Neural Network & Network Security

***Abstract*—Traditional encryption methods rely on following general rules and using corresponding algorithms for decoding. Neural networks, on the other hand, can eliminate the dependence on these general rules and introduce concepts like random encoding and deception to achieve decoding without the need for such rules. Furthermore, there is a cryptographic system based on the combination of neural networks and AES. This system utilizes synaptic weight coefficients from neural networks to construct a diagonal matrix, with image vectors serving as inputs. Each image generates a new key, implying that the key constantly changes during the encryption process, thereby enhancing system security. The integration of these methods opens up more forward-looking and flexible approaches to encryption, contributing to improved data security.**

Keywords—Neural networks, AES, diagonal matrix, format

I. INTRODUCTION

So far, the most important automated tool used for network and communication security is encryption. The core of encryption technology lies in mapping data to a domain in a way that is not easily intercepted. The two main techniques used in encryption are symmetric encryption and asymmetric encryption. The four main principles of encryption in network security are confidentiality, authenticity, integrity, and non-repudiation.

With the development of new encryption methods, mathematics has become increasingly important. Thanks to mathematics, cryptography has reached a very high level of development, with extensive mathematical computations involved in each cipher. This means that modern encryption algorithms have higher resistance to traditional cryptographic analysis. "Traditional" cryptographic analysis considers encryption algorithms from a mathematical perspective, using algebraic properties and possible key measurements.

Side-channel attacks are a type of attack aimed at breaking the physical implementation of an algorithm. Even the most complex algorithms are ultimately implemented in processors with specific configurations. Side-channel cryptographic analysis considers parameters such as operation time, power consumption, electromagnetic radiation, sound, etc. These attacks have limited applicability as they depend on the specific device used for encryption, but they can be highly effective. Most successful attacks exploit flaws in the implementation of the primitive cryptographic algorithm.

We propose a neural network-based encryption system. We use a neural network to construct an efficient encryption system with a constantly changing key. The topology of the neural network is an important consideration, as it depends on the application for which the system is designed. Neural networks provide a powerful and versatile framework for representing nonlinear mappings from multiple input variables to multiple output variables.

The process of determining the values of these parameters based on a dataset is called learning or training, and the dataset used for this purpose is typically referred to as a training set. Neural networks can be seen as an appropriate choice of functional form for the encryption and decryption operations, based on the dataset.

II. REVIEW OF RESEARCH AND PUBLICATIONS

According to research [1], the DES (Data Encryption Standard) and AES (Advanced Encryption Standard) algorithms are susceptible to side-channel attacks. Artificial neural network technology [3] is a modern trend in effective and sustainable development of encryption protection methods.

These techniques have various architectures, learning algorithms, and flexible configuration capabilities, including for data encryption [4]. Artificial neural networks are information processing systems that exhibit certain performance characteristics similar to biological neural networks. Neural networks consist of numerous simple processing units called neurons, units, cells, or nodes. Each neuron is connected to other neurons through directional communication links with corresponding weights. Each neuron has an internal state known as activation or activity level, which is a function of the inputs it receives. Typically, neurons send their activations as signals to several other neurons. While a neuron can send only one signal at a time, the signal is broadcasted to several other neurons [3]. By properly training neural networks, they can be used as mapping functions.

Among the many available neural networks, the backpropagation neural network has been used in neural network encryption algorithms. In this case, the encryption key is the neural network itself. To decrypt the key's structural form, knowledge of all the structural features and parameter values of the neural network is required. Storing these values requires thousands or tens of thousands of bits, depending on the size and architecture of the neural network. Currently, it is

not possible to search for the parameters of a neural network through complete enumeration, so increasing the key size [5] or using a neural network committee [6-8] are the primary methods to enhance the security of encryption algorithms.

One notable feature of neural networks is their ability to represent arbitrary functions, including encryption, using multivariate variables [9]. For existing algorithms such as DES or AES, using approximated neural networks significantly improves the security against side-channel attacks for the following reasons:

- Each neuron contains a small portion of information necessary for accurate algorithm operation. In this case, a cryptanalyst needs to analyze a large number of memory units.
- Calculations are performed for each neuron regardless of the input data. Therefore, the runtime of the neural network depends on its topology and size.
- The weights of a neural network cannot determine the secret key or, in some cases, even the encryption algorithm.

Perceptron and other single-layer networks have severe limitations in their capabilities. Feedforward multi-layer networks with nonlinear node functions can overcome these limitations and can be applied in various fields. This is a subclass of acyclic networks where connections are allowed only from nodes in layer i to nodes in layer i+1.

Backpropagation neural network refers to a feedforward network trained using the backpropagation learning method. Training the network through backpropagation involves three stages: forward pass of input training patterns, calculation and backpropagation of relevant errors, and adjustment of weights. After training, the application of the network only requires the forward pass calculation stage. Even though the training process may be slow, the trained network can generate outputs very quickly.

III. THREE KEY TECHNOLOGIESS

A. *Based on Neural Network*

1) *Encryption:*

Encryption Function consists of the following three sub-functions, described in the following sections:

1. Key Generation: Firstly, the key is generated.
2. Block Processing of Input Message: The input message is divided into blocks equal to the number of keys, and each block is processed individually as input to the neural network.
3. Encryption of Each Data Block using a Generalized Regression Neural Network (GRNN): The Generalized Regression Neural Network (GRNN) is used to encrypt each data block, resulting in the encrypted message.

Input			Output							
P_3	P_2	P_1	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	1	1	0
0	1	1	0	1	0	1	0	0	1	0
1	0	0	0	0	0	1	1	0	1	1
1	0	1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	0	1	0	0	1
1	1	1	0	1	1	0	1	1	0	1

Table 1. The full pattern of training data sets

2) *Proposed:*

We propose a simple, single-parameter neural network model called the General Regression Neural Network (GRNN) for encryption and decryption. The GRNN uses standard statistical formulas to calculate the scalar random variable y, which represents the conditional mean Y of a measurement X given the vector random variable x. The vector random variable x corresponds to the input of the network, while the random variable y corresponds to the output. In addition to being used as a static regression technique, the GRNN can also be used when data statistics change over time, with a recent application being through the specification of a time constant and a threshold.

To establish a GRNN, the following steps are taken:

1. Set the number of input, pattern, and output layers (processing elements or PEs).
2. Choose the pattern units.
3. Select the time constant and reset factor.
4. Set the influence radius.

This represents a simple clustering mechanism where, if the cluster center is closest to the input vector and the distance is less than the influence radius, the input vector is assigned to that cluster. Otherwise, if possible, the input vector is assigned as the center of a new cluster.

The implementation of GRNN allows for the use of exponentially decaying σ, which is formulated as follows:

$$\sigma = \frac{S}{N^{E/M}},$$

where N is the number of pattern units, M is the number of input processing elements (PEs), and E must be between 0 and 1.

This formula generates an exponentially decaying σ that is used in the clustering mechanism to assign input vectors to clusters. The influence radius determines whether an input vector is assigned to an existing cluster or becomes the center of a new cluster. Smaller σ values result in tighter clusters and larger influence radii.

3) *Result:*

To evaluate the discussed mechanism, the encryption steps for typical numerical data are presented below. We tested the behavior of the neural network described in the previous section and found the following:

1. When using the complete pattern, the neural network operates reliably without any errors in the output.
2. When using half or other partial patterns of the complete input, the performance of the neural network is poorer. Due to the presence of

numerous errors in the output, the network fails to encrypt the input data.

3. To investigate the impact of the number of hidden units on model convergence, another test was conducted. The results of this test, as shown in Figure 2, illustrate how the errors vary with the changing number of neurons in the hidden encryption layer. The errors rapidly decrease to zero at 8 hidden neurons, indicating that the number of neurons in the hidden layer must be equal to the number of neurons in the output layer.

B. Neural Network for Randomd

1) ARTIFICIAL NEURAL NETWORKS

Neural networks are composed of a large number of simple processing elements called neurons, units, cells, or nodes. Each neuron is connected to other neurons through directed communication links with corresponding weights. Typically, a neuron sends its activation as a signal to several other neurons. It can only send one signal at a time, but that signal is broadcasted to several other neurons. By properly training a neural network, it can be used as a mapping function. Among the many available neural networks, the backpropagation neural network is commonly used.

2) Backpropagation neural nets

Perceptron and other single-layer networks have severe limitations in their capabilities. Feedforward multilayer networks with nonlinear node functions can overcome these limitations and can be applied in many fields. The term "backpropagation neural network" refers to a feedforward network trained using the backpropagation learning method. Through backpropagation, the training of the network involves three stages: forward propagation of the input training pattern, computation and backpropagation of the associated error, and adjustment of the weights. After training, the application of the network only requires the computation of the forward propagation stage. Even though the training process may be slow, a well-trained network can generate outputs very quickly.

3) Encryption

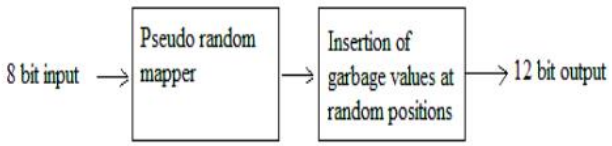

Figure 1. Block diagram of cipher

We will use a substitution cipher method. Each letter is converted to its ASCII value, while each digit is replaced with another digit. The operations performed on the digits are at the binary level. Figure 1 shows the block diagram of the encryption method used. The encryption consists of two steps:

Plain Text	ASCII Value in Binary	Pseudorandom Mapper Output	Cipher Text
n	01101110	00110111	001110101111
e	01100101	10110010	101010010100
u	01110101	10111010	101111010101
r	01110010	00111001	001011011011
a	01100001	10110000	101010000001
l	01101100	00110110	001110110101

Table 2 SAMPLE ENCRYPTION RESULTS

Step 1: Pseudorandom mapper: Consider an 8-bit binary number b7 b6 b5 b4 b3 b2 b1 b0. The number is transformed into (b0 xor b7) b7 b6 b5 b4 b3 b2 b1. This operation can be repeated multiple times if desired. However, exceeding 7 repetitions will result in the same plaintext appearing again.

Step 2: Insertion of garbage values at random positions: This ensures the randomness of the encryption method. The 8-bit ciphertext is converted into a 12-bit ciphertext. This is achieved by performing xor operations on 3 randomly chosen bits from the 8-bit output of the pseudorandom mapper to obtain 4 garbage bits. The positions where these garbage bits are inserted are also randomly chosen. This ensures a one-to-one mapping. Table 2 shows the encrypted version of the word "neural". However, any random encryption method can be used as long as it satisfies a one-to-one mapping.

4) DECRYPITON

Since the encryption is random, it requires training a neural network to achieve decryption. Using a backpropagation neural network, the 12-bit ciphertext is used as input, and the corresponding 8-bit plaintext is used as the target for training. The weights and biases of the trained neural network are provided to the recipient.

Design and operation of backpropagation net:

The designed backpropagation neural network consists of 3 layers. The first layer is the input layer, containing 12 neurons to receive the ciphertext. The second layer consists of 20 neurons. The output layer contains 8 neurons, giving 8 decoded bits (plaintext). In general, the number of neurons is proportional to the accuracy of the network. We attempt to improve the accuracy of the network by increasing the number of neurons in the middle layer (also known as the hidden layer). However, this needs to be done carefully as increasing the number of neurons increases the computational complexity and training time of the network. Taking these factors into consideration, we used 20 neurons in the middle layer.

To train the neural network, the sender uses a set of standard inputs and outputs. Provide a comprehensive ASCII character set to the encryption block and record the outputs. These outputs are then used as inputs to the neural network, and the corresponding character set is used as the target for training.

The obtained weights and biases are provided to the recipient. The recipient assigns these weights and biases to its neural network. When the encrypted message arrives at the recipient, it passes through the neural network, modified by the weights and biases, and eventually obtains the desired 8-bit decoded output. The

output, represented in bipolar form, is converted to binary form and then transformed into ASCII values, finally resulting in the actual plaintext. Thus, decryption is achieved using the neural network.

5) Analysis

If the eavesdropper successfully determines the block length and the number of garbage values, finding the positions of the garbage values would require considering 12C4 = 495 possibilities. Furthermore, for each combination, mapping each 8-bit sequence to 255 possible ASCII characters would be needed. Thus, once the block length and the number of garbage values are guessed, it would require 495 * 255 = 126,225 calculations to decode the ciphertext.

However, since the pseudorandom mapping can be performed any number of times and the positions of the garbage values are different for each ciphertext, the eavesdropper would need to perform 126,225 calculations every time they intercept a ciphertext. This makes decryption extremely tedious in the absence of a neural network.

In the scenario where the ciphertext is intercepted and decryption is attempted, there is always a possibility of being cracked. In such cases, including lies in the transmitted message can mislead the eavesdropper. To enable the recipient to distinguish between truth and lies, predetermined keywords or markers are used.

C. AES on Neural Network

The objective of this work is to build an information encryption model based on the AES algorithm and nonlinear diagonalization neural networks. The model aims to incorporate the characteristics of input images into the encryption key, providing a comprehensive approach to information encryption.

1) Architecture of Modified Nonlinear Neural

Suppose we want to use a two-layer feedforward neural network (as shown in Figure 1) to encrypt the image V. The image can be described by a deterministic vector, denoted as V = (V₁, V₂, ..., Vn), where Vi represents the projection of V onto the i-th base vector ei (ei being the base vector of the coordinate system).

To capture the prototype of the image (information signal), we impose restrictions on the synaptic connections λij (the synaptic connections from the sources V₁, V₂, ..., Vn to neurons 1, 2, 3, ..., N), where

$$\lambda_{ij} = V_i \cdot V_j, \lambda_{ij} \neq \lambda_{ji},$$

when i ≠ j. We form a matrix λ̂, where the elements λij are computed as

$$\langle \lambda_{ij} \rangle = \langle V_i \rangle \cdot \langle V_j \rangle$$

and then transform it into a diagonal form with eigenvalues λi, which are real numbers.

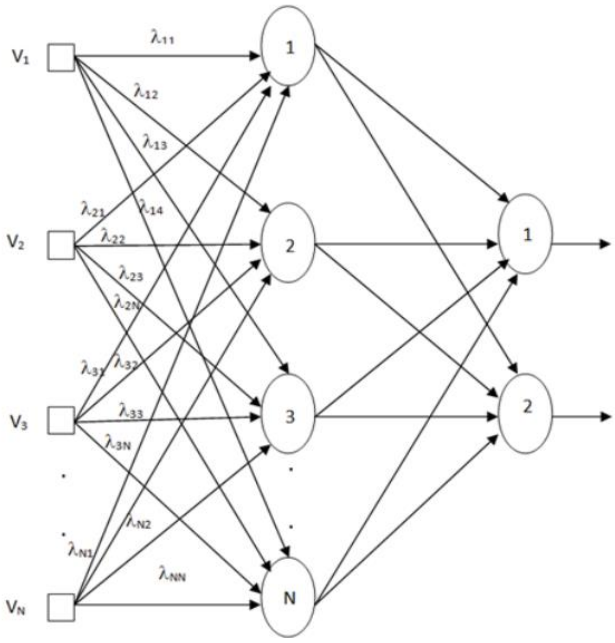

Fig. 2. Schematic representation of direct propagation of a two-layer neural network with diagonal and non-diagonal weighted coefficients of synaptic connections λ

To transform the matrix into a diagonal form, we aim to convert it into a matrix where all the off-diagonal elements are zero.

$$\hat{\tilde{\lambda}} = \hat{U}^{-1} \hat{\lambda} \hat{U},$$

where Uˆ − the matrix consists of its own base vectors

$$\hat{\lambda} \vec{u}_j = \beta_j \vec{u}_j.$$

The diagonalization operation can be used to construct asymmetric fixed-variable key encryption and simplify the weight coefficient settings of synaptic connections in neural networks. During the learning process, it eliminates some existing connections between neurons through network reconstruction and reduces the matrix element relationships between vector input signals and neurons, as well as between neurons in the hidden layer and the output layer (Figure 3).

Comparative analysis of the neural network diagrams (Figure 1, Figure 2) reveals that after diagonalizing the weight matrix of synaptic connections, the number of weights reduces from 2N to N. This change in the number of synaptic connection weights leads to a transformation of the encryption key, which becomes highly reliable, and provides a new key for the formation of each input image.

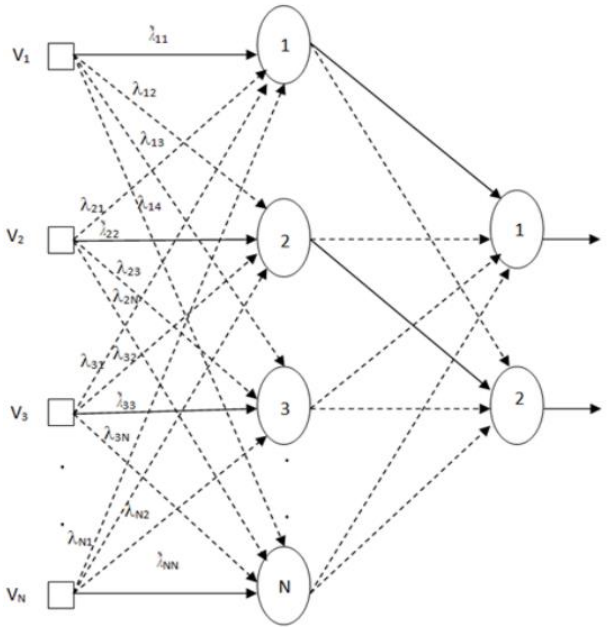

Fig. 3. Schematic representation of the two-layer neural network of direct distribution after the diagonalization of the matrix of weight coefficients of synaptic connections λij .

2) AES Encryption Systion Based on a Dlagonalized Nonliner Neural Network

The diagonalized neural network is a structure (Figure 3) composed of a set of interconnected main components, which are the synaptic connections λii between artificial neurons. The main components of the synaptic connections λii are established based on the input image vector. Each neuron has specific inputs and outputs and performs local computations or functions. The output of any neuron is determined by its input and output features, its relationships with other neurons, and external inputs. During the learning phase, the AES algorithm encrypts specific inputs (plaintext) into ciphertext. The neural network takes the input text as an input vector and the output text as a reference output vector (target) and learns to produce the same output text (target). The neural network structure model is called a "concatenated parallel model" , where the AES algorithm influences the dynamic behavior of the neural model. In the operation phase, a multi-layer feedforward neural network with limited weights is used to obtain the encrypted or decrypted output signals. In this phase, encryption or decryption continues until a new key is used in the system, and then the neural network needs to learn the new key again. In encryption and decryption, the sigmoid function is used as the nonlinear activation function for each neuron. The data range used in AES is (0-255). Therefore, during the activation phase, to make the neural network compatible with AES data, the data must be transformed to the range (0-1). This is achieved by a scaling factor of 1/256 to combine with the output of the activation function. Coefficients can be used at the output of each neuron to reverse the range transformation of the output, placing it within the interval (0-255). For simplicity, only the AES-128 version is used, which has a 128-bit key. Hence, the neural network works with data at the byte level, so that the size of both input and output is 16 bytes. The neural network must have at least one hidden layer consisting of 16 neurons to match the size of the 16-byte input key. The key structure is asymmetric and determined by the basis forming each input image vector, generating a new key for each input image.

The proposed neural cryptosystem with diagonalized weight coefficients generates an asymmetric new key for each input image. The cryptosystem consists of four parallel working layers. The input vector length for each layer of the neural network is 4 bytes. Therefore, the architecture of the diagonalized AES neural cryptosystem includes the following elements:

- The first layer has 4 neurons, with each neuron receiving one byte of the input vector (plaintext).
- The second layer has 16 neurons, with each neuron connected to all neurons in the first layer through weights.
- The third layer has 16 neurons, with each neuron influenced by the main weight components from all neurons in the second layer.
- The fourth layer has 1 neuron, influenced by the weights from all neurons in the third layer.

The analysis of the research results indicates that the cryptosystem integrating diagonalized neural networks with the AES algorithm exhibits greater resistance to attacks compared to the AES algorithm alone [29-30]. This is due to the utilization of non-linear activation functions and a key generation method that varies for each input image. Different neural network topologies can be employed, allowing encryption and decryption of longer data blocks and the use of larger asymmetric keys.

IV. CONLCUSION

A cryptographic system design based on the GRNN (Generalized Regression Neural Network) type of artificial neural network is proposed, which exhibits invariance to the secret key. The proposed neural network is tested with different training iterations, hidden layer neuron quantities, and input data. Simulation results demonstrate excellent performance, outperforming traditional encryption methods.

To make decryption more difficult for eavesdroppers, a random encryption technique is employed, incorporating pseudo-randomness and the introduction of garbage values. Neural networks are utilized as the receiving end to enhance security. In the future, there is a hope to develop special decryption techniques that can transform the decrypted message into another comprehensible message, causing confusion for eavesdroppers.

A new neural network cryptography system was developed based on the integration of the AES algorithm with neural networks. An encryption system was proposed that utilizes the diagonalization of the weight coefficients of input image vectors based on the symmetric matrix in the neural network, providing a new asymmetric key for each input image. The research findings indicate that the proposed approach of information encryption using the AES neural network with continuously changing keys enhances the security

level of cryptographic algorithms compared to existing encryption methods.

REFERENCES

[1] Specht D F 1991 IEEE Trans. Neural Networks 2 (6) 568 B. Rieder, *Engines of Order: A Mechanology of Algorithmic Techniques*. Amsterdam, Netherlands: Amsterdam Univ. Press, 2020.

[2] Schalkoff R 1999 Artificial Neural Networks, McGraw-Hill

[3] Hossein R, Anoloni M and Samee M 2004 Neural Network in Network Security, ACIT 2003 Proceeding, pp. 274– 281

[4] Stallings W 2002 Cryptography and Network Security Principles and Practice, Pearson Edition Asia

[5] Thomas Calabrese. "Information Security Intelligence: Cryptographic Principles and Applications." Thomson Delmar Learning. New York, 2006.

[6] John E. Hershey. "Cryptography Demystified." Tata McGraw-Hill. New York, 2004.

[7] I. A. Pyatnitsky, "Application of neural networks in encryption", Security of information space – 2017 : XVI all-Russian scientific and practical conference of students, postgraduates, young scientists. Yekaterinburg, 2017, pp. 44 – 46.

[8] V. Gridin, V. Solodovnikov, "Investigation of a cryptographic strngth and cryptanalysis methods for the neural network algorithm of a symmetric encryption", Journal of Ufu. Technical Science, vol. 7, 2016, pp. 114–122.

[9] A. Gozhyj, V. Vysotska, I. Yevseyeva, I. Kalinina, V. Gozhyj, "Web Resources Management Method Based on Intelligent Technologies", Advances in Intelligent Systems and Computing, vol. 871, 2019, pp. 206-221.

[10] S. Khaikin, "Neural networks. Full course", Edition. Williams, 2016.

[11] D. Kriesel, "A Brief Introduction to Neural Networks", 2007.