

RESEARCH ARTICLE

NeuroChaosCrypt: Revolutionizing Chaotic-Based Cryptosystem With Artificial Neural Networks—A Comparison With Traditional Cryptosystems

TALAL BONNY  **AND WAFAA AL NASSAN**

Department of Computer Engineering, University of Sharjah, Sharjah, United Arab Emirates


Corresponding author: Talal Bonny (tbonny@sharjah.ac.ae)

ABSTRACT Encryption systems play a vital role in the transfer of sensitive data, and the integration of chaotic systems into this domain has garnered significant attention. However, these systems often grapple with complexity and insufficient security, posing challenges for real-world implementation. Researchers introduced the synchronization techniques to fix these problems, which means making sure that the chaotic systems in both the transmitter and receiver systems behave in a way that can be understood so that accurate signal recovery can happen. Chaotic system synchronisation presents challenges and security risks besides the limited of the encryption keys make them subjected to attacks. Because of their advantages over traditional chaotic systems in terms of flexibility, adaptability, and computational efficiency, artificial neural networks, or ANNs, are being used more and more to study chaotic systems. This paper presents NeuroChaosCrypt, a novel cryptographic framework employing unique methodologies for secure data transmission. It utilizes an Artificial Neural Network (ANN)-based chaotic system at both transmitter and receiver, eliminating the need for synchronization. A comprehensive case study, including audio signal transmission, underscores NeuroChaosCrypt's efficacy. Comparison with a traditional encryption system integrating a Linear Quadratic Regulator (LQR) controller reveals comparable security levels, correlation coefficient (cc), Signal-to-Noise Ratio (SNR), Peak-to-Root Mean Square Distortion (PRD), and encryption time. NeuroChaosCrypt, enhanced by ANNs, excels in decryption speed, key-space coverage, and hardware implementation using field-programmable gate arrays (FPGAs). This methodology achieves a higher maximum frequency while requiring fewer logic units. The comparison offers valuable insights into audio encryption methods, aiding informed decision-making for selecting the most suitable solution based on specific application requirements. Finally, we introduce an application of the proposed NeuroChaosCrypt for image encryption to ensure that the study can exploit other data types for broader applicability.

INDEX TERMS ANN-based chaotic systems, jerk chaotic system, hardware implementation, FPGA, image cryptosystem, security.

I. INTRODUCTION

The protection and privacy of sensitive data are the highest priority in the current digital age. Data encryption methods are extensively utilised for protecting data from unauthorized access or interception. An area of great interest in cryptography is the application of chaotic systems. Chaotic systems

The associate editor coordinating the review of this manuscript and approving it for publication was Congduan Li .

are known for their intricate and unpredictable behavior, making them suitable for generating cryptographic keys and enhancing encryption algorithms [1], [2].

Chaotic systems, characterized by their sensitivity to initial conditions and the occurrence of nonlinear dynamics, have garnered considerable attention in various scientific disciplines. These systems exhibit behavior that appears random yet possesses underlying patterns and structures. Encryption systems incorporating chaotic systems have gained

prominence, using chaotic signals as encryption components [3], [4], [5], [6], [7]. These signals function as encryption keys, pseudo-random number generators, or the foundation for cryptographic transformations [7], [8]. Integrating chaotic systems into encryption adds complexity, making it tougher for unauthorized parties to decipher encrypted data. Chaotic-based encryption spans image encryption, secure communication protocols, and data storage systems [9], [10], [11].

Chaotic synchronization occurs when multiple oscillators are linked or influence one another, ensuring coordinated phase changes between systems. This synchronization is vital for secure communication and data transmission, ensuring sender and receiver systems exhibit similar chaotic behaviors for successful transmission and decryption [12], [13], [14]. The Linear Quadratic Regulator (LQR) is a widely used technique for chaotic system synchronization [15], [16]. LQR facilitates synchronization between master and slave chaotic systems, aligning the slave's behavior with the desired master behavior, pivotal for reliable and secure communication [17], [18].

Artificial Neural Networks (ANNs) have revolutionized various fields [19] especially modeling chaotic systems through training on chaotic-generated data [20], [21], [22]. This capability to mimic chaotic behavior opens new encryption avenues [23], [24], allowing ANNs to replicate chaotic dynamics for secure encryption key generation and algorithm development. ANNs offer flexibility, adaptability, and computational efficiency compared to traditional chaotic systems.

Field-Programmable Gate Arrays (FPGAs) offer a versatile platform for efficient and secure encryption system implementation [25], [26]. They provide high-speed parallel processing and can implement complex encryption algorithms in real-time, ideal for applications requiring fast and secure data processing [27], [28], [29]. FPGA-based hardware implementation ensures efficient resource utilization, enhanced security, and customization based on specific application needs [8].

The motivation for this study arises from the difficulties and constraints presented by traditional chaotic encryption systems, which need synchronization techniques to guarantee coherent communication between the transmitter and receiver. The efficacy and integrity of the encryption process are compromised when synchronization methods are used, since they add complexity and possible weaknesses. Furthermore, traditional chaotic encryption systems could be vulnerable to assaults since they depend on a limited set of keys that are derived from system characteristics. An alternative method for simulating and modelling chaotic systems is provided by artificial neural networks (ANNs). They provide benefits including adaptability, flexibility, and computing efficiency and can reduce the requirement for synchronisation. This paper aims to present the NeuroChaosCrypt, a new type of cryptography system. This framework use artificial neural networks (ANNs) to precisely depict and simulate

chaotic systems, guaranteeing secure data transmission. The research seeks to establish the efficacy and superiority of NeuroChaosCrypt over a traditional encryption system that integrates a chaotic system with a LQR controller. The article presents a detailed examination of audio encryption through a case study, evaluating the effectiveness, level of protection, and feasibility of two systems using various metrics and tests. The study also includes detailed information about the hardware development for both systems that makes use of field-programmable gate arrays (FPGAs). It evaluates their effectiveness and resource use as well.

Our main technical contributions are as follows:

- 1) Introducing NeuroChaosCrypt, an innovative cryptographic framework that uses ANNs to model and simulate chaotic systems for secure data transmission.
- 2) Using both NeuroChaosCrypt and a traditional encryption method that depends on a chaotic system and an LQR controller in an in-depth case study on audio encryption.
- 3) Utilising a variety of parameters to compare the security and performance of the two systems and provide insightful analysis of their advantages and disadvantages.
- 4) Assessing the FPGA-based hardware implementation of both systems and emphasising the benefits of NeuroChaosCrypt in terms of operating speed and resource consumption.

The structure of the paper is as follows: The traditional and NeuroChaosCrypt systems are presented in Section II; the audio encryption case study is described in Section III; the security performance of both systems is analyzed in Section IV; the FPGA implementation of both systems is discussed in Section V; the overall performance of both systems is compared in Section VI; and the paper is concluded with suggestions for future work in Section VII.

II. TRADITIONAL AND NEUROCHAOSCRYPT SYSTEMS

Traditional cryptographic systems have long protected confidential information that employs established techniques and algorithms. By using mathematical transformations and principles for both encryption and decryption, these systems provide a solid basis for data security. When employing chaotic systems for secure communication, synchronization becomes necessary. Achieving synchronization ensures that the chaotic dynamics exhibited by the transmitter and receiver, enabling successful encryption and decryption of the transmitted data.

To realize synchronization between chaotic systems, a synchronization controllers are often introduced. These controllers aim to establish and maintain coherence between chaotic behaviors of the communication systems. The cryptographic system becomes more sophisticated when synchronisation controllers are integrated. Typically, these controllers use feedback mechanisms to modify system parameters that affect the overall efficiency of the system.

Figure 1 illustrates the block diagram of Traditional Cryptosystem, consisting of two part; transmitter and receiver. On the transmitter side; the encryption algorithm that uses the chaotic oscillator to encrypt the sensitive data which could be bio-medical signal, image, or audio signal generated the encrypted signal that transmit through the communication channel to the receiver side.

On the receiving end, the encrypted signal is decrypted using the decryption function, which utilizes the chaotic oscillator to recreate the recovered message. Ensuring the effective recovery of the encrypted signal on the receiver side necessitates a complete alignment between the chaotic systems employed in the transmitter and the receiver. However, achieving such alignment is very improbable in practice due to the delicate nature of chaotic systems. If a slight change in one initial value of our chaotic system occurred in the receiver, this would produce a dramatic difference between the recovered and the original signals. To overcome this problem, synchronization between master and slave chaotic systems is required.

The challenges posed by synchronization mechanisms and their associated complexities have prompted a paradigm shift toward integration Artificial Neural Networks (ANNs) to obviate the need for chaotic system synchronization controllers. Figure 2 demonstrates the incorporation of ANN-based chaotic systems into cryptosystems. Notably, the transmitter and receiver components in the NeuroChaosCrypt resemble those in the traditional cryptosystem. The key distinction lies in the use of ANN-based chaotic oscillators in place of chaotic oscillators. This innovation effectively eliminates the necessity for synchronization controllers.

III. CASE STUDY: AUDIO SIGNAL TRANSMISSION THROUGH TRADITIONAL AND NEUROCHAOSCRYPT SYSTEMS

A. THE JERK CHAOTIC SYSTEM

This section presents the dynamic equations describing the attractors for the 3-D jerk system (proposed in [30])used in this study. which decribed with the equations (1).

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= ax_1 - bx_2 - x_3 - cx_2x_3 - x_1^2 - x_2^2 \end{aligned} \quad (1)$$

where $x_1, x_2,$ and x_3 are state variables. $a, b,$ and b are positive parameters. The system exhibits a chaotic behavior when the coefficient parameters take the following values $a = 7.5, b = 4,$ and $c = 0.3.$

The Lyapunov exponents (LE) of the hyperchaotic system are obtained as follows: $(L1, L2, L3) = (0.1631, 0, -1.1631).$

Figures 3a, 3b, and 3c illustrate phase attractors of the 3-D jerk chaotic oscillator. The fourth-order Runge-Kutta integrator with a fixed step size equal to (0.001) produces the simulation results. Where the constant parameters will take the following values $[a, b, c] = [7.5, 4, 03].$ The initial values of the variables are set to $[x_{1_0}, x_{2_0}, x_{3_0}] = [0.3, 0.2, 0.3].$

B. TRANSMISSION THROUGH NEROCHAOSCRYPT SYSTEM

Our NeroChaosCrypt system typically comprises multiple components that secure a message or data transmission. The Overall block diagram for the audio encryption system is shown in Figure 4 consisting of two sections; transmitter and receiver.

On the transmitter side, the audio encryption process is divided into two distinct phases, each with its own set of operations.

The first phase is signal masking; this phase aims to make the original audio signal more difficult to detect and decipher by using the ANN based jerk system. To do this, the original signal $s(t)$ combines with the output of the artificial neural network (ANN) jerk model, which produces random outputs $x_1, x_2, x_3,$ in accordance with Equation 2. The new signal $s_m(t)$ that is produced by this process is hidden and more challenging to recognise.

$$s_m(t) = s(t) + x_1(t) + x_2(t) + x_3(t) \quad (2)$$

After the signal is masked, it can be subjected to encryption using the encryption function during the second phase. This function requires four inputs: $x_1, x_2, x_3,$ which are the secret keys created by the ANN based jerk system, and $s_m(t),$ which is the masked signal produced in phase one. The encryption function generates the encrypted output $s_e(t)$ in the prescribed way by applying several mathematical operations to the given inputs:

$$\begin{aligned} s_e(t) &= (1 + x_1(t) + x_2(t) + x_3(t))^3 s_m(t) \\ &+ (x_1(t)x_2(t) + x_1(t)x_3(t) + x_2(t)x_3(t)) \end{aligned} \quad (3)$$

Splitting the Encryption process into two separate phases, this approach can increase the complexity of the signal and the unpredictability. Which in turn making the signal more difficult for an unauthorised party to intercept or decode.

On the receiver side, once the encrypted signal $s_e(t)$ the intended recipient receives, it must be decrypted to recover the original audio signal. The decryption process consists of two main phases, which are the reverse of the encryption process:

The first decryption phase involves applying the decryption function to the encrypted signal. The decryption function is designed to use the encrypted signal with the same inputs as the encryption function, which are $y_1, y_2,$ and y_3 generated by the ANN-based jerk system. Its goal is to recover the hidden signal by reversing the encryption process. This phase can be represented as follows:

$$s_d(t) = \frac{(x_1(t)x_2(t) + x_1(t)x_3(t) + x_2(t)x_3(t))}{((1 + x_1(t) + x_2(t) + x_3(t))^3)} \quad (4)$$

For the recovery of the original signal, the masked signal is unmasked in the second decryption phase. To do this, the actions taken during the masking phase are reversed. Specifically, the masked signal is passed through a ANN based jerk system that removes the added layer of chaos,

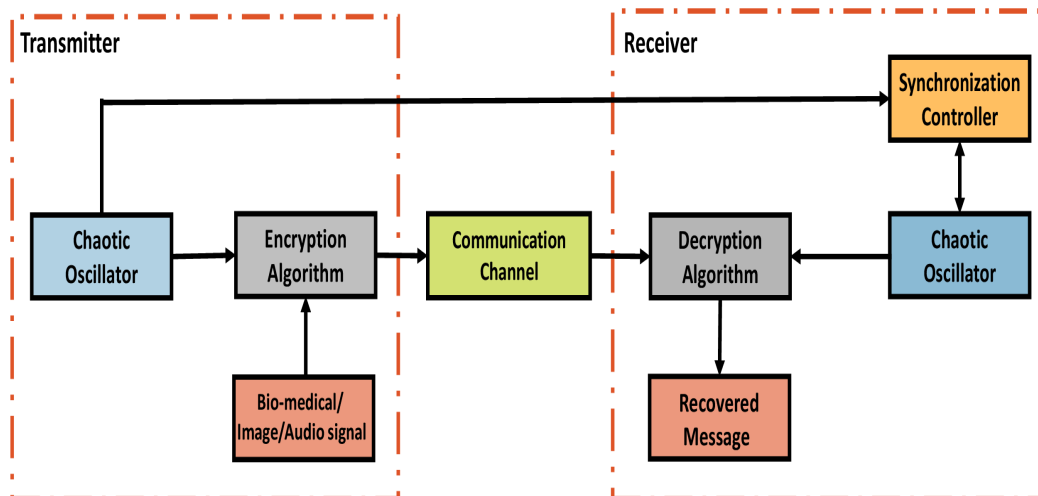


FIGURE 1. Model 1, the traditional encryption system using chaotic oscillator.

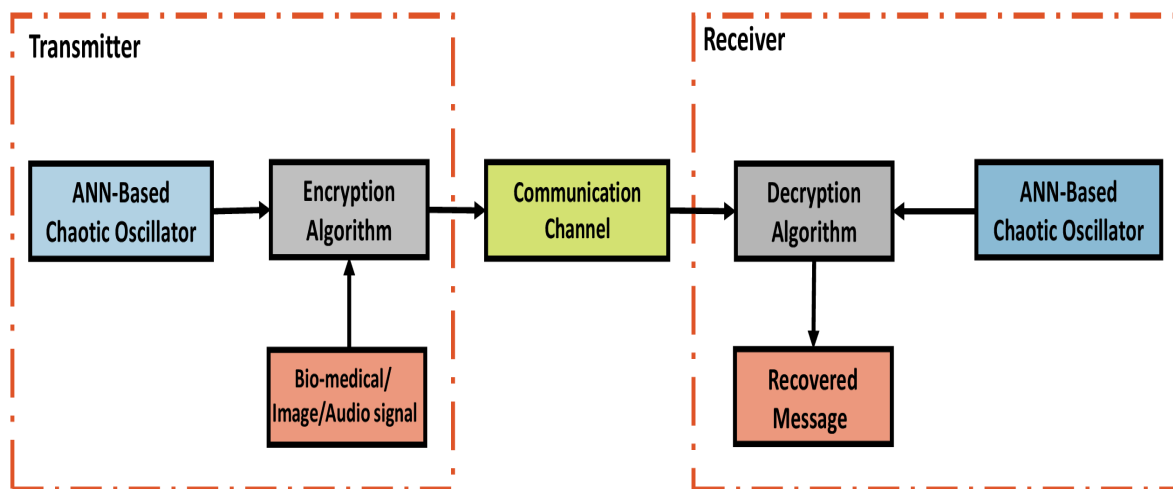


FIGURE 2. Model 2, the NeuroChaosCrypt encryption system using ANN-based chaotic oscillators.

leaving behind the original signal. To do this, the output of the ANN based jerk system is added and subtracted from signal $s_d(t)$, resulting in recovered audio signal $s_r(t)$ that matched the original signal $s(t)$.

$$s_r(t) = s_d(t) - (x_1(t) + x_2(t) + x_3(t)) \tag{5}$$

The use of ANN based jerk system guarantee that the encrypted signal can be successfully recovered on the receiver side. Hence there is no need to use any synchronization method.

1) THE ANN MODEL THAT MIMICS THE JERK CHAOTIC SYSTEM

This section is dedicated to constructing an Artificial Neural Network (ANN) model that can forecast chaotic time series.

The specific type of ANN topology utilised is the Feedforward Neural Network (FFNN). FFNN, unlike other artificial neural network topologies, operates by processing data in a unidirectional manner without any feedback connections. To put it another way, the output data $x(i)$ is predicted using the input data $x(i-1)$.

The ANN is designed to predict a jerk chaotic system consisting of three inputs, three outputs, and one hidden layer consists of 10 neurons. The Artificial Neural Network (ANN) model used to predict the jerk system shown in Figure 5. The state variables $x_1, x_2, \text{ and } x_3$ are represented by the three inputs and outputs of the model. The input layer receives the cipher key, and 10 neurons compose the hidden layer. In contrast, three neurons in the output layer showed values for the state variables $x_1, x_2, \text{ and } x_3$.

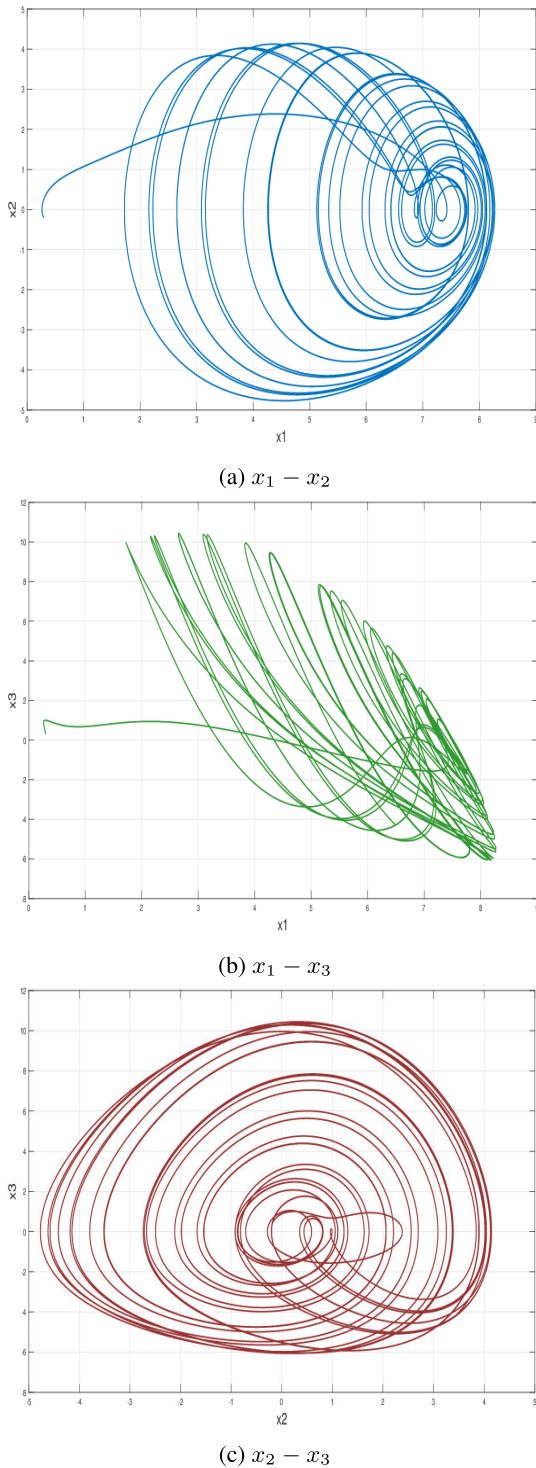


FIGURE 3. Phase attractors of the 3-D jerk chaotic oscillator.

The training process of the ANN model is done using MATLAB software. Using a dataset from the jerk system, this model is trained to forecast the initial step in the future, which consists of 10,000 samples. The model is trained using each and every sample. Next, the testing data is used to do the ANN prediction in the following manner: each output is fed back

to the input and creates the prediction data iteratively once the beginning conditions have been determined. Figures 6a, 6b, and 6c demonstrate the comparison between x original and x prediction, and x - y original and x - y prediction will be performed using the first future step prediction.

The predicted output produced by the ANN model almost matches the original outputs, as shown in Figure. However, to measure the overall performance precisely, the MSEs and RMSEs between the ANN outputs and the target outputs are used to measure the training performance effectively. Table 1 illustrates the values of MSE and RMSE for each state variable x , y , and z .

$$MSE = \frac{1}{n} \sum (x_i - \hat{x}_i)^2 \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n} \sum (x_i - \hat{x}_i)^2} \tag{7}$$

TABLE 1. Performance metrics for the ANN designs.

	x_1	x_2	x_3
MSE	$1.6917e^{-8}$	$1.3144e^{-8}$	$3.4201e^{-8}$
RMSE	0.00013	0.00011	0.00018

The waveforms obtained from NeuroChaosCrypt are illustrated in Figure 7; the original audio’s waveform is shown in Figure 7a. While Figure 7b illustrates the encrypted audio waveform. The decrypted speech waveform in Figure 7c matches the original audio waveform on the transmitter side.

C. TRANSMISSION THROUGH TRADITIONAL ENCRYPTION SYSTEM

To demonstrate the efficiency of our NeuroChaosCrypt encryption system, we will compare it with another system uses the chaotic jerk system and LQR controller for encryption and decryption. Figure 8 shows the comparative audio encryption system’s block diagram, derived from our NeuroChaosCrypt model by replacing the ANN based jerk system with a master jerk chaotic system on the transmitter side, and slave jerk system with LQR controller on the receiver side. The Overall block diagram for the Traditional Crypto System is shown in Figure 8 consisting of transmitter and receiver.

The master chaotic system’s randomized outputs are employed to mask the audio signal on the transmitter side. After the signal is masked, it may be encrypted using the encryption function, which is comparable to the encryption function found in our encryption system, NeuroChaosCrypt. With the use of the secret keys produced by the master jerk chaotic system, this function encrypts the decoy.

The encrypted signal is then subjected to the decryption process on the receiving end. This function mainly accepts the encrypted signal and the same inputs as the encryption function, which are produced by the slave jerk system. By utilizing these inputs, the decryption function reverses the

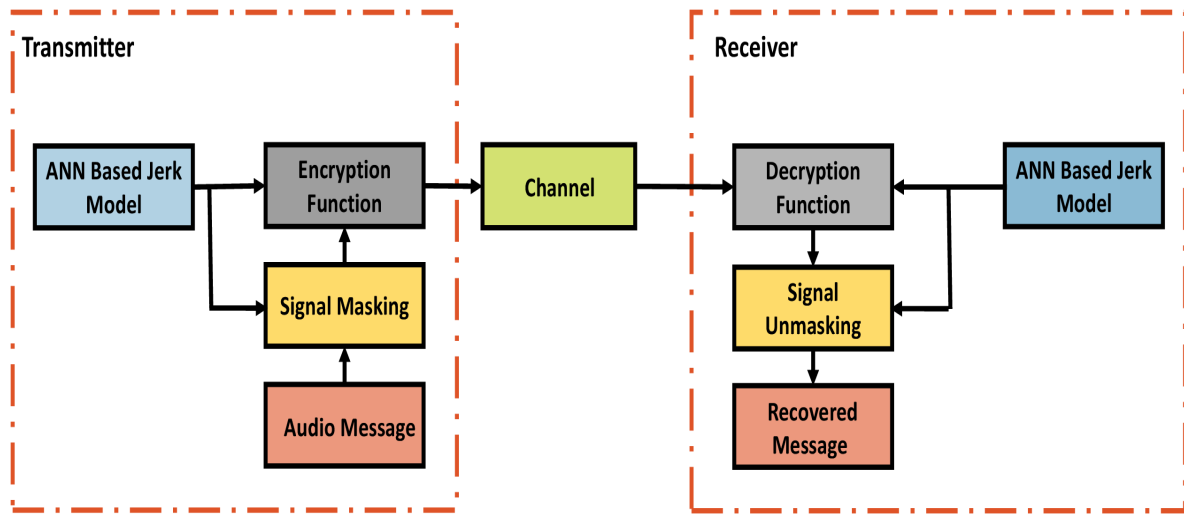


FIGURE 4. The NeuroChaosCrypt audio encryption system using an ANN-based jerk oscillator.

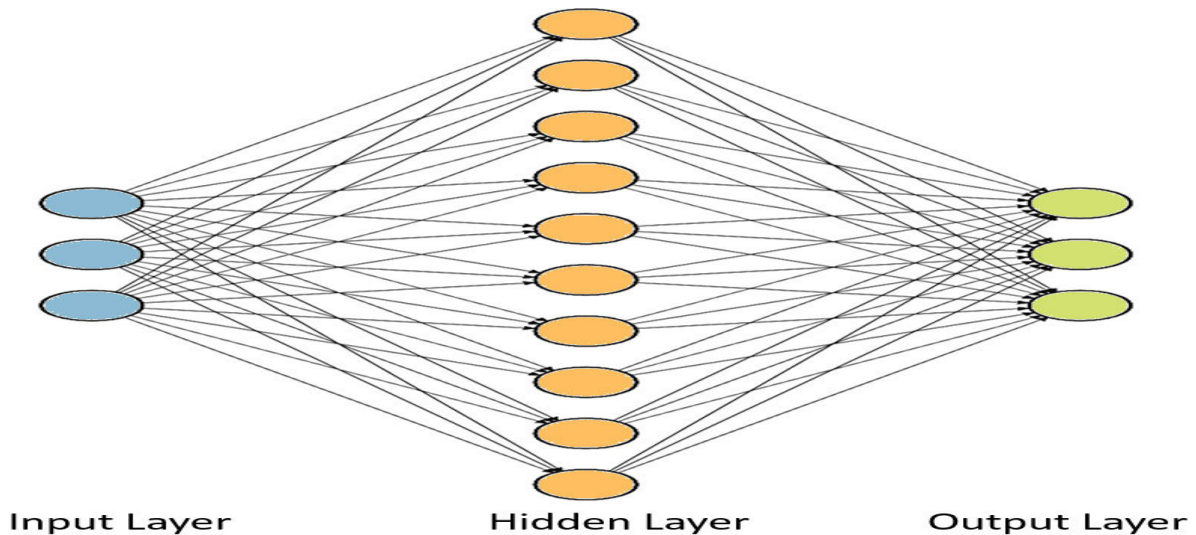


FIGURE 5. The ANN designs used for chaotic time series prediction.

encryption process and retrieves the masked signal. Then, the obtained signal is unmasked to recover the original signal where the masked signal is passed through a slave jerk chaotic system producing the recovered signal that matched the original signal.

To guarantee full matching between chaotic systems in the transmitter and the receiver, the synchronization between master and slave chaotic systems is required. In this paper, the synchronization is implemented using a Linear Quadratic Regulator (LQR) for the slave jerk system on the receiver side, as explained in the next section.

1) SYNCHRONIZATION USING LINEAR QUADRATIC REGULATOR (LQR)

The main goal of optimal control is to find the control vector $u(t)$ that is added to the dynamic equations on the slave

side. This control input shapes the behavior of the controlled system (slave oscillator) to minimize a cost function and maximize the system's output. Optimal control methodology aims to control a system most favorably, considering a cost index that includes optimization metrics. For the master jerk chaotic system described by Equations 1, the equations of the slave system with the added control law can be expressed as:

$$\begin{aligned} \dot{y}_1 &= y_2 + u_1 \\ \dot{y}_2 &= y_3 + u_2 \\ \dot{y}_3 &= ay_1 - by_2 - y_3 - cy_2y_3 - y_1^2 - y_2^2 + u_3 \end{aligned} \quad (8)$$

Here, $u = [u_1, u_2, u_3]$ represents the control vector stabilizing the slave system to follow the desired trajectory.

To derive the linear quadratic regulator, we consider a state-space representation for the nonlinear master and

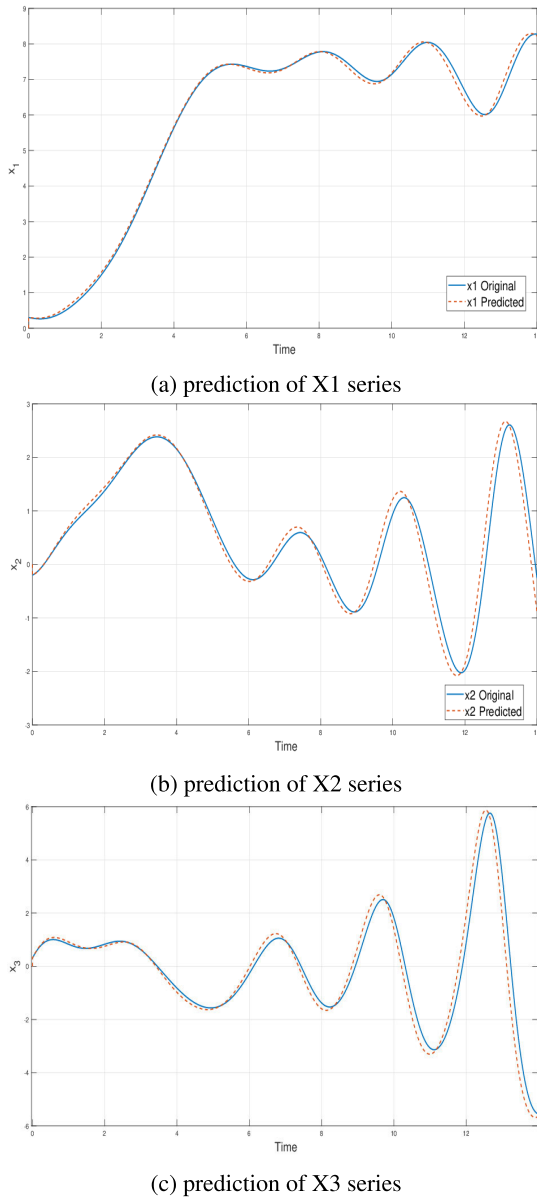


FIGURE 6. The results of training the ANN model.

slave systems:

$$\begin{aligned} \dot{x} &= Ax + g(x) \\ \dot{y} &= Ay + g(y) + u \end{aligned}$$

In these equations, x and y are state vectors in R^n , $g(\cdot)$ is the vector of continuous nonlinear functions, u is a control vector that keeps the slave system on the intended trajectory, and A is a $n \times n$ matrix that represents the linear terms. The definition of the system's error vector is:

$$\begin{aligned} e &= y - x \\ \dot{e} &= Ae + h(x, y) + u \end{aligned} \tag{9}$$

where $h(x, y) = g(y) - g(x)$. The purpose of the control law is making the error e converge to zero as t gets closer to infinity.

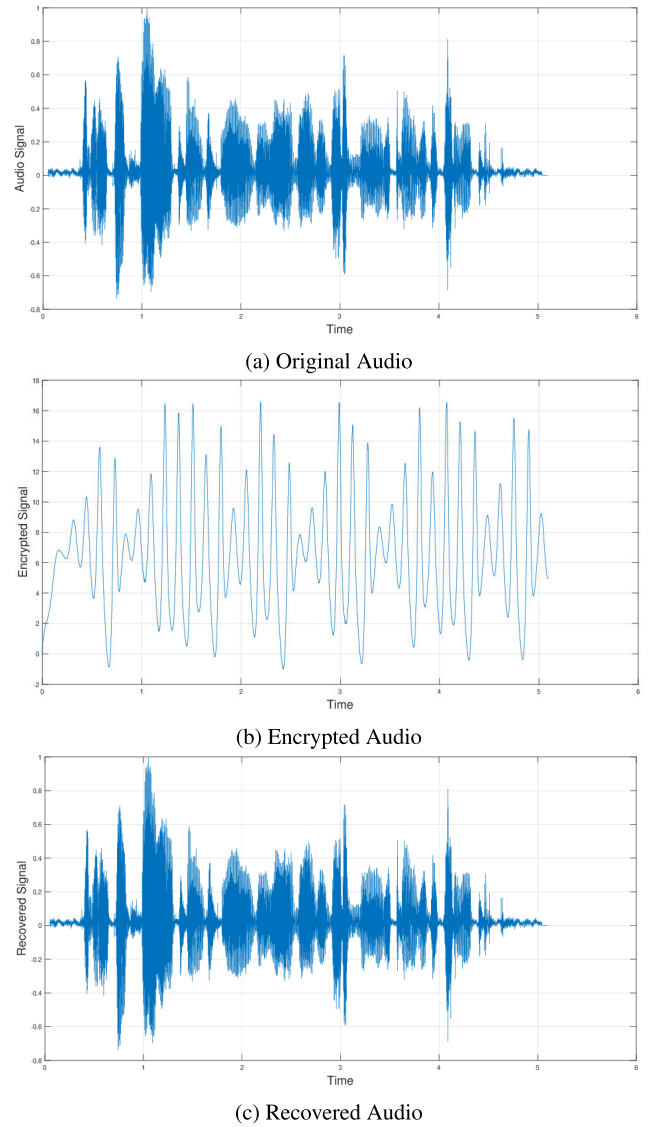


FIGURE 7. Waveform plots for the NeuroChaosCrypt encryption system: (a) Original speech, (b) Encrypted speech, and (c) decrypted speech.

where u is the sum of linear and nonlinear terms:

$$u = -h(x, y) + Bu_l \tag{10}$$

B is an $n \times m$ matrix, then substituting Equation (10) into Equation (9), we have:

$$\dot{e} = Ae + Bu_l$$

Let's assume the linear control term is given by:

$$u_l = -Ke$$

where K is an $m \times n$ linear gain matrix. In this case, the dynamic error can be written as:

$$\dot{e} = (A - BK)e$$

A linear control method called the Linear Quadratic Regulator (LQR) can be used to determine the gain matrix K .

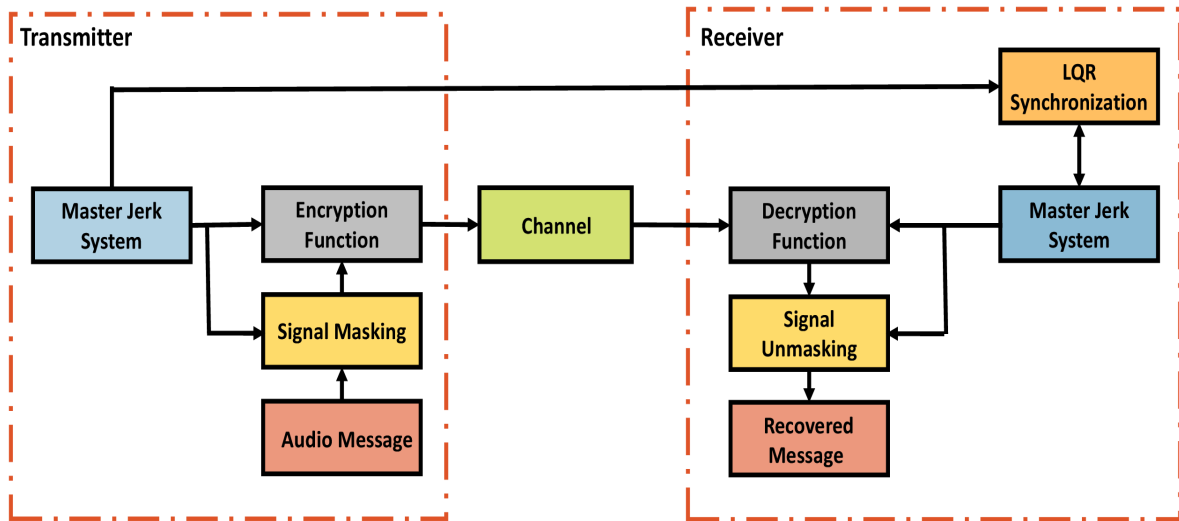


FIGURE 8. The traditional audio encryption system using the jerk system.

Considering the cost function:

$$\begin{aligned}
 J &= \int_0^\infty f(x, u) dt \\
 &= \int_0^\infty [X^T QX + u_l^T R u_l] dt \quad (11)
 \end{aligned}$$

where Q and R are positive definite matrices ensuring a positive cost function. According to optimal

Control theory, the optimal control gain is given by:

$$u_l = -R^{-1} B^T P e$$

Here, P is a positive symmetrical matrix obtained as a solution to the Algebraic Riccati Equation (ARE):

$$0 = Q + A^T P + PA - PBR^{-1} B^T P$$

For the master and slave jerk systems, the values of A, g(x), and g(y) are as follows:

$$\begin{aligned}
 A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & -b & -1 \end{bmatrix} \\
 g(x) &= \begin{bmatrix} 0 \\ 0 \\ -cx_2x_3 - x_1^2 - x_2^2 \end{bmatrix} \\
 g(y) &= \begin{bmatrix} 0 \\ 0 \\ -cy_2y_3 - y_1^2 - y_2^2 \end{bmatrix}
 \end{aligned}$$

The control matrix B is the identity matrix, $I_{3 \times 3}$. Based on these values, the Simulink block diagram for the chaotic jerk system is shown in Figures 9 and 10 for the master and slave jerk systems, respectively.

Figure 11 illustrates the Simulink block diagram of the LQR controller. Let's choose $Q = 100 \times I_{4 \times 4}$ and $R = I_{4 \times 4}$. According to the LQR method, the control law can be

obtained by solving the Riccati equation in MATLAB using the 'lqr' command to determine the matrices P and K.

$$P = K = \begin{bmatrix} 11.7608 & -0.2795 & 3.2745 \\ -0.2795 & 10.3547 & -1.1463 \\ 3.2745 & -1.1463 & 9.2559 \end{bmatrix}$$

Figure 12 shows the result of the master-slave system simulation where the initial conditions. In the Matlab/Simulink, the master and slave system initial values were set as $x_0=[0.3,-0.2,0.3]$ and $y_0=[0.2,-0.5,0.2]$. The simulation results show that our designed LQR controller with a jerk chaotic system had good synchronization performance and stability. In addition, the master-slave systems achieved synchronization quickly.

On the other hand, Figure 13 shows the waveform obtained from the Traditional Crypto System; the original audio's waveform is shown in Figure 13a. Comparatively, the encrypted audio waveform is shown in Figure 13b. Figure 13c displays the encrypted speech waveform, corresponding to the original audio waveform from the transmitter.

IV. SECURITY ANALYSIS

A. STATISTICAL TESTS

Many tests were run to show the effectiveness and security of the NeuroChaosCrypt, correlation coefficient, signal-to-noise ratio (SNR), and peak-to-random deviation (PRD) tests to evaluate and compare the security performance of the encryption system. These measurements shed light on several facets of how well encryption technologies protect the confidentiality and integrity of data. These metrics offer information on multiple aspects of how well encryption technologies maintain privacy and secrecy.

The correlation coefficient indicates the straight line that connects the original and encrypted data. A higher correlation coefficient indicates a greater similarity, potentially revealing patterns that an adversary can exploit. A lower correlation

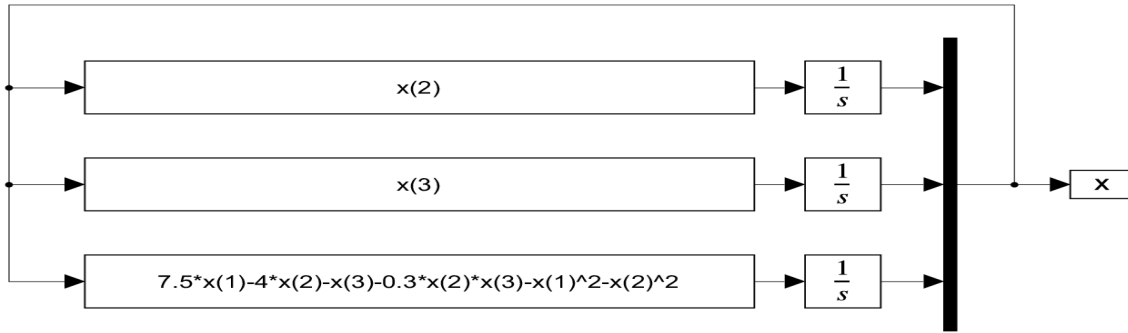


FIGURE 9. Matlab/Simulink block diagram of the master jerk system.

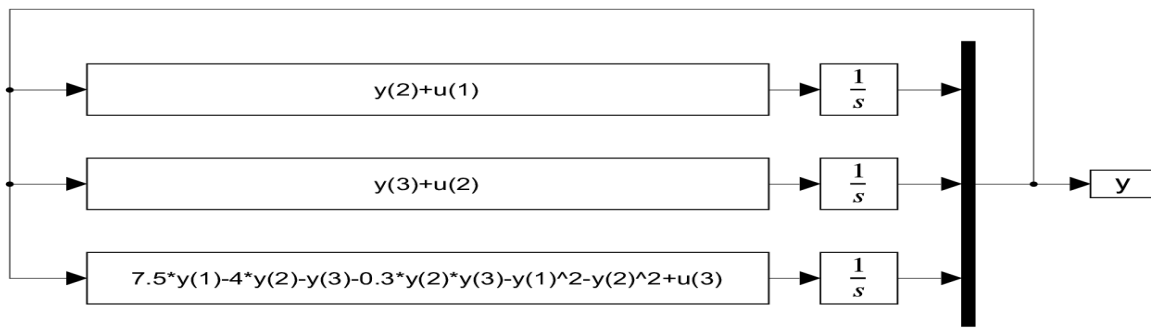


FIGURE 10. Matlab/Simulink block diagram of the slave jerk system.

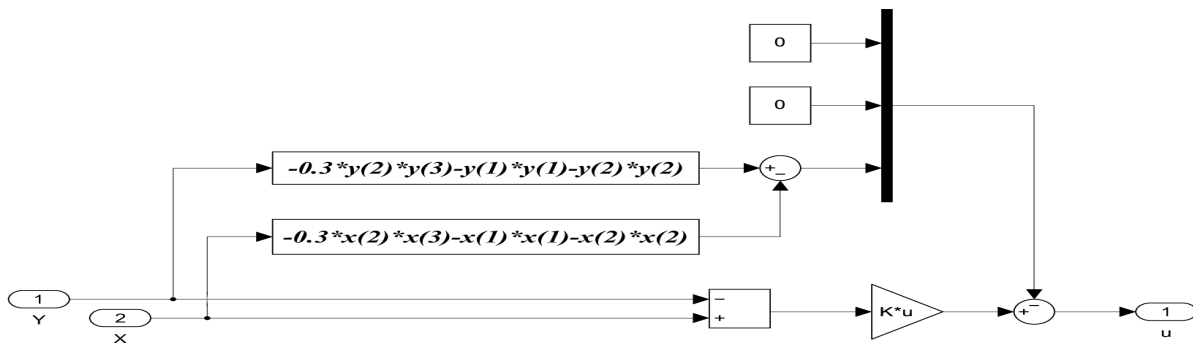


FIGURE 11. Matlab/Simulink block diagram of LQR controller.

coefficient suggests a higher level of randomness and encryption strength.

The SNR measures how strong the encrypted signal is in relation to any distortion or noise that may have been added. A higher SNR indicates a stronger and more reliable encryption system, ensuring accurate retrieval of the original information. A lower SNR may compromise the confidentiality and integrity of the encrypted data.

For the given original audio signal $x(i)$ and obtained encrypted speech $y(i)$, the SNR is defined as:

$$SNR = 10 \log_{10} \left(\frac{\sum x(i)^2}{\sum (x(i) - y(i))^2} \right) dB \quad (12)$$

PRD calculates the power ratio between the highest potential signal strength and the amount of random noise in the encrypted data. A lower PRD implies a lower level of noise and better encryption quality. A higher PRD value signifies more noise or distortion, potentially compromising data security.

$$PRD = 100 \left(\sqrt{\frac{\sum (x(i) - y(i))^2}{\sum x(i)^2}} \right) \quad (13)$$

where $x(i)$ and $y(i)$ represents the original and encrypted audio signals respectively.

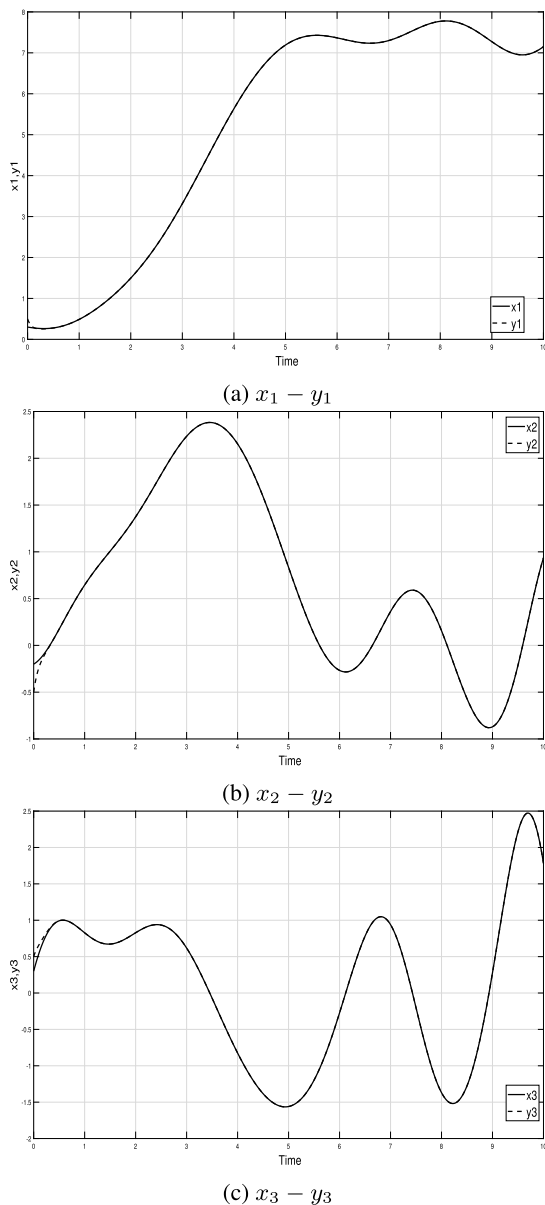


FIGURE 12. Synchronization of the mater-slave systems vary with time using LQR.

The measurement results for various performance metrics are presented in two separate tables, corresponding to each encryption system. These measurements were obtained from six different audio signals with a sampling rate of 8000 Hz and eight quantization bits.

The security performance of the NeuroChaosCrypt utilising an ANN-based jerk system and the Traditional Crypto System employing a jerk system and LQR, respectively, are shown in Tables 2 and 3. The findings indicate that the two systems have comparable efficacy in maintaining the confidentiality and integrity of data. There are similarities in the measures between the two systems, such as the correlation coefficient, SNR, and PRD. Considering the security and integrity of the encrypted data, these results show that

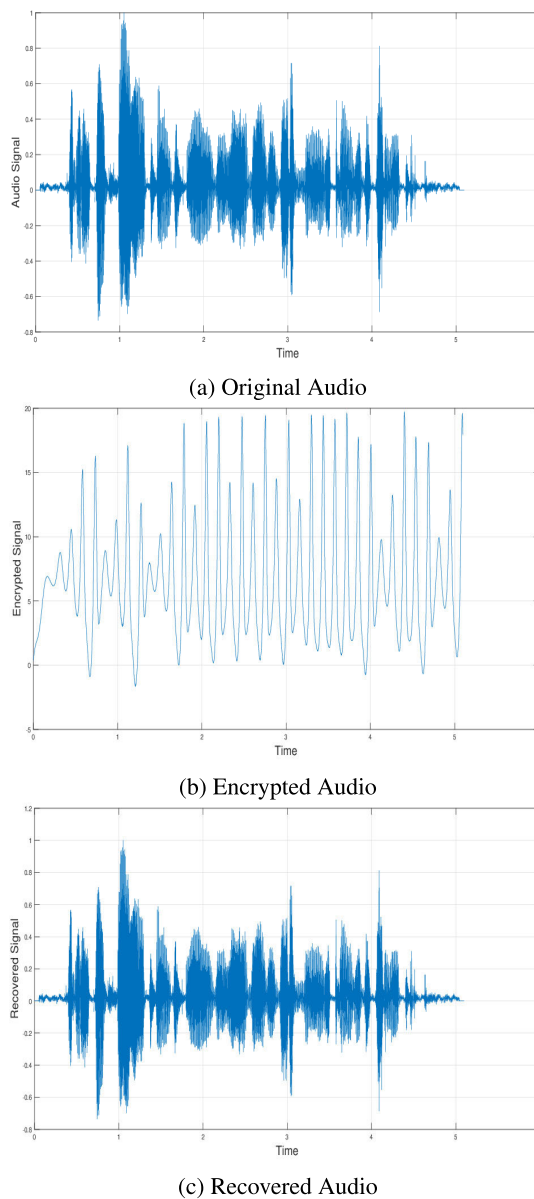


FIGURE 13. Waveform plots for the NeuroChaosCrypt encryption system: (a) Original speech, (b) Encrypted speech, and (c) decrypted speech.

TABLE 2. Comparison of security performance metrics for the NeuroChaosCrypt encryption system.

Name	CC	SNR in dB	PRD
Audio 1	0.000149	-75.2313	5.7752e+05
Audio 2	0.0031	-76.0982	6.3813e+05
Audio 3	0.0037	-74.3844	5.2387e+05
Audio 4	0.0038	-73.9910	5.0067e+05
Audio 5	0.0042	-73.6286	4.8021e+05
Audio 6	0.0049	-72.5083	4.2210e+05

the NeuroChaosCrypt encryption system and the traditional cryptosystem are equally secure.

TABLE 3. Comparison of security performance metrics for the traditional encryption system.

Name	CC	SNR in dB	PRD
Audio 1	0.0028	-75.5598	5.9978e+05
Audio 2	0.0026	-76.2820	6.5178e+05
Audio 3	0.0033	-74.3196	5.1997e+05
Audio 4	0.0032	-74.7692	5.4759e+05
Audio 5	0.0036	-74.0134	5.0196e+05
Audio 6	0.0043	-72.9010	4.4162e+05

TABLE 4. Comparison of security performance metrics for the traditional encryption system.

	Encryption Time	Decryption Time
NeuroChaosCrypt	0.0092 (sec)	0.0080 (sec)
Traditional system	0.0089 (sec)	0.0149 (sec)

B. ENCRYPTION AND DECRYPTION TIME COMPARISON

Table 4 presents a comparison of the encryption and decryption times between the NeuroChaosCrypt system and a traditional encryption system. The table illustrates how long each operation took in seconds, demonstrating the effectiveness of the NeuroChaosCrypt method.

Remarkably, the NeuroChaosCrypt system and the traditional encryption method have the exact encryption times—the former lasting around 0.0092 seconds and the latter taking about 0.0089 seconds. However, the true distinction lies in the decryption times. One notable characteristic of the NeuroChaosCrypt system is its decryption procedure, which takes only 0.0080 seconds to complete. This amazing speed is especially impressive when contrasted with the traditional encryption method, which takes about 0.0149 seconds to decipher. This significant difference can be attributed to the existing of Linear Quadratic Regulator (LQR) synchronization in the traditional system, allowing it to outperform the traditional approach regarding decryption efficiency.

C. KEY-SPACE ANALYSIS

Key-space analysis is a method used to assess the strength of a cryptographic algorithm by analyzing the possible combinations of keys that can be generated. The key-space represents the entire set of possible keys that can be used with a particular algorithm. The primary goal of key-space analysis is to determine the size and complexity of the key-space, which directly impacts the algorithm’s resistance to various attacks.

For the traditional encryption system, the jerk chaotic system consists of three parameters and three initial conditions in the proposed algorithm. All these parameters constitute the keyspace. On the other hand, for the NeuroChaosCrypt system, the ANN-based chaotic oscillator consists of 73 parameters for weight/bias. Thus, the NeuroChaosCrypt

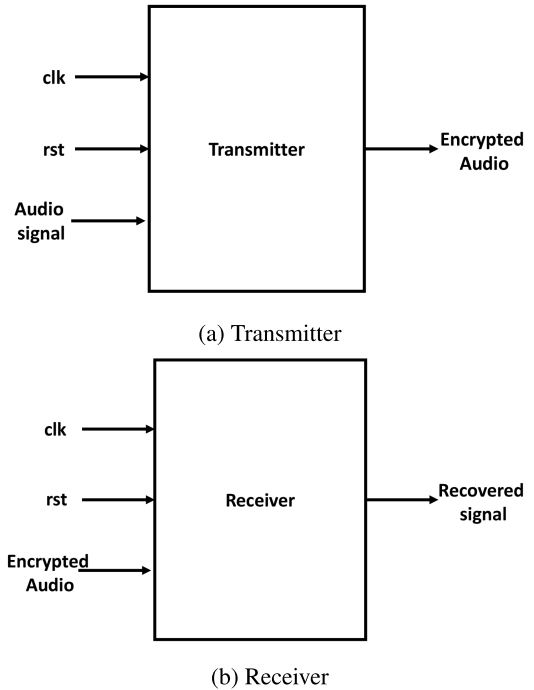


FIGURE 14. The top-level entities of the NeuroChaosCrypt encryption system.

system is robust and rigid to be attacked since it has larger keyspace than the traditional encryption system.

V. FPGA IMPLEMENTATION

FPGAs play a crucial role in implementing chaotic encryption algorithms due to their inherent characteristics, including parallel processing capabilities and real-time responsiveness. This section illustrates the hardware implementation of both studied audio encryption systems.

A. FPGA IMPLEMENTATION OF THE NEUROCHAOSCRYPT ENCRYPTION SYSTEM

Figure 14a shows the top-level entity of the transmitter consisting of three inputs; clock, reset, and audio signal, and one output representing the encrypted signal.

Similarly, the top-level block diagram for the receiver is shown in Figure 14b. The inputs of these systems are the encrypted signal, clock, and reset.

The detailed block diagrams are shown in Figures 15a and 15b for the transmitter and receiver, respectively. In those figures, the ANN model that mimics the jerk system is identical, consisting of a clock, reset, three inputs representing the current state variables $x_1[k], x_2[k], x_3[k]$, and three inputs represent the initial conditions $x_1[0], x_2[0], x_3[0]$, and three output representing the future state variables $x_1[k + 1], x_2[k + 1], x_3[k + 1]$. At the beginning, the initial conditions $x_1[0], x_2[0], x_3[0]$ define the starting values of the state variables. Then, to generate a chaotic sequence, the outputs

of the ANN model are fed back to the inputs to calculate the future values.

The encryption equations utilize the outputs of the ANN based jerk system and the audio signal to create the encrypted signal. The Encryption equations in Figure 15a, represent a set of mathematical operations that manipulate the audio signal using the generated chaotic values. These operations include signal masking [14] and then encryption using the encryption function [15], where the values generated by the ANN based jerk system are used as encryption keys, data modifiers, and parameters for the encryption function to enhance security.

$$s_m[k] = s[k] + x_1[k] + x_2[k] + x_3[k] \quad (14)$$

$$s_e[k] = (1 + x_1[k] + x_2[k] + x_3[k])^3 s_m[k] + (x_1[k]x_2[k] + x_1[k]x_3[k] + x_2[k]x_3[k]) \quad (15)$$

The Decryption Function in Figure 15b, performs mathematical operations to decrypt and reconstructs the information signal. While the specific decryption equations are not provided, this subsystem likely includes algorithms or formulas necessary to reverse the encryption process. The decryption function obtains the recovered information signal by applying the decryption equations to the outputs of the ANN-based jerk system. Using the chaotic values produced, the decryption equations represent a series of mathematical processes, which alter the audio signal. These operations include signal decryption using the decryption function in Equation 16 and signal unmasking in Equation 17.

$$s_d[k] = \frac{(y_1[k]y_2[k] + y_1[k]y_3[k] + y_2[k]y_3[k])}{((1 + y_1[k] + y_2[k] + y_3[k])^3)} \quad (16)$$

$$s_m[k] = s[k] - (y_1[k] + y_2[k] + y_3[k]) \quad (17)$$

B. FPGA IMPLEMENTATION OF THE TRADITIONAL ENCRYPTION SYSTEM

The top-level entity for the transmitter and receiver of this system is similar to the previous system in Figures 14a and 14b. The detailed block diagram of those entities are shown in Figures 16a and 16b for the transmitter and receiver respectively.

In Figure 16a, the encryption system employs a master chaotic system and encryption equations to ensure secure and robust data encryption. This system aims to exploit chaotic systems' complex and unpredictable nature to generate encryption keys and scramble the data effectively. Now let's examine the mathematical representation in more depth.

The main source of randomness and unpredictability for the encryption process is the master chaotic system, which also acts as the key generator. In order to modify the data and create encryption keys, it creates chaotic values. The mathematical representation of the master chaotic system can be described using a set of differential equations. The forward Euler integration method will be utilised for the numerical

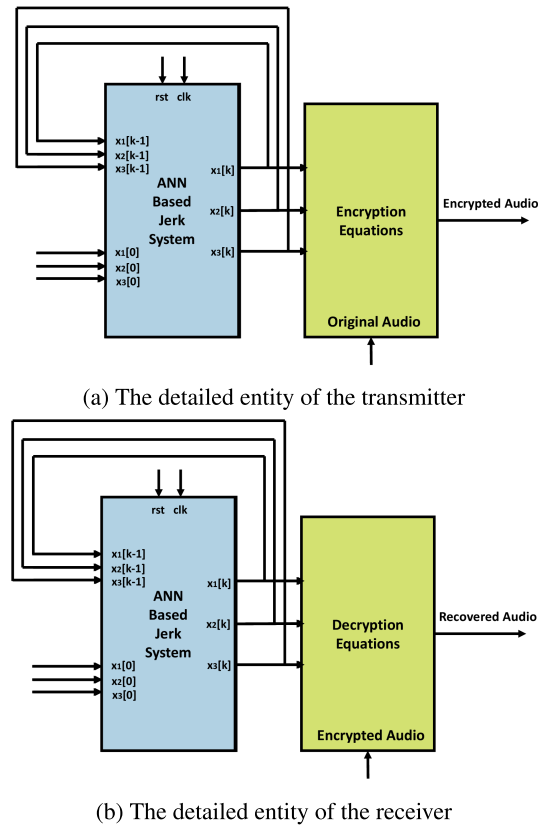


FIGURE 15. The detailed entities of the NeuroChaosCrypt encryption system.

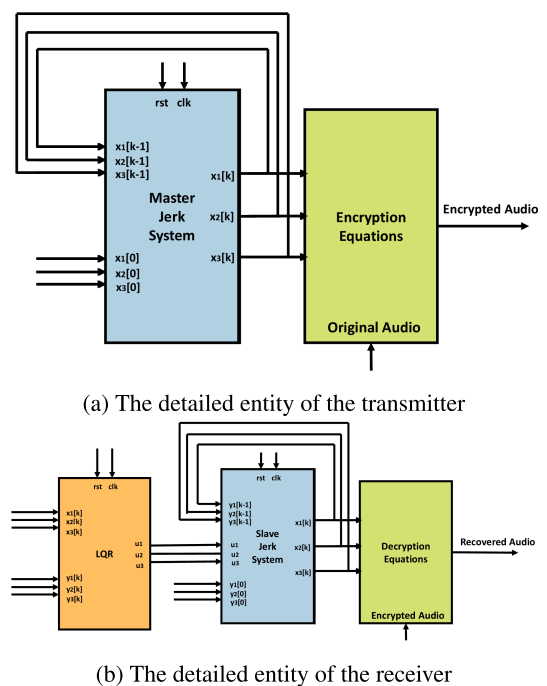


FIGURE 16. The detailed entities of the traditional encryption system.

solutions of the aforementioned system. Consequently, the numerical solution for the master chaotic oscillator system

given by Equations 1 would be as follows:

$$\begin{aligned} x_1[k+1] &= x_1[k] + dtx_2[k] \\ x_2[k+1] &= x_2[k] + dtx_3[k] \\ x_3[k+1] &= x_3[k] + dt(ax_1[k] - bx_2[k] - x_3[k] \\ &\quad - cx_2[k]x_3[k] - x_1^2[k] - x_2^2[k]) \end{aligned} \quad (18)$$

In the provided equations, k represents the current state, $k+1$ denotes the subsequent state, and dt signifies the discretization step size. The implementation of the chaotic oscillator system on a digital FPGA will utilize these discrete-time equations.

The VHDL entity that represents the master jerk system is shown in Figure 16a consisting of a clock, reset, and three inputs represent the current state variables $x_1[k]$, $x_2[k]$, $x_3[k]$, and three inputs represent the initial conditions $x_1[0]$, $x_2[0]$, $x_3[0]$, and three output representing the future state variables $x_1[k+1]$, $x_2[k+1]$, $x_3[k+1]$. Initial conditions $x_1[0]$, $x_2[0]$, $x_3[0]$ specify the initial values of the state variables at the outset. The future values are then calculated by feeding the outputs back into the inputs in a chaotic sequence.

The encryption equations perform mathematical operations to encrypt the information signal is similar to the previous system. These blocks apply Equations 14 and 15 to the inputs of the master chaotic system in order to get the mask and encrypt the information signal, respectively. The Slave Chaotic System, the Linear Quadratic Regulator (LQR), and the Decryption equations are the three interrelated subsystems that make up the decryption system seen in Figure 16b. Every subsystem contributes to the overall functioning of the code by carrying out a certain function.

The slave chaotic oscillator system represented by Equations 8 has a numerical solution that can be explained as follows:

$$\begin{aligned} y_1[k+1] &= y_1[k] + dty_2[k] \\ y_2[k+1] &= y_2[k] + dty_3[k] \\ y_3[k+1] &= y_3[k] + dt(ay_1[k] \\ &\quad - by_2[k] - y_3[k] - cy_2[k]y_3[k] \\ &\quad - y_1^2[k] - y_2^2[k]) \end{aligned} \quad (19)$$

The VHDL entity that represents the slave jerk system is shown in 16b consisting of a clock, reset, three inputs represent the current state variables $y_1[k]$, $y_2[k]$, $y_3[k]$, and three inputs represent the initial conditions $y_1[0]$, $y_2[0]$, $y_3[0]$, and three inputs representing the control signal from the LQR system for synchronization purposes. The system utilizes these inputs to predict the future state variables $x_1[k+1]$, $x_2[k+1]$, $x_3[k+1]$. At the beginning, the initial conditions $y_1[0]$, $y_2[0]$, $y_3[0]$. The outputs are fed back to the inputs to calculate the future values.

The LQR system is used for synchronization between the master and slave systems. It receives inputs from both the master system and the slave chaotic system. While the inputs from the slave chaotic system $y_1[k]$, $y_2[k]$, $y_3[k]$ offer

information on the present state of the slave system, the inputs from the master system $x_1[k]$, $x_2[k]$, $y_x[k]$ specify the intended synchronisation behaviour. Using a linear quadratic optimisation method, the LQR system processes these inputs and outputs u_1 , u_2 , u_3 , three control signals. The following is a description of the equations that control the LQR system:

$$\begin{aligned} u_1 &= K_{11}(x_1[k] - y_1[k]) \\ &\quad + K_{12}(x_2[k] - y_2[k]) + K_{13}(x_3[k] - y_3[k]) \\ u_2 &= K_{21}(x_1[k] - y_1[k]) \\ &\quad + K_{22}(x_2[k] - y_2[k]) + K_{23}(x_3[k] - y_3[k]) \\ u_3 &= K_{31}(x_1[k] - y_1[k]) \\ &\quad + K_{32}(x_2[k] - y_2[k]) + K_{33}(x_3[k] - y_3[k]) \\ &\quad - (-cx_2[k]x_3[k] - x_1[k]^2 - x_2[k]^2) \\ &\quad + (-cy_2[k]y_3[k] - y_1[k]^2 - y_2[k]^2) \end{aligned} \quad (20)$$

Here, K_{11} , K_{12} , K_{13} , K_{21} , K_{22} , K_{23} , K_{31} , K_{32} , $abdK_{33}$ are the LQR gain parameters that determine the strength of the control signals. These gain parameters are obtained through a design process to optimize the synchronization performance.

The Decryption Function performs mathematical operations to decrypt and reconstructs the information signal. While the specific decryption equations are not provided, this subsystem probably contains the formulae or methods required to undo the encryption. By applying the decryption equations to the slave chaotic system's outputs, $y_1[k]$, $y_2[k]$, and $y_3[k]$, the Decryption Function can recover the information signal. With the help of the produced chaotic values, the decryption equations signify a series of mathematical operations that modify the audio stream. Using the decryption function in Equation 16, these processes include signal decryption, and signal unmasking, which involves Equation 17.

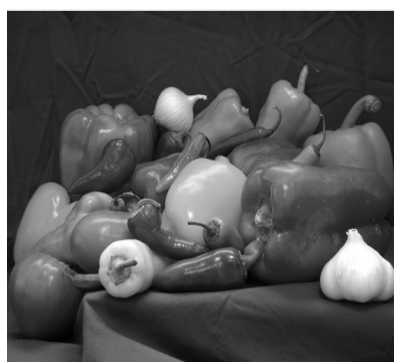
VI. OVERALL PERFORMANCE COMPARISON

In this section, we will compare two encryption systems, the NeuroChaosCrypt encryption system and the traditional encryption system, based on various aspects, including security metrics correlation coefficient, Signal-to-Noise Ratio (SNR), Peak-to-Root Mean Square Distortion (PRD), and FPGA hardware implementation considerations such as the number of utilized logic units and the maximum frequency (Fmax). Table 5 offers a thorough summary of the different characteristics that have been assessed. The values for each aspect are provided for the NeuroChaosCrypt encryption system and the standard encryption method, enabling a straightforward comparison between the two systems.

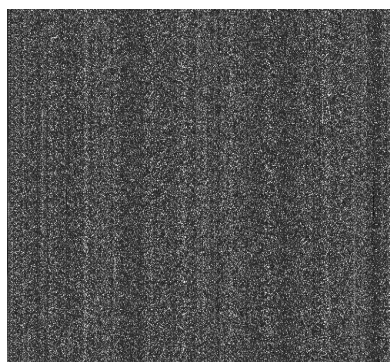
Upon examination of the results, it is evident that the security characteristics, such as correlation coefficient, SNR, PRD, and encryption time, are identical for both the NeuroChaosCrypt and traditional encryption methods. The match between the two systems demonstrates how both systems' encryption procedures provide equivalent levels of security and signal quality. The character in the user's text is a backslash.

TABLE 5. Comparison table between the systems A and B.

Metrics	Security Metrics			
	NeuroChaosCrypt encryption system		Traditional encryption system	
Correlation coefficient	0.0031		0.0026	
SNR	-76.0982		-76.2820	
PRD	6.3813e+05		9.9496e+04	
Encryption time (seconds)	0.0092		0.0089	
Decryption time (seconds)	0.0080		0.0149	
Key-space	73		6	
	FPGA Implementation			
	NeuroChaosCrypt encryption system		Traditional encryption system	
	Transmitter	Receiver	Transmitter	Receiver
Logic Units Utilization	1661	1578	1475	4541
Maximum Frequency	17.24	17.41	17.54	3.99



(a) The original image



(b) The encrypted image



(c) The decrypted image

FIGURE 17. Experimental result: (a) The original image (b) The encrypted image (c) The decrypted image.

However, the NeuroChaosCrypt system stands out with distinct advantages over current encryption techniques when comparing parameters like the time it takes to decrypt, the range of potential keys, and the hardware implementation. Because the NeuroChaosCrypt system does not require a synchronisation controller, its reduced decryption time is indicative of its efficacy. A more expansive key-space increases the intricacy of the system and fortifies its defence against attacks. Furthermore, the NeuroChaosCrypt system’s simple integration into FPGA requires fewer logic units, which has implications for cost-effectiveness, low power consumption, and efficient resource use. In addition, the NeuroChaosCrypt technology provides a higher maximum frequency (Fmax) than the prior encryption technique. The NeuroChaosCrypt system operates at a higher speed because to its higher Fmax, which enables faster encryption and data processing and transfer. The NeuroChaosCrypt system performs better because to its increased maximum frequency (Fmax), particularly when real-time encryption is required.

In summary, the NeuroChaosCrypt encryption system performs very well regarding data decoding speed because

TABLE 6. The correlation between adjacent pixels in the original, encrypted and decrypted images.

Model	Original Image	Encrypted Image	Decrypted Image
Horizontal	0.9913	0.0266	0.9927
Vertical	0.9857	0.0196	0.9863
Diagonal	0.9792	0.0272	0.9812

of its effective hardware implementation and enlarged key space, which increases complexity and security. Because of its enhanced Fmax and capacity to reduce the requirement for logic units in FPGA integration, the NeuroChaosCrypt system provides a special set of advantages. Consequently, these attributes collaborate to improve the efficiency and speed of encryption procedures, establishing the NeuroChaosCrypt system as a remarkably secure and persuasive alternative to the traditional encryption technique.

VII. APPLICATION FOR IMAGE ENCRYPTION USING THE NEUROCHAOSCRYPT SYSTEMS

In this section, we will demonstrate the application of our proposed NeuroChaosCrypt (depicted in Figure 2) for

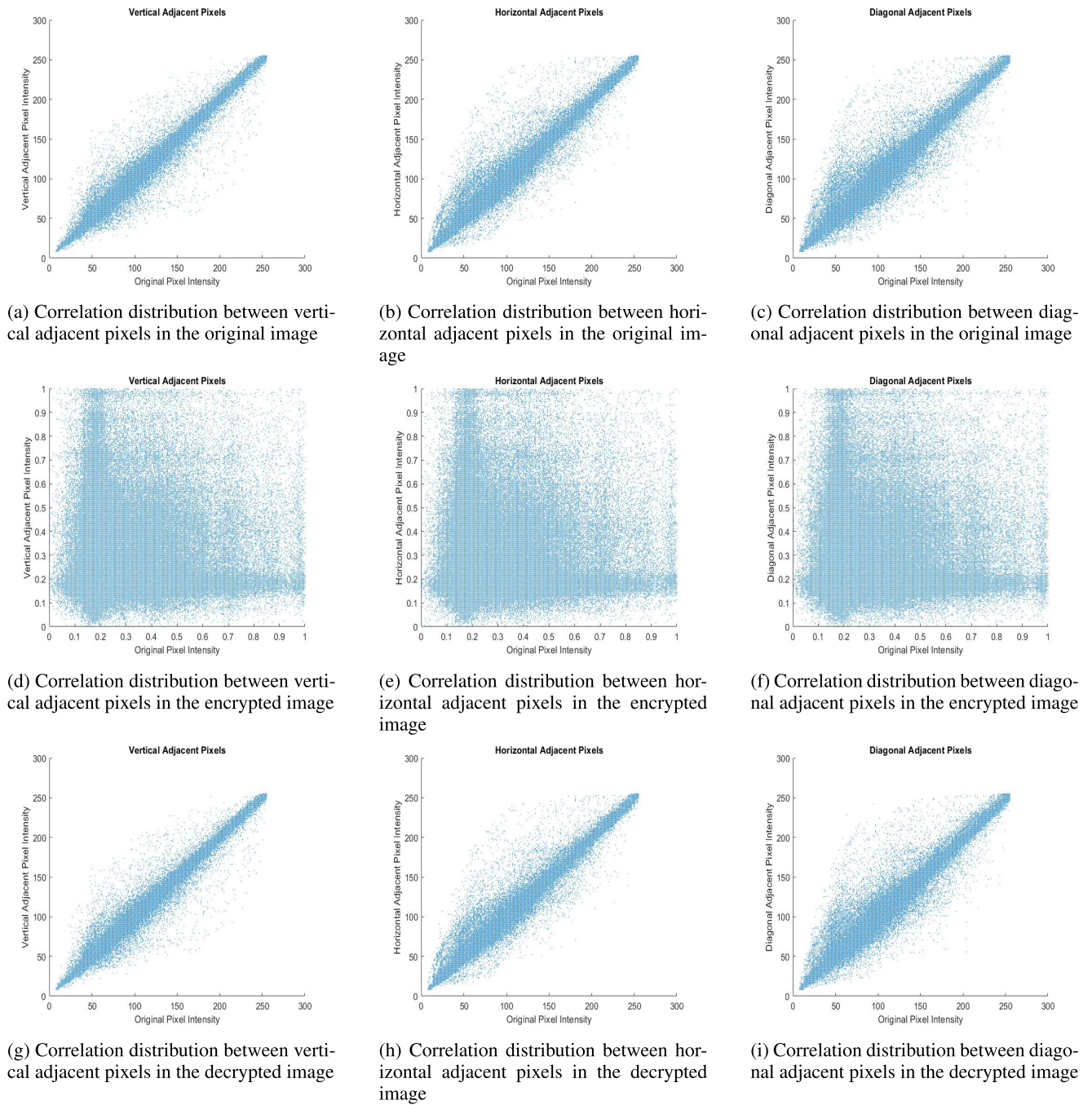


FIGURE 18. The correlation distribution between adjacent pixels in the original, encrypted and decrypted images.

image encryption and decryption. The encryption process utilizes a state vector derived from an Artificial Neural Network (ANN)-based chaotic oscillator model, denoted as $[x_1, x_2, \text{ and } x_3]$, as the encryption key.

Firstly, the encryption function employs the x_1 state variable to permute the original image. Subsequently, the permuted image undergoes diffusion using the x_2 state variable. The permuted and diffused images are then combined using XOR operations to generate the ciphered image.

On the decryption side, an identical ANN-based chaotic oscillator model is employed to generate the same state vector used during encryption for decrypting the ciphered image. The decryption process reverses the operations conducted during encryption. Initially, the ciphered image is un-permuted using the x_1 state variable, followed by un-diffusion using the x_2 state variable. Finally, the un-permuted and un-diffused images are XOR-ed to produce the decrypted image. Figure 17 illustrates the results of

implementing the Neurocryptosystem for image encryption. The encrypted image exhibits noise and deviation from the original, yet the decrypted image matches the original precisely.

A. CORRELATION COEFFICIENT

The robust encryption method applied to the encrypted image aims to reduce the correlation or likeness between neighboring pixels, even though there may be a significant correlation among adjacent pixels in the original image.

Plain images typically show a strong correlation among neighbouring pixels, whereas encrypted images are expected to show no correlation at all. In order to gain more insight, we tested both the encrypted and decrypted images and looked at the correlation between adjacent pixels. The correlation distribution between adjacent pixels in all directions for the original and encrypted images is shown in Figure 18, respectively. There is a strong correlation between adjacent pixels, as seen by the original image's linear correlation distribution. In contrast, there appears to be no discernible connection between any pixels in the encrypted image, suggesting that the pixels are distributed randomly. Therefore, we can conclude that correlation analysis poses no threat to our cryptosystem's security. Table 6 illustrates the correlation values along the diagonal, horizontal, and vertical axes between neighbouring pixels in each original image, encrypted image and decrypted image. Based on the findings presented in Table 6, it is evident that the correlation coefficient among adjacent pixels in the encrypted image is notably low, nearing zero. This observation highlights the effectiveness of our algorithm in disrupting pixel correlation, demonstrating its robustness against statistical attacks.

B. INFORMATION ENTROPY

One important metric to evaluate the degree of randomness in the value distribution of an image is information entropy. An information source is considered to provide 256 symbols, and an optimal information entropy value of 8 is found. An encrypted image with a higher information entropy is more secure against attacks and makes it more difficult for adversaries to extract important data. The information entropy value in our suggested system measures at 7.4326. These results clearly show that the encrypted image's entropy value closely approximate the ideal value of 8.

C. DIFFERENTIAL ATTACK

To evaluate the impact of altering a single pixel on the entire image encrypted through the proposed algorithm, two standard metrics were employed: NPCR and UACI. NPCR assesses the rate of change in the pixel count of the encrypted image when a single pixel in the original image is modified. On the other hand, UACI, or the unified average changing intensity, gauges the average intensity of discrepancies between the original and encrypted images. In this context, we designate the encrypted I1, and the encrypted image resulting from altering the gray value of the first pixel is

labeled as I2. Researchers commonly employ NPCR and UACI as benchmarks to evaluate the algorithm's resilience against differential attacks. Here, we utilize equations 21, 22, and 23 to compute NPCR and UACI.

$$C(i, j) = \begin{cases} 0, & \text{if } I1(i, j) = I2(i, j) \\ 1, & \text{if } I1(i, j) \neq I2(i, j) \end{cases} \quad (21)$$

$$\text{NPCR} = \frac{\sum_{i=1}^M \sum_{j=1}^N C(i, j)}{M \times N} \times 100 \quad (22)$$

$$\text{UACI} = \frac{\sum_{i=1}^M \sum_{j=1}^N |I1(i, j) - I2(i, j)|}{255 \times M \times N} \times 100 \quad (23)$$

In these equations, $C(i, j)$ represents the indicator function which evaluates to 0 if the corresponding pixels in I1 and I2 are equal, and 1 if they are not equal. M and N denote the dimensions of the image. The simulation yielded an NPCR of 99.49% and a UACI of 0.39%. These findings indicate that our algorithm exhibits robust resistance against differential attacks.

VIII. CONCLUSION

This study presents NeuroChaosCrypt, an innovative cryptographic paradigm that utilizes new approaches to guarantee safe data transport. In NeuroChaosCrypt, both transmitting and receiving ends harness an ANN-based chaotic system, eliminating the need for synchronization methods and further enhancing security.

To underscore the effectiveness of NeuroChaosCrypt, an extensive case study involving audio signal transmission is conducted. A comparative analysis between NeuroChaosCrypt and a traditional encryption system, incorporating a Linear Quadratic Regulator (LQR) controller, is carried out. The analysis showcases comparable levels of security, correlation coefficient (cc), Signal-to-Noise Ratio (SNR), Peak-to-Root Mean Square Distortion (PRD), and encryption time for both systems. Notably, the NeuroChaosCrypt, fortified by ANNs, excels in terms of decryption time, key-space, and hardware implementation using FPGA. This innovative approach reduces logic unit requirements while achieving an elevated maximum frequency.

Through this comparative exploration, valuable insights into the strengths and limitations of audio encryption systems are illuminated. With the use of this data, decision-makers will be better equipped to choose a system that meets the needs of a certain application. The excursion into the complex world of audio encryption technology shows how inventive and practical methods are always being used to improve data security.

Future research directions for NeuroChaosCrypt could involve investigating techniques to optimize the neural network architectures used in NeuroChaosCrypt to improve computational efficiency and enhance encryption/decryption performance.

REFERENCES

- [1] F. Al Mutairi and T. Bonny, "New image encryption algorithm based on switching-type chaotic oscillator," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2019, pp. 1–5.
- [2] F. AlMutairi and T. Bonny, "Image encryption based on Chua chaotic oscillator," in *Proc. 3rd Int. Conf. Signal Process. Inf. Secur. (ICSPIS)*, Nov. 2020, pp. 1–4.
- [3] G. Grassi, "Special issue editorial 'chaotic systems and nonlinear dynamics,'" *Symmetry*, vol. 14, no. 6, p. 1137, 2022.
- [4] D. Veeman, M. Mehrabbeik, H. Natiq, K. Rajagopal, S. Jafari, and I. Hussain, "A new chaotic system with coexisting attractors," *Int. J. Bifurcation Chaos*, vol. 32, no. 3, Mar. 2022, Art. no. 2230007.
- [5] H. Chaudhary, A. Khan, U. Nigar, S. Kaushik, and M. Sajid, "An effective synchronization approach to stability analysis for chaotic generalized Lotka–Volterra biological models using active and parameter identification methods," *Entropy*, vol. 24, no. 4, p. 529, Apr. 2022.
- [6] N. James, M. Menzies, and K. Chin, "Economic state classification and portfolio optimisation with application to stagflationary environments," *Chaos, Solitons Fractals*, vol. 164, Nov. 2022, Art. no. 112664.
- [7] T. Bonny, "Chaotic or hyper-chaotic oscillator? Numerical solution, circuit design, MATLAB HDL-coder implementation, VHDL code, security analysis, and FPGA realization," *Circuits, Syst., Signal Process.*, vol. 40, no. 3, pp. 1061–1088, Mar. 2021.
- [8] T. Bonny, R. Al Debsi, S. Majzoub, and A. S. Elwakil, "Hardware optimized FPGA implementations of high-speed true random bit generators based on switching-type chaotic oscillators," *Circuits, Syst., Signal Process.*, vol. 38, no. 3, pp. 1342–1359, Mar. 2019.
- [9] M. A. Mohamed, T. Bonny, A. Sambas, S. Vaidyanathan, W. Al Nassan, S. Zhang, K. Obaideen, M. Mamat, and M. K. M. Nawawi, "A speech cryptosystem using the new chaotic system with a capsule-shaped equilibrium curve," *Comput., Mater. Continua*, vol. 75, no. 3, pp. 5987–6006, 2023, doi: 10.32604/cmc.2023.035668.
- [10] T. Bonny, W. A. Nassan, S. Vaidyanathan, and A. Sambas, "Highly-secured chaos-based communication system using cascaded masking technique and adaptive synchronization," *Multimedia Tools Appl.*, vol. 82, no. 22, pp. 34229–34258, Sep. 2023.
- [11] T. Bonny, W. A. Nassan, and A. Baba, "Voice encryption using a unified hyper-chaotic system," *Multimedia Tools Appl.*, vol. 82, no. 1, pp. 1067–1085, Jan. 2023.
- [12] M. Taheri, C. Zhang, Z. R. Berardehi, Y. Chen, and M. Roohi, "No-chatter model-free sliding mode control for synchronization of chaotic fractional-order systems with application in image encryption," *Multimedia Tools Appl.*, vol. 81, no. 17, pp. 24167–24197, Jul. 2022.
- [13] A. A. K. Javan, A. Zare, R. Alizadehsani, and S. Balochian, "Robust multi-mode synchronization of chaotic fractional order systems in the presence of disturbance, time delay and uncertainty with application in secure communications," *Big Data Cognit. Comput.*, vol. 6, no. 2, p. 51, May 2022.
- [14] O. Martínez-Fuentes, J. J. Montesinos-García, and J. F. Gómez-Aguilar, "Generalized synchronization of commensurate fractional-order chaotic systems: Applications in secure information transmission," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103494.
- [15] F. Wu, G. Wang, S. Zhuang, K. Wang, A. Keimer, I. Stoica, and A. Bayen, "Composing MPC with LQR and neural network for amortized efficiency and stable control," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 2088–2101, Apr. 2024.
- [16] F. W. Alsaade, Q. Yao, S. Bekiros, M. S. Al-zahrani, A. S. Alzahrani, and H. Jahanshahi, "Chaotic attitude synchronization and anti-synchronization of master-slave satellites using a robust fixed-time adaptive controller," *Chaos, Solitons Fractals*, vol. 165, Dec. 2022, Art. no. 112883.
- [17] Q. Fu, X. Xu, and C. Xiao, "LQR chaos synchronization for a novel memristor-based hyperchaotic oscillator," *Mathematics*, vol. 11, no. 1, p. 11, Dec. 2022.
- [18] P. Alexander, S. Emiroglu, S. Kanagaraj, A. Akgul, and K. Rajagopal, "Infinite coexisting attractors in an autonomous hyperchaotic megastable oscillator and linear quadratic regulator-based control and synchronization," *Eur. Phys. J. B*, vol. 96, no. 1, p. 12, Jan. 2023.
- [19] N. Tahoun, A. Awad, and T. Bonny, "Smart assistant for blind and visually impaired people," in *Proc. 3rd Int. Conf. Adv. Artif. Intell.*, Oct. 2019, pp. 227–231.
- [20] W. A. Al-Musawi, W. A. Wali, and M. Abd Ali Al-Ibadi, "New artificial neural network design for Chua chaotic system prediction using FPGA hardware co-simulation," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 2, p. 1955, Apr. 2022.
- [21] S. Huang and T. Ma, "A fault detection method of memristor in chaotic circuit based on artificial neural network," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–10, Jul. 2022.
- [22] B. Ramadevi and K. Bingi, "Chaotic time series forecasting approaches using machine learning techniques: A review," *Symmetry*, vol. 14, no. 5, p. 955, May 2022.
- [23] M. Vaziri and H. Jahanirad, "Highly efficient implementation of chaotic systems utilizing high-level synthesis tools," in *Proc. 30th Int. Conf. Electr. Eng. (ICEE)*, May 2022, pp. 501–506.
- [24] A. Kadir, M. S. Azzaz, and R. Kaibou, "Chaos-based key generator using artificial neural networks models," in *Proc. Int. Conf. Adv. Electron., Control Commun. Syst. (ICAEECS)*, Mar. 2023, pp. 1–5.
- [25] T. Bonny and J. Henkel, "Huffman-based code compression techniques for embedded processors," *ACM Trans. Design Autom. Electron. Syst.*, vol. 15, no. 4, pp. 1–37, Sep. 2010.
- [26] T. Bonny and A. Haq, "Emulation of high-performance correlation-based quantum clustering algorithm for two-dimensional data on FPGA," *Quantum Inf. Process.*, vol. 19, no. 6, pp. 1–21, Jun. 2020.
- [27] S. M. Mohamed, W. S. Sayed, A. H. Madian, A. G. Radwan, and L. A. Said, "An encryption application and FPGA realization of a fractional memristive chaotic system," *Electronics*, vol. 12, no. 5, p. 1219, Mar. 2023.
- [28] H. Cai, J.-Y. Sun, Z.-B. Gao, and H. Zhang, "A novel multi-wing chaotic system with FPGA implementation and application in image encryption," *J. Real-Time Image Process.*, vol. 19, no. 4, pp. 775–790, Aug. 2022.
- [29] F. Yu, Z. Zhang, H. Shen, Y. Huang, S. Cai, and S. Du, "FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient," *Chin. Phys. B*, vol. 31, no. 2, Jan. 2022, Art. no. 020505.
- [30] T. Bonny, S. Vaidyanathan, A. Sambas, K. Benkouider, W. A. Nassan, and O. Naqaweh, "Multistability and bifurcation analysis of a novel 3D jerk system: Electronic circuit design, FPGA implementation, and image cryptography scheme," *IEEE Access*, vol. 11, pp. 78584–78600, 2023.



TALAL BONNY received the M.Sc. degree from the Technical University of Braunschweig, Germany, in 2002, and the Ph.D. degree from Karlsruhe Institute of Technology, Germany, in 2009. He has been an Associate Professor with the Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, and as a Faculty Member, since 2013. His current research interests include embedded systems, hardware digital design, image processing, chaotic oscillator realizations, secure communication systems, AI and machine learning, and bioinformatics. He served as a reviewer/a TPC member for many IEEE/ACM journals/conferences and was the Session Chair of the IEEE Conference on Advances in Artificial Intelligence.



WAFAA AL NASSAN received the bachelor's degree in electronic engineering from the University of Aleppo, Syria, in 2013. She is currently a Research Assistant with the University of Sharjah. Her current research interests include modeling and simulating dynamic systems, control systems, embedded systems, information security, machine vision, and intelligent robotics.