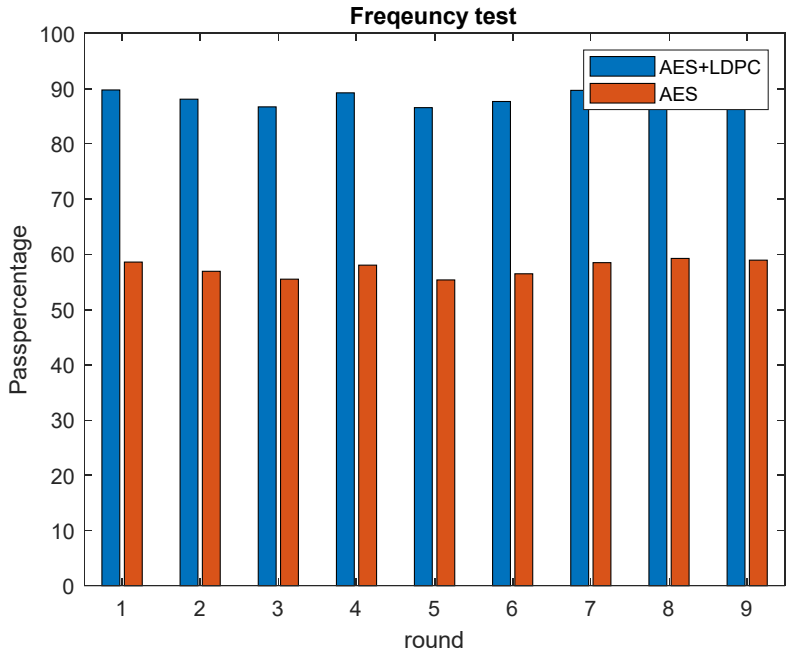

Figure 5. Frequency test for the proposed KSA.

5.1.2. Bit Independence Tests (BITs)

Using this test, we can find out how secret key and subkeys are related to each other, and how a single bit change in a key affects the subkeys. We define a function around which the tests are carried out as

$$f: (GF(2))^n \rightarrow (GF(2))^m$$

where n indicates input bits and m indicates output bits.

(i) Completeness (d_c): a function f is said to be complete if each output bit depends upon all input bits.
(ii) Avalanche effect (d_a): a function f has the avalanche effect if a one bit change in input affects more than half of the output bits.
(iii) Strict avalanche criteria (SAC-d_{sa}): a function f satisfies the SAC if the complement of a single bit in input affects more than half of the output bits.

The ideal values of d_c, d_a, and d_{sa} should be 1.

Results Discussion:

From the results tabulated in Table 3, we see that the proposed method has a good degree of completeness, avalanche effect, and the strict avalanche effect as d_c, d_a, are equal to 1, and d_{sa} is close to 1.

Table 3. Bit independence test (BIT).

	Completeness d_c	**Avalanche d_a**	**Strict Avalanche d_{sa}**
Proposed SARNC technique	1	1	0.904879
Original AES-128	0.7	0.7	0.605883

5.1.3. Bitwise Uncorrelation Tests (BUCT)

- A bitwise uncorrelation test finds out if all subkeys are bitwise uncorrelated with each other.
- A new sequence is generated by using Equation (15). Sequence generation is conducted by XORing all possible combinations of bits of subkeys X_i and X_j.

$$\text{Binary sequence} = (X_1 \oplus X_2) \,||\, (X_1 \oplus X_3) \,||\, \ldots\ldots \,||\, (X_2 \oplus X_3) \ldots\ldots \,||\, (X_i \oplus X_j) \,||\, \ldots\ldots \,||\, (X_{r-1} \oplus X_r) \quad (15)$$

where $(Xi \oplus Xj)$ = (Xi [1] $\oplus$ Xj) || (Xi[2] $\oplus$ Xj) || …… || $(Xi[L] \oplus Xj)$

Here, $i \neq j$, $i = 1, 2, \ldots, r$, $j = 1, 2, \ldots, r$.

(i) Frequency test: this test is the same as the one explained in Section 5.1.1. However, this test is carried out on the sequence generated by Equation (15).

(ii) Poker test: this test finds out how many times the p-bit block appears in the sequence derived from Equation (18). The sequence is divided into N non-overlapping blocks, each of length P. bi is the ith bit of a P-bit sequence. Equation (16) is used to find a distribution of P-bit blocks.

$$Z = \frac{2^P}{B} \sum_{i=1}^{2^P} (bi)^2 - B \quad (16)$$

Results Discussion:

The threshold level is set at 10%, which means that 10 out of 100 sequences generated by Equation (15) can be rejected i.e., minimum of 90% of the sequence should pass the test. The results tabulated in Table 4 suggest that the proposed techniques successfully pass the BUCT test, as the randomness factor is 98% and 94.5% of sequences pass the poker test. Figure 6 shows the plot that helps us to understand the behavior of the proposed method and AES-128 for the BUCT test. The proposed SARNC technique has a higher pass percentage compared to AES-128.

Table 4. Bitwise uncorrelation test (BUCT).

	Frequency Test in (%)	**Poker Test in (%)**
Proposed SARNC technique	98	94.5
Original AES-128	97	93.6

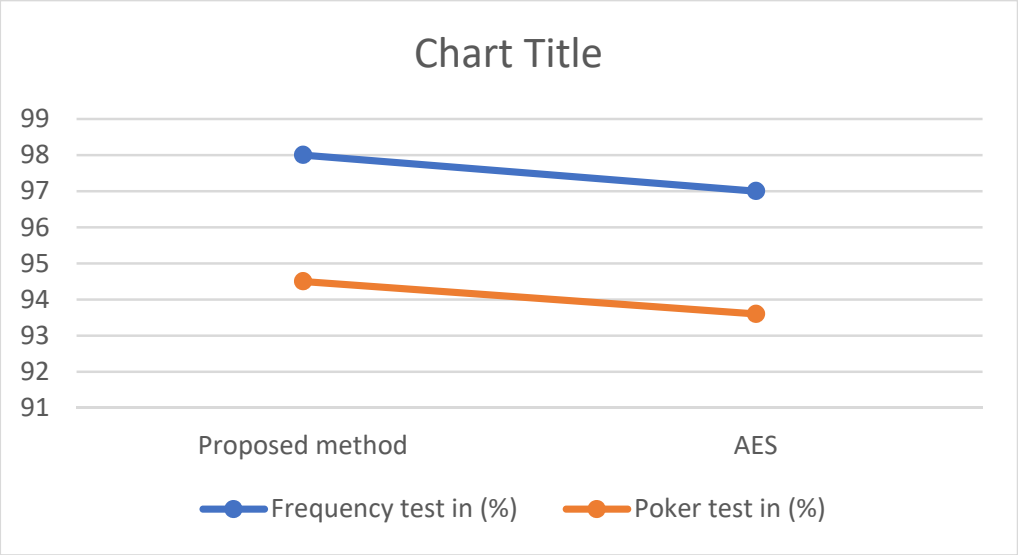

Figure 6. Bitwise uncorrelation test (BUCT).

5.2. Performance Parameters

Performance parameters are essential to find out the quality of the proposed security model. The assessment is carried out based on the parameters: correlation coefficient, NPCR, and UACI values, in which an original image is compared with a cryptcoded image. Peak signal-to-noise ratio (PSNR), mean square error (MSE), bit error rate (BER), and structural similarity (SSIM) [23] indicate the efficacy of the channel coding technique. This is calculated between an original and received decrypted image.

PSNR estimates the noiselessness of the cryptcoded image. The greater value of PSNR suggests a better quality of the received decrypted image. The PSNR is calculated as [15]:

$$PSNR = 10 \log \left[\frac{I^2}{MSE} \right] \tag{17}$$

where I is a pixel value in the image. The maximum value for a grayscale image is 255.

Mean square error (MSE) [15] is the parameter that estimates the error between the original and received decrypted image.

The *MSE* is computed as:

$$MSE = \frac{1}{[M * N]^2} \sum_{i=0}^{n} \sum_{j=1}^{m} \left(X_{ij} - Y_{ij} \right)^2 \tag{18}$$

M and N are two dimensions of an image.

X_{ij} and Y_{ij} are pixel intensity of an original and recovered image.

The main advantage of MSE and PSNR is to find the noise level in a reconstructed image. These parameters also help us analyze the efficiency of channel coding techniques used. Table 5 shows that we are achieving good results for PSNR and MSE, as the PSNR value is high and MSE is low.

Table 5. Comparison of performance parameters.

	Original AES-128	**Ref. [21]**	**Ref. [24]**	**Ref. [25]**	**Proposed SARNC Technique**
PSNR	36.1236	-	54.26	-	48.1648
MSE	16	-	0.24	-	1
SSIM	0.7495	-	0.99	-	0.99996
Correlation coefficient	−0.0121	−0.0036	-	0.001178542895092	−0.0074
NPCR (%)	99.4141	99.60	-	99.7570	99.5117
UACI (%)	33.2802	33.41	-	39.12	33.2837

Bit errors occurring during transmission are indicated by BER [23]. This is the number of bits received in error, divided by the total number of bits transmitted.

$$\text{BER} = \text{Errors}/\text{Total Number of Bits}$$

The structural similarity index (*SSIM*) measures the structural similarity between two images. The value of 1 suggests nearly identical images. *SSIM* is calculated as follows [16]:

$$SSIM = \frac{[2 * mu_1(p)mu_2(p) + c_1]}{\left[mu_1(p)^2 mu_2(p)^2 + c_1\right]} * \frac{[2 * cov(p) + c_2]}{\left[s_1(p)^2 + s_2(p)^2 + c_2\right]} \tag{19}$$

where, $mu_1(P)$ and $mu_2(p)$ are the mean value of seq_1 and seq_2 computed over a small *XY* window located around *P*; $s_1(p)$ and $s_2(p)$ are the standard deviations of seq_1, and seq_2 computed over the same window; and $cov(p)$ is the covariance between seq_1 and seq_2.

One of the purposes of encryption is to reduce the association between two pixels [17]. The reduced correlation value suggests an improved encryption effect and better security. The correlation coefficient is calculated using the equations given below:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{20}$$

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{21}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{22}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2 \tag{23}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2 \tag{24}$$

where,

x and *y*: values of the two adjacent pixels

N: Number of selected adjacent pixels

A study on the algorithm's sensitivity is accomplished by finding the number of pixels changing rate (NPCR) and the unified average changing intensity (UACI) [17]. E_1 and E_2 are the two encrypted images derived by changing a single bit of the original image.

$$X(i, j) = \begin{cases} 1, & if\ \ E_1(i, j) \neq E_2(i,\ j) \\ 0, & if\ \ E_1(i, j) = E_2(i,\ j) \end{cases} \tag{25}$$

where $E_1(i, j)$ and $E_2(i, j)$ define the grayscale area's value of a pixel in the (i, j) coordinate of the encrypted images E_1 and E_2.

NPCR and UACI values are calculated using Equations (26) and (27), as shown below:

$$NPCR = \frac{1}{x * y} \sum_{i,j} [X(i, j)] * 100 \tag{26}$$

$$UACI = \frac{1}{x * y} \sum_{i,j} (E_1(i, j) - E_2(i, j)) / \left(2^l - 1\right) * 100 \tag{27}$$

The parameters x and y indicate the dimensions of the original image. The greater values of UACI and NPCR ensure better security of the encryption algorithm.

Results Discussion:

A comparison study was carried out between proposed SAARC technique and AES-128, and is tabulated in Table 4. The greater values of PSNR, MSE, and SSIM indicate the better quality of received decrypted image. The greater values of the correlation coefficient, NPCR, and UACI are indicative of better quality of encryption algorithm. By referring to Table 5, we can conclude that the proposed SARNC technique provides improved security and reliability for the transmitted image.

Figures 7 and 8 show the variation of PSNR and BER with SNR.

Histogram Analysis.

An image histogram represents the number of pixels as a function of their intensity [24].

Results Discussion:

A comparative study was carried out between the original and received decrypted image histograms, as shown in Figure 9. The histogram of the original and received decrypted image is almost the same, with a negligible histogram error of 0.005157.

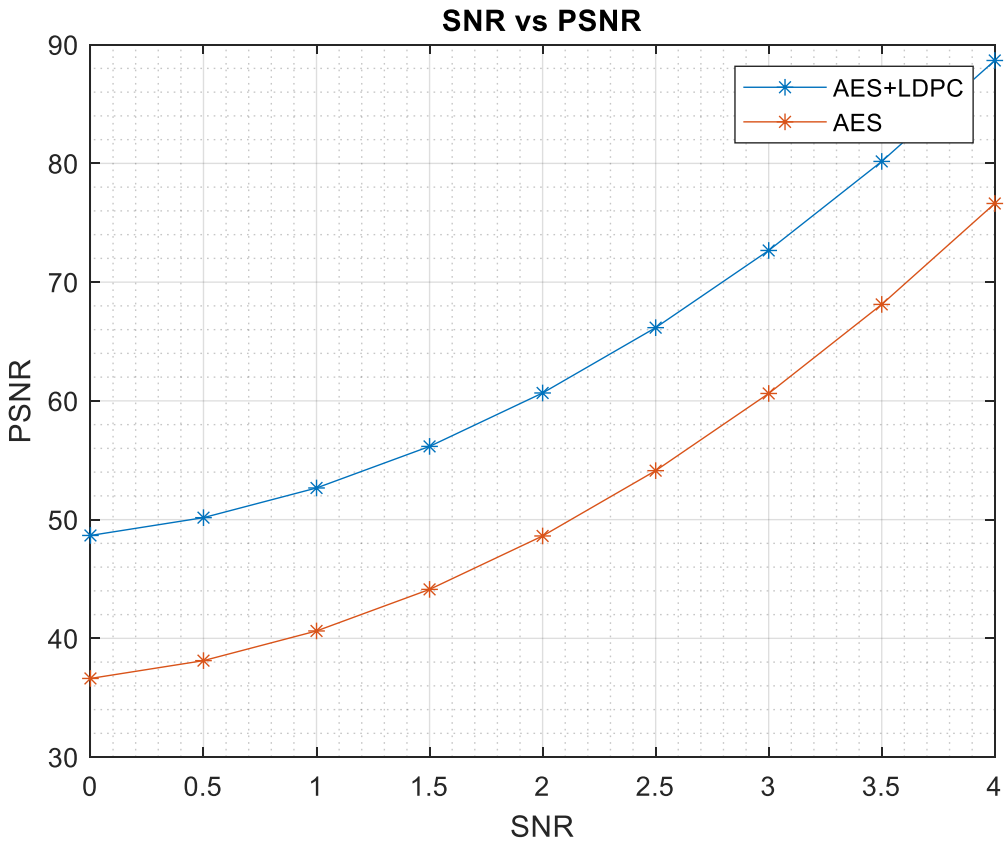

Figure 7. PSNR comparison of SARNC and AES-128.

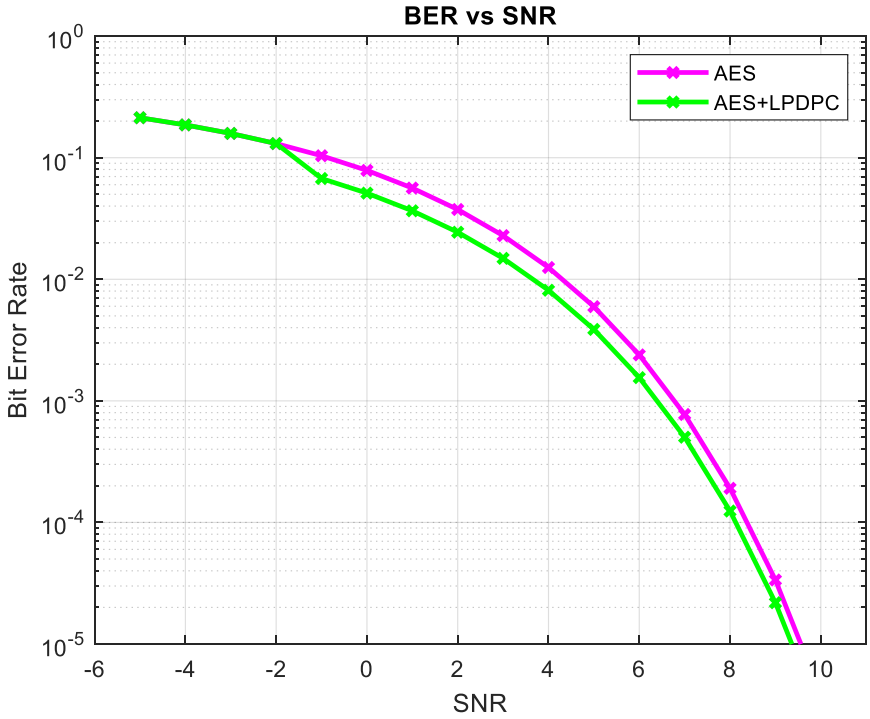

Figure 8. BER curve for proposed SARNC technique.

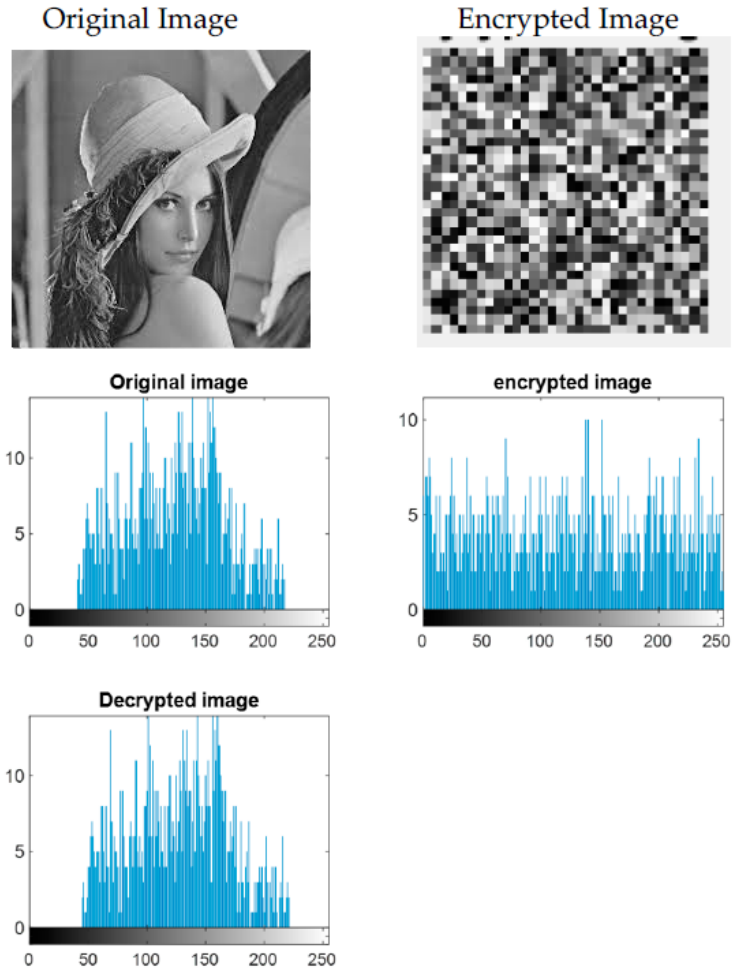

Figure 9. Result of histogram analysis for the SARNC technique.

6. Conclusions

In this paper, we proposed a secure and reliable neural cryptcoding technique (SARNC) for images in the untrusted cloud environment. A neural key exchange protocol was constructed in such a way that any compromise of a shared secret key will not reveal the actual key used for encryption. To achieve better key security, we propose a complex method of subkey generation, in which the attacker has to break two successive rounds of sub-keys to get the whole key bits. For a better utilization of bandwidth and time, the LDPC encoder is combined with modified AES. This enhances the diffusion and error correction ability with little additional complexity. The results of key strength analysis via a frequency test, bit independence test, and bitwise uncorrelation test show that the generated keys have a higher degree of randomness and better confusion and diffusion properties. We assessed the quality of the proposed scheme in terms of security and reliability with the help of PSNR, MSE, SSIM, correlation coefficient, NPCR, and UACI. The simulation results and the comparative study between the proposed and other existing techniques proves that the proposed SARNC technique offers better security and reliable performance.

Author Contributions: All authors contributed to the study's conception and design. Methodology and analysis were carried out by P.K. and G.B. A draft of the manuscript was prepared by P.K., R.K., D.T. and N.I. contributed to the final review and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ramachandran, B.; Subramaniam, K. Secure and efficient data forwarding in untrusted cloud environment. *Clust. Comput.* **2019**, *22*, 3727–3735. [CrossRef]
2. Wang, H.; Wu, S.; Chen, M.; Wang, W. Security protection between users and the mobile media cloud. *IEEE Commun. Mag.* **2014**, *52*, 73–79. [CrossRef]
3. Kaur, J.; Sharma, S. HESSIS: Hybrid Encryption Scheme for Secure Image Sharing in a Cloud Environment. In Proceedings of the International Conference on Advanced Informatics for Computing Research, Shimla, India, 14–15 July 2018; Springer: Singapore, 2018; pp. 204–216.
4. Pasupuleti, S.K.; Ramalingam, S.; Buyya, R. An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. *J. Netw. Comput. Appl.* **2016**, *64*, 12–22. [CrossRef]
5. Saha, R.; Geetha, G.; Kumar, G.; Kim, T.-H. RK-AES: An Improved Version of AES Using a New Key Generation Process with Random Keys. *Secur. Commun. Netw.* **2018**, *2018*, 9802475. [CrossRef]
6. Zeghid, M.; Machhout, M.; Khriji, L.; Baganne, A.; Tourki, R. A modified AES based algorithm for image encryption. *Int. J. Comput. Sci. Eng.* **2007**, *1*, 70–75.
7. Awan, I.A.; Shiraz, M.; Hashmi, M.U.; Shaheen, Q.; Akhtar, R.; Ditta, A. Secure framework enhancing AES algorithm in cloud computing. *Secur. Commun. Netw.* **2020**, *2020*, 8863345. [CrossRef]
8. Chourasia, S.; Bharadwaj, H.C.; Das, Q.; Agarwal, K.; Lavanya, K. Vectorized neural key exchange using tree parity machine. *Compusoft* **2019**, *8*, 3140–3145.
9. Wang, Z.-X.; Lou, Y.; Wang, W.-Q.; Zhang, M.; Li, X.-L. Research on the application of LDPC code in chaotic sequence image encryption. *Clust. Comput.* **2019**, *22*, 6359–6370. [CrossRef]
10. Gupta, M.; Gupta, M.; Deshmukh, M. Single secret image sharing scheme using neural cryptography. *Multimed. Tools Appl.* **2020**, *79*, 12183–12204. [CrossRef]
11. Li, N.; Lin, K.; Lin, W.; Deng, Z. A joint encryption and error correction method used in satellite communications. *China Commun.* **2014**, *11*, 70–79.
12. Mursi, M.F.M.; Ahmed, H.E.H.; El-Samie, F.E.A.; El-Aziem, A.H.A. Combination of Hybrid Chaotic Encryption and LDPC for Secure Transmission of Images over Wireless Networks. *Int. J. Image Graph. Signal Process.* **2014**, *6*, 8–16. [CrossRef]
13. Pisek, E.; Abu-Surra, S.; Taori, R.; Dunham, J.; Rajan, D. Enhanced cryptcoding: Joint security and advanced dual-step quasi-cyclic LDPC coding. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.
14. Afzal, S.; Yousaf, M.; Afzal, H.; Alharbe, N.; Mufti, M.R. Cryptographic strength evaluation of key schedule algorithms. *Secur. Commun. Netw.* **2020**, *2020*, 3189601. [CrossRef]

15. Shakir, H.R. An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling. *Multimed. Tools Appl.* **2019**, *78*, 26073–26087. [CrossRef]
16. Liu, J.; Tong, X.; Liu, Y.; Zhang, M.; Ma, J. A joint encryption and error correction scheme based on chaos and LDPC. *Nonlinear Dyn.* **2018**, *93*, 1149–1163. [CrossRef]
17. Arab, A.; Rostami, M.J.; Ghavami, B. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **2019**, *75*, 6663–6682. [CrossRef]
18. Rahman, Z.; Yi, X.; Billah, M.; Sumi, M.; Anwar, A. Enhancing AES Using Chaos and Logistic Map-Based Key Generation Technique for Securing IoT-Based Smart Home. *Electronics* **2022**, *11*, 1083. [CrossRef]
19. Lakshmi, C.; Thenmozhi, K.; Rayappan, J.B.B.; Rajagopalan, S.; Amirtharajan, R.; Chidambaram, N. Neural-assisted image-dependent encryption scheme for medical image cloud storage. *Neural Comput. Appl.* **2021**, *33*, 6671–6684. [CrossRef]
20. Kakkad, V.; Patel, M.; Shah, M. Biometric authentication and image encryption for image security in cloud framework. *Multiscale Multidiscip. Model. Exp. Des.* **2019**, *2*, 233–248. [CrossRef]
21. Wang, C.; Wang, Q.; Ren, K.; Cao, N.; Lou, W. Toward Secure and Dependable Storage Services in Cloud Computing. *IEEE Trans. Serv. Comput.* **2011**, *5*, 220–232. [CrossRef]
22. Sood, S.K. A combined approach to ensure data security in cloud computing. *J. Netw. Comput. Appl.* **2012**, *35*, 1831–1838. [CrossRef]
23. Elhoseny, M.; Ramírez-González, G.; Abu-Elnasr, O.M.; Shawkat, S.A.; Arunkumar, N.; Farouk, A. Secure medical data transmission model for IoT-based healthcare systems. *IEEE Access* **2018**, *6*, 20596–20608. [CrossRef]
24. Raja, S.P. Joint medical image compression–encryption in the cloud using multiscale transform-based image compression encoding techniques. *Sādhanā* **2019**, *44*, 28. [CrossRef]
25. Mondal, B.; Mandal, T. A light weight secure image encryption scheme based on chaos & DNA computing. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *29*, 499–504.