I.J. Intelligent Systems and Applications, **2024, 3, 1-17**
Published Online on June 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2024.03.01

Performance Evaluation of Laguerre Transform and Neural Network-based Cryptographic Techniques for Network Security

Lateef A. Akinyemi
Department of Computer Science, School of Computing, CSET, University of South Africa, Roodepoort, Johannesburg, Gauteng, South Africa
Department of Electronic &Computer Engineering, Faculty of Engineering, Lagos State University, Epe, Lagos, Nigeria
E-mail: Lateef.akinyemi@lasu.edu.ng, akinyla@unisa.ac.za
ORCID iD: https://orcid.org/0000-0003-4207-9880

Bukola H. Akinwole*
Department of Electrical/Electronic Engineering, University of Port-Harcourt, Choba, Rivers state, Nigeria
E-mail: bakinwol@gmail.com
ORCID iD: https://orcid.org/0009-0003-5673-2163
*Corresponding author

Received: 12 December 2023; Revised: 14 February 2024; Accepted: 07 April 2024; Published: 08 June 2024

Abstract: As the world evolves day by day with new technologies, there is a need to design a secure network in such a way that intruders and unauthorized persons should not have access to the network as well as information regarding the personnel in any firm. In this study, a new cryptographic technique for securing data transmission based on the LaplaceLaguerre polynomial (LLP) is developed and compared to an existing auto-associative neural network technique (AANNT).The performance of the LLPT and AANNT was tested with some selected files in a MATLAB environment and the results obtained provided comparative information (in respect of AANNT versus LLPT) as follows: encryption time (1.67 ms versus 3.9931s), decryption time (1.833 ms versus 2.1172s), throughput (26.2975 Kb/s versus 0.01098 Kb/s), memory consumption (3.349 KB versus 15.958 KB). From the compared results, it shows that AANNT offers a faster processing time, higher throughput, and takes up less memory space than the LLPT. However, cryptanalysis of the AANNT is possible if the network's weight and design are known; hence, the technique is unreliable for ensuring the data integrity and confidentiality of encrypted data. The proposed LLP cryptographic algorithm is designed to provide a higher security level by making the LLP algorithm computationally tedious to invert using the standard Laplace transform inversion method. When compared to the AANN-based cryptographic technique, cracking the algorithm to uncover the encryption key takes time. This shows the strength and robustness of the proposed LLP cryptographic algorithm against attacks, as well as its suitability for solving the problem of data privacy and security when compared to the AANN-based cryptographic algorithm.

Index Terms: Artificial Neural Network (ANN), Cryptanalysis, Cryptographic Techniques, Laguerre Transform and Laplace Transform.

1. Introduction

In the rapidly evolving landscape of digital communication and information exchange, the security of networks has become a paramount concern. The exponential growth in interconnected systems and the reliance on digital platforms highlight the critical need to establish robust defenses against potential network threats. To address these challenges and safeguards the integrity of digital ecosystems, network security emerges as a crucial framework-a set of measures and protocols designed to protect data and resources from unauthorized access, misuse, alteration, and unauthorized copying of information by hackers, corporate spies, saboteurs and frustrated employees seeking to compromise and expose network resources to different risks. Key components and practices of network security such as firewalls, intrusion detection and prevention systems, encryption, antivirus software play a crucial role in establishing a foundational layer of security but often face challenges in keeping pace with the dynamic nature of network

threats [1-2].

As network threats continue to advance in sophistication, there is an increasing need for innovative approaches to fortify network defenses. Organizations are seeking more adaptive and robust security measures that can effectively counteract emerging cyber risks. One promising avenue for research and development lies in the application of mathematical models to network security. Mathematical models, with their precision and versatility, offer a compelling foundation to address these challenges. The integration of advanced mathematical techniques into network security systems provides a robust and secure foundation for protecting sensitive data and bolstering network defences during communication over networks.

Cryptography is the mathematics underpinning the encryption technique and plays a significant role in network security by obscuring crucial data within the encrypted information from the view of any outside entity able to eavesdrop or break into the network, thereby laying the groundwork for secure communication [3-5]. The cryptography technique engages both the encryption and decryption processes to address privacy and security issues in the transmission of private and restricted data across any insecure network. A cryptographic key is an essential component of the encryption and decryption processes. Simple methods use a single key for both encryption and decryption. However, in more robust systems, a pair of keys is utilized. The second key possesses the exclusive capability to decrypt data that was initially encrypted with the first key. Cryptography as a transformation tool employs a set of algorithms and an encryption key to convert the plaintext into an unreadable ciphertext. Without disclosing the secret key to any entity, the ciphertext can be safely transmitted to the expected users. To decrypt the ciphertext into a readable format, the expected users can either use a secret key or a shared key.

Several cryptographic techniques have been employed over the years for encrypting and decrypting information. These techniques predominantly rely on mathematical principles and incorporate a range of integral transforms to enhance the security of sensitive data. The effectiveness of these techniques hinges on the complexity of the underlying algorithms. Integral transforms, including but not limited to Laplace transform [6,7], Mohand transform [8], Elzaki transform [9], Jafari transform [10], Z-transform [11], Kamal transform [12], Hermite transform [13] play a crucial role in this encryption process.

Artificial Intelligence (AI) is increasingly being used in network security to ensure safe data transmission. Several studies have explored auto-associative neural networks (AANNs) to design encryption algorithms. The AANNs are trained to autonomously transform plaintext into ciphertext independently from previous iterations. A fascinating feature of these neural networks is their ability to pre-calculate weights, which are then used to implement cryptographic protections. To enhance cryptographic data protection, AANNs can be configured with identical encryption and decryption keys or with a decryption key easily calculated from the encryption key.

This study presents a novel cryptographic technique that leverages the integration of Laguerre polynomial (LP) and Laplace Transform (LT) to provide an extra layer of security in data transmission. The effectiveness of this innovative cryptographic technique is thoroughly tested through rigorous examination. To gauge its performance, the proposed technique is compared with the auto-associative neural network (AANN) cryptographic technique. The research employs a range of metrics, such as encryption/decryption times, throughput, memory usage and security. The systematic evaluation of these parameters provides a nuanced understanding of the strengths and limitations of the new cryptographic technique in comparison to the existing AANN cryptographic method. This comprehensive analysis not only contributes to the advancement of cryptographic practices but also aids in identifying optimal solutions for securing data transmissions in diverse scenarios.

The structure of this paper is as follows: Section 1 presents the introduction. Section 2 provides a review of related works and literature. Section 3 offers a thorough evaluation of the performance of the proposed LLPT and the existing AANN technique. In Section 4, the results obtained from the proposed cryptographic technique are discussed and compared with those from existing AANN techniques using various performance metrics. Section 5 concludes the paper.

2. Theoretical Analysis

This section offers a comprehensive review of related works and relevant literature, covering topics such as auto-associative neural networks, the Laplace transformation, the Laguerre transformation, and the performance evaluation metrics used in the research study.

2.1. Related Works

The authors in [6] pioneered the utilization of Laplace transform with hyperbolic functions as a mathematical framework in a cryptographic scheme. This approach not only effectively encrypted messages into ciphertext but also enhanced both security and efficiency. In [7], a new cryptographic scheme that leverages the unique properties of Chebyshev polynomials and Laplace transforms was introduced to secure and transform data. While this study strengthens the concept presented by authors in [1], its vulnerability to attacks remains unaddressed. It was suggested that the cryptosystem parameters should be chosen accurately for maximum security. The authors in [14] introduced Jensen polynomial and Laplace transform to convert the plaintext message to cipher text message. During decryption, the process is reversed using structured mathematical techniques along with the corresponding key to recover the original message. The encryption scheme was designed to be secure against both passive and active attacks. However, it

is noteworthy that the performance of the encryption scheme was not conducted in terms of computation time and throughput.

The authors in [15] provided a thorough security analysis of encryption systems based on Laplace transform. The encryption algorithm employs a letter substitution technique generated through the Taylor series with the assistance of the Laplace transform. The study examines the cryptanalysis of the proposed algorithm detailing a comprehensive attack scenario and demonstrating how plaintext can be extracted from ciphertext without requiring knowledge of the key parameter. The proposed algorithm asserts a connection between numbers in the ciphertext and modular arithmetic. Notably, understanding the secret key is unnecessary, as the cypher is deciphered based on the principles of modular arithmetic.

In [16], a mathematical approach that employed the Kamal transform was introduced for both encryption and decryption of information. The integral transform was applied to encrypt plain text, while its inverse was utilized for decryption. Additionally, the congruence modulo operator added an extra layer of security. They demonstrated the effectiveness of the Kamal transform in cryptography and also emphasized its capacity to address issues of message insecurity posed by hackers. However, a drawback of the algorithm lies in the presence of dependencies between numbers in the ciphertext and modular arithmetic.

A novel method for encrypting and decrypting messages by combining Laplace and Elzaki transforms was devised in [17]. In this method, the message slated for encryption was treated as coefficients within the Laplace functions and applied to the polynomial function to encrypt the message. The overall security measures were significantly bolstered through a dual iteration of the cryptographic processes.

The authors in [18] introduced a cryptographic method that utilized the Aboodh transform for encrypting plaintext, along with its corresponding inverse Aboodh transform for decryption. The algorithm provided flexibility for different transformations, essential for changing keys effectively and preventing potential attacks. The adoption of Visual Basic in the implementation enhanced usability and reduced the likelihood of errors in cryptography. Furthermore, the overall security of the cryptographic method was reinforced by determining the private key through multiples of mod n.

In [19], the proposed algorithm integrated mathematical concepts, combining convolution sum and deconvolution with the Z-transform technique to enhance the security of message transmission. This increased complexity surpassed classical substitution and transposition cipher algorithms. However, the security of the system became compromised when the encryption was exposed to unauthorized users.

2.2. Auto –associative Neural Network

The auto-associative neural network (AANN) is a two-feedforward system with multiple layers of interconnected nodes connected back to back. Each node in a layer is connected to every node in the subsequent layer, and these connections have associated weights. The typical architecture of AANNs includes input, map, bottleneck, de-map and output layers, which are shown in Figure 1. This structure can be conceptualized as a fusion of two distinct network-compression and de-compression networks, converging at the bottleneck layer. The compression network operates with a known input but an unknown output, whereas the de-compression network operates in the reverse. The network transforms input data layer by layer, producing an output. During the training of AANNs, the dataset undergoes compression into a reduced set of potential variables, the quantity of which corresponds to the number of nodes in the bottleneck layers. Importantly, this count must be lower than the number of nodes in the input or output layers. Subsequently, the output from the bottleneck undergoes decompression at the de-mapping layer. AANNs'capacity to handle linear and non-linear correlations between variables is contingent on the transfer function employed in each layer. Neurons in the map and de-map layers necessitate non-linear or sigmoid transfer functions to ensure proper functionality. Conversely, nodes in the bottleneck and output layers can utilize either linear or sigmoid transfer functions. This choice is crucial for the effective operation of AANNs.

The performance of the AANNs is contingent on their ability to adeptly learn the underlying characteristics of input data during the encoding phase and subsequently reconstruct an output with identical information in the decoding phase. AANN employs an encoder featuring input and output layers, alongside a narrow hidden layer to enhance encoding and decoding. The network proficiently minimizes errors and captures important data, with the hidden layer's size tailored to its error-handling capabilities. AANN finds widespread application in various fields, including face recognition [20], pattern recognition [21], data compression [22], data validation [23], network clustering [23], fault detection [24], etc. In the training process, AANN accurately aligns a target dataset with the input training dataset at the output layer. Auto-associative neural networks are typically trained using Hebb's rule, where weights increase with the product of the input and the learning signal. Hebb's rule acts as a guiding principle, fostering connections between neurons and enhancing the network's ability to reproduce patterns from the training data. Implementing Hebb's rule Implementing Hebb's rule guarantees that AANN can accurately predict new input data as it advances through the network's architecture. AANN algorithms specifically opt for bipolar input as opposed to binary input. This choice is driven by the recognition that bipolar input offers a broader range of information representation, capturing not only the presence but also the absence of features in the input data. This nuanced representation proves advantageous for the network's ability to discern complex patterns and improve its recall accuracy. The AANN algorithm incorporates an essential security measure by utilizing a confidential secret key, known exclusively to the sender and the receiver. This cryptographic safeguard plays a pivotal role in fortifying the privacy and integrity of data transmission. However, the algorithm's reliance on a symmetric key makes it susceptible to potential key leakage.

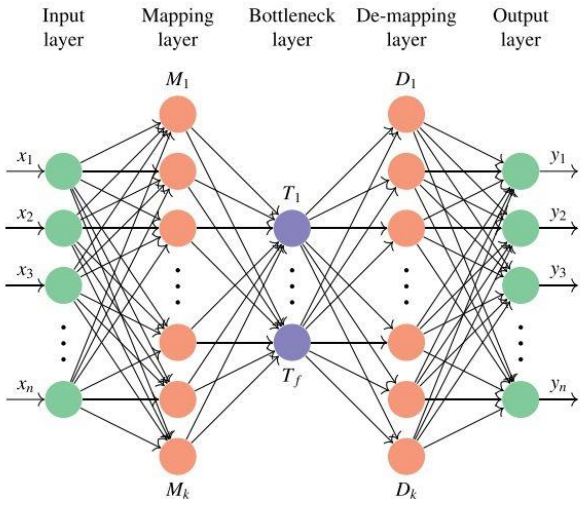

Fig.1. Auto-associative neural network (AANN) architecture

2.3. Encryption Algorithm using Auto-associative Neural Network Technique (AANNT)

The encryption key, denoted as K, is represented by a matrix with dimensions i × j, where i denotes the number of rows and j the number of columns. In this particular context, i is set to twice the value of j, where j is 6, and i is 12, resulting in a key matrix size of 12 by 6. Within this matrix, the even columns (columns 2, 4, and 6) are exclusively filled with zeros, while the odd columns can encompass various combinations of -1 and 1. The matrix K is expressed as follows, Generation of key matrix i*j where j = 6; i =12.

$$K = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \quad (1)$$

The training input vector is formed by replacing each element in the even columns of the key matrix with the bipolar equivalent of the plaintext. The AANN consists of i input neurons, j output neurons and associated weights, W_{ij}. The output matrix [Y_{ij}], can be represented as,

$$\begin{bmatrix} Y_{i1} \\ Y_{12} \\ . \\ . \\ Y_{ij} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{1i} \\ W_{21} & W_{22} & W_{2i} \\ : & . & . \\ W_{i1} & W_{i2} & W_{ii} \end{bmatrix} \begin{bmatrix} X_{i1} \\ X_{i2} \\ . \\ . \\ X_{ij} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} W_{11} & W_{12} & W_{1i} \\ W_{21} & W_{22} & W_{2i} \\ : & . & . \\ W_{i1} & W_{i2} & W_{ii} \end{bmatrix} = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ . \\ . \\ Y_{ij} \end{bmatrix}^T \begin{bmatrix} X_{i1} \\ X_{i2} \\ . \\ . \\ X_{ij} \end{bmatrix} \quad (3)$$

The resulting matrix, W_{ij} maps the output vector $[Y_i]$ to the input vector $[X_i]$, so equation (3) can be represented as,

$$[W_{ij}] = [Y_{ij}]^T * [X_{ij}] \quad (4)$$

The auto-associative network has the same input vector $[X_{ij}]$ and output vector $[Y_{ij}]$,

$$[X_{ij}] = [Y_{ij}] \quad (5)$$

The weight matrix is set by using,

$$[W_{ij}] = [X_{ij}]^T * [X_{ij}] \tag{6}$$

where, T is the learning rate given by Hebb's learning, W is the weighted matrix and X is the n-dimensional bipolar input

2.4. Decryption Algorithm Using Auto-associative Neural Network Technique (AANNT)

The sender sent both the weight matrix, $[W_{ij}]$ and the encryption key, K to the receiver. The following steps are required for decrypting the ciphertext. The weight matrix is the cipher text of the encrypted input data sent to the receiver. The ciphertext, $[W_{ij}]$ and the key matrix undergo multiplication to create matrixU [i, j]. This matrix is then subjected to the activation function of the auto-associative memory network. Through this activation, the network facilitates the retrieval of plaintext specifically from the even columns of matrixU [i, j]. Subsequently, the plaintext is converted from bipolar to binary and mapped to the corresponding letters for further processing.

$$\text{U [i, j]} = \begin{cases} +1, & if\ \text{U [i, j]} > 0 \\ -1, & if\ \text{U [i, j]} \leq 0 \end{cases} \tag{7}$$

2.5. Laplace Transformation

The Laplace transformation technique is an integral transform widely used in science and engineering [7, 25] for system modeling and analysis by simplifying the complex higher-order differential equations in the time domain into the frequency domain. The inverse Laplace transform converts the algebraic function in the frequency domain to its equivalent in the time domain. The Laplace transformation technique has a variety of properties that can be employed in securing transmitted data over the network. The Laplace transform of a function, f(t) for all real numbers t, $0 < t < \infty$ is defined by [26]:

$$\mathcal{L}_T[f(t)] = F(s) = \int_0^\infty e^{-st}\, \text{f (t) dt} \tag{8}$$

The function f (t) is a unilateral function at the t-domain and is defined for all real numbers t >0. The symbol $\mathcal{L}_T$ denotes the operator for the Laplace transform and the improper integral of f (t) will converge provided the conditions exist. The inverse Laplace transform of the function f(t), is defined as,

$$\mathcal{L}_T{}^{-1}[\text{F(s)}] = \text{f (t)} \tag{9}$$

2.6. Laguerre Transformation

The Laguerre transform, named after the French mathematician Edmond Laguerre is a valuable mathematical technique frequently employed in signal processing and quantum mechanics [27-28]. As a complex integral transform, it proves useful for analyzing linear time-invariant systems, leveraging Laguerre polynomials as fundamental basis functions. Laguerre polynomials denoted as $L_n(t)$ hold significant importance in various fields, including pure science [29], data compression [30], and numerical analysis [31], particularly in the context of finding polynomial roots. The application of Laguerre polynomial functions in data compression has demonstrated superior results compared to classical compression methods. This versatility underscores the transformative impact of the Laguerre transform across scientific and computational domains.

The Laguerre transform, $\Im$ of f(t) over the variable t, where $0 \leq t < \infty$ is defined as :

$$\Im\{f(t)\} = \int_0^\infty e^{-st}\, f(t) L_n(\text{t})\, \text{dt} \tag{10}$$

The Rodrigues formula for the Laguerre polynomial, $L_n(t)$ is given by [29]:

$$L_n(t) = \frac{e^n}{n!}\frac{d^n}{dt^n}(t^n e^{-n}) \tag{11}$$

The series representation of the Laguerre polynomial using Leibnitz's theorem is expressed as:

$$L_n\,(\text{t}) = \sum_{r=0}^{n} (-1)^r \frac{n!}{(r!)^2 (n-r)!}\, t^r \tag{12}$$

The Laguerre Polynomial, $L_n(t)$ of order n is generated using equation (13) and the results are shown in Table 1.

Table 1. Laguerre polynomial function

	$L_n(t)$
	1
	1-t
	$\frac{[t^2 - 4t + 2]}{2!}$
	$\frac{[-t^3 + 9t^2 - 18t + 6]}{3!}$
	$\frac{[t^4 - 16t^3 + 72t^2 - 96t + 24]}{4!}$
	$\frac{[-t^5 + 25t^4 - 200t^3 + 600t^2 - 600t + 120]}{5!}$
	$\frac{[t^6 - 36t^5 + 450t^4 - 2400t^3 + 5400t^2 - 4320t + 720]}{6!}$
	$\frac{[-t^7 + 49t^6 - 882t^5 + 7350t^4 - 29400t^3 + 52920t^2 - 35280t + 5040]}{7!}$
	$\frac{t^8 - 64t^7 + 1568t^6 - 18816t^5 + 117600t^4 - 376320t^3 + 564480t^2 - 322560t + 40320}{8!}$

Inverse Laguerre transform is given by:

$$\Im\{f(n)\}^{-1} = \sum_{n=0}^{\infty} \binom{n+\alpha}{n}^{-1} \frac{n!}{\Gamma(\alpha+1)} f(n) L_n(t) \tag{13}$$

where, $\alpha = 0$.

2.7. Laplace-laguerre Transformation

This study introduces an innovative cryptographic technique aimed at fortifying the security of data transmission by leveraging the Laplace-Laguerre Polynomial (LLP). The choice of LLP in this technique enhances security by introducing complexity to the encryption scheme but also underscores the versatility of employing mathematical transforms in cryptographic protocols. The intricate interplay between Laplace and Laguerre transforms adds an extra layer of sophistication to the encryption-decryption paradigm, contributing to the robustness and resilient data protection mechanism. The Laplace-Laguerre transformation is employed to transform the original message into an ambiguous format using:

$$\mathcal{L}_T[\Im\{f(t)\}] = \int_0^{\infty} \int_0^{\infty} e^{-st} f(x)\, L_n(t)\, ds\, dt \tag{14}$$

This study presents an innovative cryptographic technique designed to enhance the security of data transmission by harnessing the power of the Laplace-Laguerre Polynomial (LLP). The strategic use of LLP not only adds complexity to the encryption scheme, bolstering security but also highlights the adaptability of mathematical transforms in cryptographic protocols. The intricate interplay between Laplace and Laguerre transforms introduces an additional layer of sophistication to the encryption-decryption paradigm, thereby contributing to a robust and resilient data protection mechanism. The Laplace-Laguerre transformation is employed to convert the original message into an ambiguous format, achieving a dual objective of data obfuscation and heightened security. This transformative process adds an extra layer of intricacy to the cryptographic approach, further fortifying the overall resilience of the data protection mechanism. The transformed representations generated through the Laplace transform of the Laguerre polynomial can be decrypted into a readable version through the application of the inverse Laplace–Laguerre Polynomial. This decryption process entails acquiring the inverse Laplace-Laguerre transform of the encrypted data, followed by the simplification of the resulting simultaneous equation.

2.8. Performance Evaluation Metrics

In this study, the performance of LLP and AANN-based cryptographic techniques is evaluated based on encryption/ decryption times, memory usage, throughput and security (vulnerability to attacks). The encryption time is the duration it takes to convert plain and readable data into encrypted data. Decryption time is the time it takes to convert encrypted data back into its original and readable form. Memory usage refers to the amount of computer memory (RAM) consumed during specific operations, such as encryption and decryption. Throughput (kilobytes per second) measures the amount of data that can be processed or transmitted within a given timeframe. In the context of encryption, it would refer to how quickly data can be encrypted or decrypted. Security is a broad metric encompassing various aspects like vulnerability to attacks, resistance to unauthorized access and overall robustness of the encryption algorithms or security mechanisms.

3. Proposed Methodology

This section highlights the performance evaluation of the proposed LLPT in comparison to the existing AANN-based cryptographic technique. The evaluation aims to offer a comprehensive insight into the LLPT's efficiency and security over the existing AANN technique.

3.1. Encryption Algorithm using Laplace-Laguerre Transform Technique (LLPT)

Step 1: A plaintext "King of Men" is written in upper case with letters $M_1,\ M_2,\ldots, M_p \qquad \forall M_p$

The length of the plaintext, p = 9.

Step 2: Allocate each letter of the plaintext to its numerical equivalent, such that A= 0; B=1, then Z =25. The numerical value of each letter in the plaintext: K = 10; I= 8; N= 13; G = 6; O=14; F=5; M=12; E=4; N=13. Coefficient G_n: G_0=10; G_1 =8; G_2 =13; G_3= 6; G_4 = 14; G_5 =5; G_6 = 12; G_7 =4; G_8= 13.

The LLPT is utilized to convert the plaintext into ambiguous formats, which is later decrypted into a readable version using the inverse LLPT. Considering the message "King of Men" is to be encrypted using. The first step is to write the words in the same case letters with no spaces between each word. The first step is to put the plaintexts in the same case letter with no white space.

Step 3: Take the function of each term as,

$$\mathrm{f}(\mathrm{t}) = \sum_{n=0}^{p} t * G_n \tag{15}$$

Step 4: Take the Laplace-Laguerre Polynomial transform of both sides using the equation:

$$\mathcal{L}_{\mathrm{T}}[\Im\{\mathrm{f}(\mathrm{t})\}] = \int_0^\infty \int_0^\infty \mathrm{e}^{-\mathrm{st}} t * G_n * \mathrm{L_n}(\mathrm{t})\ \mathrm{ds}\ \mathrm{dt} \tag{16}$$

$$= \left\{ \begin{array}{c} \left(18t + \frac{26t}{2!} + \frac{36t}{3!} + \frac{600t}{5!} + \frac{336t}{4!} + \frac{8640t}{6!} + \frac{20160t}{7!} + \frac{524160t}{8!}\right) + \\ \left(-8t^2 - \frac{52t^2}{2!} - \frac{114t^2}{3!} - \frac{1344t^2}{4!} - \frac{3000t^2}{5!} - \frac{51840t^2}{6!} - \frac{141120t^2}{7!} - \frac{4193280t^2}{8!}\right) + \\ \left(\frac{13t^3}{2!} + \frac{54t^3}{3!} + \frac{1008t^3}{4!} + \frac{3000t^3}{5!} + \frac{64800t^3}{6!} + \frac{211680t^3}{7!} + \frac{7338240t^3}{8!}\right) + \\ \left(-\frac{6t^4}{3!} - \frac{224t^4}{4!} - \frac{10000t^4}{5!} - \frac{28800t^4}{6!} - \frac{117600t^4}{7!} - \frac{4892160t^4}{8!}\right) + \\ \left(\frac{14t^5}{4!} - \frac{5400t^5}{6!} + \frac{29400t^5}{7!} + \frac{1528800t^5}{8!} + \frac{125t^5}{5!}\right) + \\ \left(\frac{5t^6}{5!} - \frac{432t^6}{6!} + -\frac{3528t^6}{7!} - \frac{244608t^6}{8!}\right) \\ + \left(\frac{12t^7}{6!} + \frac{196t^7}{7!}\frac{20384t^7}{8!}\right) \\ + \left(-\frac{4t^8}{7!} - \frac{832t^8}{8!}\right) \\ + \frac{13t^9}{8!} \end{array} \right\} \tag{17}$$

Step 5: Adjust the resultant values from step 4, using modulo-26: $\mathrm{F_n} \bmod 26 = h_n$. The resultant values 85 -674 2379-4496 4945 -5274 2828 900 1177 are adjusted using modulo-26. By considering, the encryption algorithm, F_n mod 6, the ciphertext can be generated. Therefore, F_n mod 26 = h_n,

$$\begin{aligned} \mathrm{h_o} &= 85 \bmod 26 = 7 \\ \mathrm{h_1} &= -674 \bmod 26 = 2 \\ \mathrm{h_2} &= 2379 \bmod 26 = 13 \\ \mathrm{h_3} &= -4496 \bmod 26 = 2 \\ \mathrm{h_4} &= 4945 \bmod 26 = 5 \\ \mathrm{h_5} &= -5274 \bmod 26 = 4 \\ \mathrm{h_6} &= 2828 \bmod 26 = 20 \\ \mathrm{h_7} &= 800 \bmod 26 = 20 \\ \mathrm{h_8} &= 117 \bmod 26 = 13 \end{aligned}$$

Step 6: The resulting values h_n are converted to alphabetical characters to form the cipher text: The cipher text is HCNCFEUUN

Step 7: Generate the encryption key using,

$$k_n = {}^{1}/_{26}\,(F_n - h_n) \quad \text{where, } 0 \le n \le p. \tag{18}$$

$$k_n = {}^{1}/_{26}\,(F_n - h_n) \quad \text{where, } 0 \le n \le p;\ k_o = 3;\ k_1 = -26;\ k_2 = 91;\ k_3 = -173;\ k_4 = 1$$

Hence, the message KINGOFMEN is encrypted as HCNCFEUUN and sent to the receiver with keys 3, -26, 91, -173, 190, -203, 108, 30 and 4.

3.2. Decryption Algorithm using Laplace-laguerre Transform Technique

The message KINGOFMEN is encrypted as HCNCFEUUN and sent to the receiver with encryption keys 3, -26, 91, -173, 190, -203, 108, 30 and 4.

The following steps are taken to decrypt the cipher text:

Step 8: Transform the cipher text to its corresponding numerical equivalent: H=7; C= 2; N= 13; C= 2; F 5;

$$E = 4;\ = 20\ ;\ U = 20;\ N = 13.$$

Step 9: With the given key, generate the coefficients of the polynomial such that,

$$F_n = 26 * k_n + C_n \tag{19}$$

$$\left.\begin{aligned}
F_0 &= 26 * k_n + C_0 = 26 * 3 + 7 = 85\\
F_1 &= 26 * k_n + C_1 = 26 * -26 + 2 = -674\\
F_2 &= 26 * k_n + C_2 = 26 * 91 + 13 = 2379\\
F_3 &= 26 * k_n + C_3 = 26 * -173 + 2 = -4496\\
F_4 &= 26 * k_n + C_4 = 26 * 190 + 5 = 4945\\
F_5 &= 26 * k_n + C_5 = 26 * -203 + 4 = -5274\\
F_6 &= 26 * k_n + C_6 = 26 * 108 + 20 = 2828\\
F_7 &= 26 * k_n + C_7 = 26 * 30 + 20 = 800\\
F_8 &= 26 * k_n + C_8 = 26 * 4 + 13 = 117
\end{aligned}\right\}$$

Step 10: Evaluate the inverse Laplace-Laguerre transform of each term: take the inverse Laguerre of each term using equation (18) and subsequently, find the inverse Laplace of the resulting terms.

$$\Im\{f(n)\}^{-1} = \sum_{n=0}^{\infty} \binom{n+a}{n}^{-1} \frac{n!}{\Gamma(\alpha+1)} f(n) L_n(t) \tag{20}$$

$$\begin{aligned}
\{t*G_0 * [1] + t*G_1[1-t] + t*G_2\left[\frac{[t^2-4t+2]}{2!}\right] + t*G_3\left[\frac{[-t^3+9t^2-18t+6]}{3!}\right] + t*G_4\left[\frac{[t^4-16t^3+72t^2-96t+24]}{4!}\right] + \\
t*G_5\left[\frac{[-t^5+25t^4-200t^3+600t^2-600t+120]}{5!}\right] + t*G_6\left[\frac{[t^6-36t^5+450t^4-2400t^3+5400t^2-4320t+720]}{6!}\right] + t*G_7 * \\
\left[\frac{[-t^7+49t^6-882t^5+7350t^4-29400t^3+52920t^2-35280t+5040]}{7!}\right] + \\
t*G_8\left[\frac{t^8-64t^7+1568t^6-18816t^5+117600t^4-376320t^3+564480t^2-322560t+40320}{8!}\right]\}
\end{aligned} \tag{21}$$

Step 11: Compare the coefficient of each term in step 9 and step 10.

$$\begin{aligned}
85 &= G_0 + G_1 + G_2 + G_3 + G_4 + G_5 + G_6 + G_7 + G_8\\
-674 &= -2G_1 - 4G_2 - 6G_3 - 8G_4 - 10G_5 - 12G_6 - 14G_7 - 16G_8\\
2379 &= 3G_2 + 9G_3 + 18G_4 + 30G_5 + 45G_6 + 63G_7 + 84G_8\\
-4496 &= -4G_3 - 16G_4 - 40G_5 - 80G_6 - 140G_7 - 224G_8\\
4945 &= 5G_4 + 25G_5 + 75G_6 + 175G_7 + 350G_8\\
-5274 &= -6G_5 - 36G_6 - 126G_7 - 336G_8\\
2828 &= 7G_6 + 49G_7 + 196G_8\\
800 &= -8G_7 - 64G_8\\
117 &= 9G_8
\end{aligned}$$

Step 12: Solve the resulting simultaneous equations to obtain the plaintexts: G_o = 10; G_1= 8; G_2 = 13; G_3 = 6; G4 = 14;

G_5 = 5; G_6 = 12; G_7 = 4 =; G_8 = 13. The ciphertext HCNCFEUUN is decrypted to the original message as KINGOFMEN, therefore the original message of the sender can be written as King of Men.

3.3. Encryption Algorithm using Auto- associative Neural Network Technique (AANNT)

Step 1: Change the plaintext to its binary equivalent.

Plaintext	K	I	N	G	O	F	M	E	N
Binary	1010	1000	1101	0110	1110	0101	1100	0100	1101

To encrypt and decrypt using auto-associative neural network (AANNT), training involves using both keys and plaintext data. This involves the processing of the plaintext, key generation and weights computation. The plaintext of 36 bits long is set as,

1	0	1	0	1	0	0	0	1	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1

Step 2: Generate the key matrix K of size I*J such that all elements of the even columns are 0. I is the row element of matrix K and J is the column element of the matrix K. I is given as 2*J. Generation of the key matrix: J = 6; I = 12; the size of the key matrix is 12*6.

$$K = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \quad (22)$$

Step 3: Generate the training vector by replacing each element of the even column of the key matrix with its binary equivalent. The even columns are columns 2, 4 and 6. The training input data, X can be represented as:

$$X = \begin{bmatrix} 1 & 1 & -1 & 0 & 1 & 1 \\ -1 & 0 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & -1 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 1 & 1 & 0 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 1 & -1 & 1 \end{bmatrix} \quad (23)$$

Step 4: Translate the plaintext in step 3 from binary to bipolar format. The plaintext in the input vector is transformed into its bipolar equivalent:

$$X' = \begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix} \quad (24)$$

Step 5: Calculate the weight matrix and ciphertext matrix. The weight matrix, W_{ij} is the ciphertext and is expressed as,

$$W_{ij} = \sum_{p=1}^{n} X' * X'^{T} \tag{25}$$

$$\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 12 & 0 & -6 & -2 & -2 & 0 \\ 0 & 12 & 2 & 2 & 6 & 4 \\ -6 & 2 & 12 & 0 & 4 & 6 \\ -2 & 2 & 0 & 12 & 0 & 2 \\ -2 & 6 & 4 & 0 & 12 & 6 \\ 0 & 4 & 6 & 2 & 6 & 12 \end{bmatrix} \tag{26}$$

3.4. Decryption Algorithm using Auto-associative Neural Network Technique (AANNT)

The sender and the receiver share the same private key during the cryptographic process. The weighted matrix which is the same as the cipher text is decrypted using the same key matrix to recover the original plaintext. After performing the training process, the network is tested by applying an activation function to confirm that the generated output vector is the same as the input.

Step 6: Multiply Matrix, W_{ij} with the key matrix, K to generate a matrixU_{ij} :

$$U_{ij} = \mathrm{K}^{*}\, W_{ij} \tag{27}$$

$$= \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} * \begin{bmatrix} 12 & 0 & -6 & -2 & -2 & 0 \\ 0 & 12 & 2 & 2 & 6 & 4 \\ -6 & 2 & 12 & 0 & 4 & 6 \\ -2 & 2 & 0 & 12 & 0 & 2 \\ -2 & 6 & 4 & 0 & 12 & 6 \\ 0 & 4 & 6 & 2 & 6 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 18 & 18 & -6 & -10 & 18 & 14 \\ -18 & -10 & 18 & 14 & -6 & 10 \\ 18 & 2 & -26 & 10 & -18 & -18 \\ -2 & -26 & -18 & -14 & -26 & -30 \\ -10 & 14 & -6 & 14 & 10 & -6 \\ 14 & -2 & -10 & 10 & 6 & 10 \\ -6 & -22 & -18 & 10 & -26 & -26 \\ 22 & -26 & -30 & -18 & -30 & -30 \\ -18 & 22 & 30 & -6 & 30 & 26 \\ -22 & 26 & 30 & 18 & 30 & 30 \\ 22 & -26 & -30 & -18 & -30 & -30 \\ 18 & 10 & -14 & 14 & -6 & 6 \end{bmatrix}$$

Step 7: Apply the activation rule on matrix U to generate the plaintext. The activation function,

$$\mathrm{f}\,(U_{ij}) = \begin{cases} +1, & U_{ij} > 0 \\ -1, & U_{ij} \leq 0 \end{cases} \tag{28}$$

$$= \begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \quad (29)$$

Step 8: Retrieve the plaintext at each even column from step 7. The original message is retrieved at each even column as:

1	-1	1	-1	1	-1	-1	-1	1	1	-1	1	-1	1	1	-1	1	1	1	-1	-1	1	-1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1

Step 9: Convert the plaintext in Step 8 from the bipolar to its binary equivalent. The plaintext in binary format is obtained as:

1	0	1	0	1	0	0	0	1	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1

Step 10: Convert the binary equivalent in Step 9 to its corresponding letter. Finally, the plaintext of the binary 1010 1000 1101 0110 1110 0101 1100 0100 1101 is decrypted as 'KINGOFMEN'.

The flowchart diagrams of the encryption and decryption process for the LLPT and AANNT are given in Figures 2 and 3, respectively.

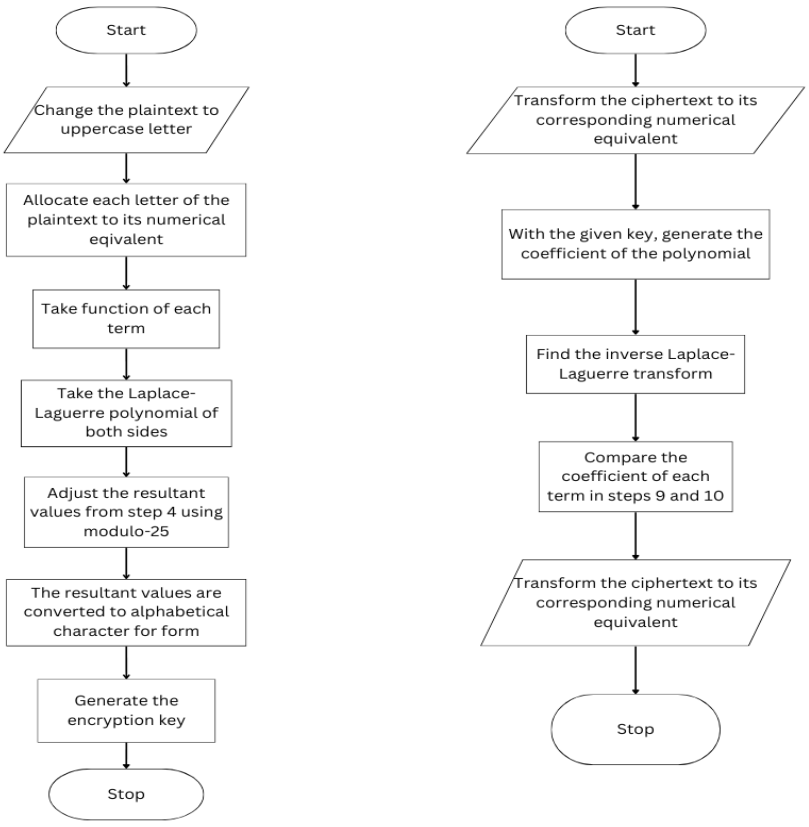

Fig.2. Flowchart diagram of LLP Technique (a) encryption (b) decryption

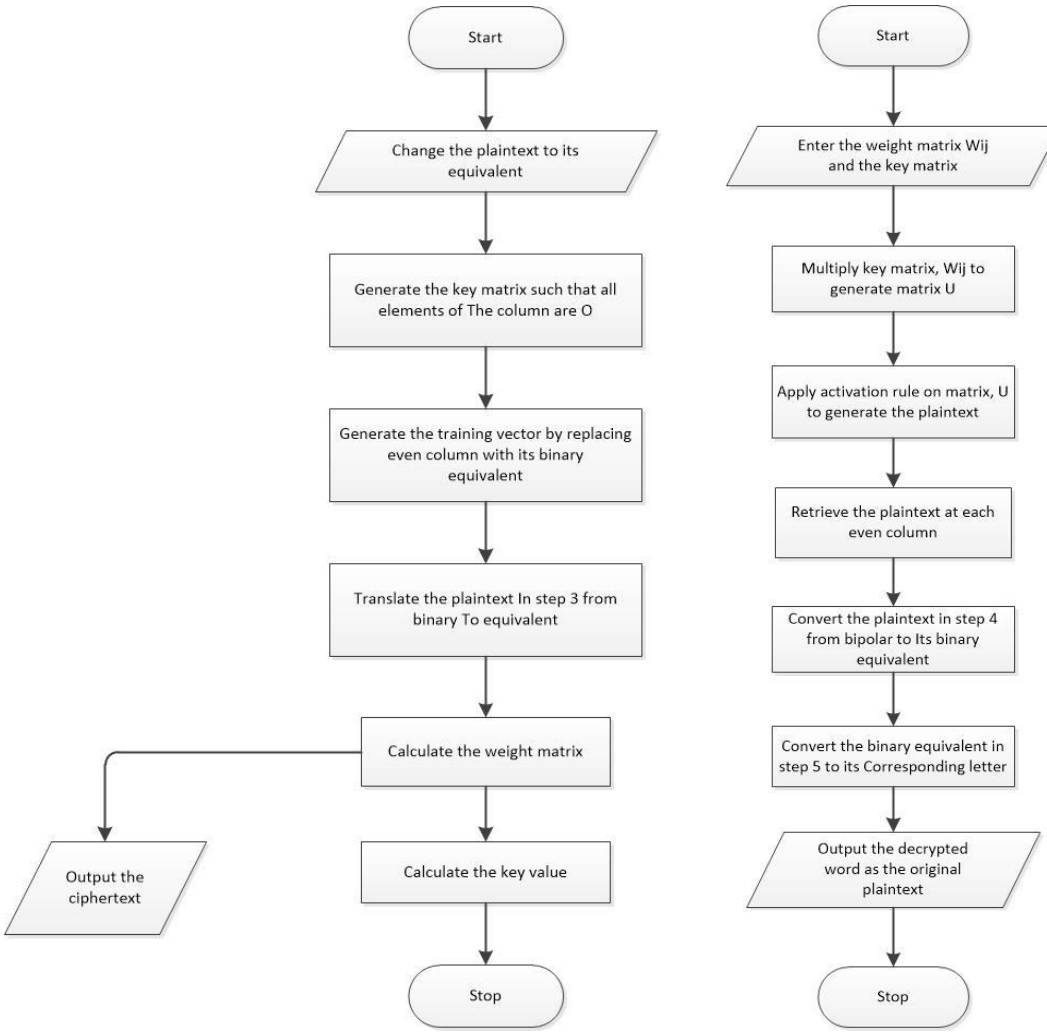

Fig.3. Flowchart Diagram of AANNT (a) encryption (b) decryption

4. Results and Discussion

The proposed LLP and the AANN cryptographic techniques have been implemented in the MATLAB 2018 programming environment and compared based on parameters such as encryption/ decryption times, memory usage, throughput and security. The performance of these techniques was evaluated using text files of various sizes. The experiments were conducted on a Windows 10 Professional Edition computer with a 2.40GHz Dual-processor. The subsequent sections provide and discuss the simulation results.

4.1. Comparative Evaluation based on Encryption Time

The simulation results for this comparison are shown in Table 2 and Figure 4. The average encryption time of AANNT is 0.00167 seconds, while LLPT takes 0.00399 seconds; thus, AANNT exhibits a shorter encryption time and can encrypt data more quickly than LLPT. As the file size increases, LLPT experiences a corresponding rise in encryption time, making it slower. However, the processing time of AANNT occasionally increases with increasing file size. This phenomenon can be attributed to the values of the weight matrix and the network architecture.

4.2. Comparative Evaluation based on Decryption Time

Table 3 and Figure 5 depict the simulation results for the comparison highlighting the comparative performance. Upon examination, AANNT demonstrates faster decryption with an average time of 0.00183 seconds, while LLPT, in contrast, requires an average of 0.00212 seconds to decrypt an encrypted file. Faster decryption times significantly enhance operational efficiency and user experience by facilitating quicker access to encrypted files, messages and data. However, it is crucial to strike a balance between speed and robust security measures to ensure the overall effectiveness of the encryption solution.

Table 2. Comparison of encryption time of LLPT and AANN-based cryptographic techniques

Data text Sizes (Kbytes)	**Encryption time (seconds)**	
	LLPT	**AANNT**
1.50	1.54029	0.00135
2.00	2.13913	0.00168
3.13	2.44140	0.00165
3.75	3.41695	0.00137
4.38	3.81432	0.00158
5.00	4.06579	0.00172
4.50	4.05709	0.00149
6.25	4.91836	0.00175
6.88	6.18703	0.00194
7.50	7.35059	0.00214
Average Encryption time	3.99310	0.00167

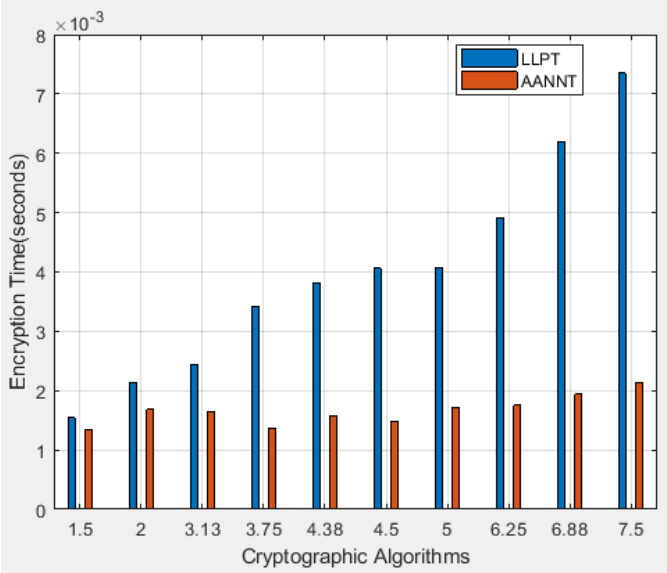

Fig.4. Encryption time of LLPT and AANN-based cryptographic algorithms

Table 3. Comparison of decryption time of LLPT and AANN-based cryptographic algorithms

Data text sizes (bytes)	Decryption Time (seconds)	
	LLPT	AANNT
1.50	0.00131	0.00168
2.00	0.00136	0.00173
3.13	0.00156	0.00179
3.75	0.00175	0.00145
4.38	0.00195	0.00178
4.50	0.00244	0.00170
5.00	0.00217	0.00185
6.25	0.00265	0.00190
6.88	0.00288	0.00207
7.50	0.00311	0.00238
Average decryption time	0.002118	0.00183

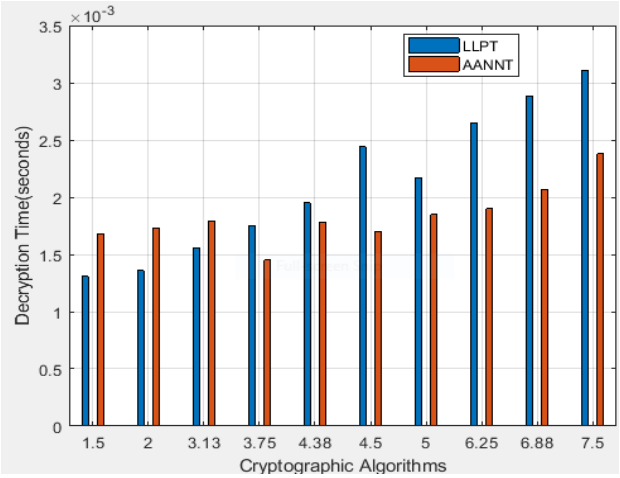

Fig.5. Decryption time of LLPT and AANN-based cryptographic techniques

4.3. Comparative Evaluation Based on Throughput

Table 4 presents the experimental results for comparison. The AANNT outperforms LLPT with a throughput value of 26.29747 Kbyte/sec. This faster encryption throughput signifies that AANNT processes data more quickly with reduced latency than LLPT. Additionally, as given in Table 6, AANNT demonstrates a faster superior decryption throughput rate than LLPT. This correlation suggests that a higher decryption throughput contributes to a faster transformation of encrypted data back to its original form, thereby reducing the energy consumed by the AANNT.

Table 4. Comparative throughput (Kytes/sec) for LLPT and AANN-based cryptographic techniques

Techniques	Average encryption time (sec)	Average decryption time (sec)	Encryption Throughput (Kbyte/sec)	Decryption throughput (Kbyte/sec)
LLPT	0.00399	0.002118	11.25063	21.19452
AANNT	0.01636	0.00183	26.29747	23.95513

4.4. Comparative Evaluation Based on Memory Usage

Table 5 and Figure 6 present the experimental results for evaluating the memory usage of LLPT and AANNT. AANNT requires a smaller amount of memory compared to LLPT. LLPT consumes a larger amount of memory space to process the same data file. Understanding the implications of increased memory consumption is crucial for making informed decisions about the deployment and optimization of cryptographic techniques. This necessitates consideration of factors such as performance, security and resource constraints, ensuring a well-rounded approach to the selection and implementation of the techniques.

Table 5. Comparison of memory usage of LLPT and AANN-based cryptography techniques with different text files

Data text sizes (bytes)	**Memory usage (Kbytes)**	
	LLPT	**AANNT**
1.50	0.726	0.147
2.00	0.942	0.232
3.13	1.140	0.227
3.75	1.520	0.274
4.38	1.710	0.339
4.50	2.290	0.356
5.00	2.080	0.301
6.25	2.620	0.401
6.88	2.930	0.405
7.50	3.550	0.667

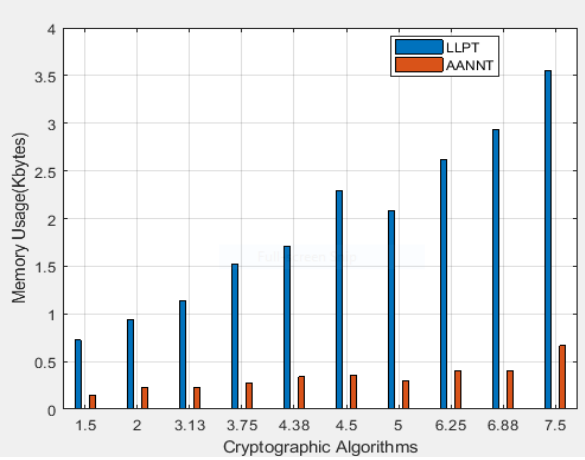

Fig.6. Memory usage of LLPT and AANN-based cryptographic techniques

The following section presents a security analysis of the cryptographic techniques, evaluating their resilience against various attacks including brute force attacks, cipher text-only attacks, model recovery and known plaintext-only attacks.

4.5. Security Analysis of the Cryptographic Techniques

The strength and security of the cryptographic techniques heavily rely on the robustness of the underlying algorithms, the secrecy of the key and resilience against various potential attacks. Various attacks, including brute force attacks, known plaintext-only attacks, ciphertext-only attacks and model recovery attacks were employed to assess the resilience of the two cryptographic techniques.

A brute force attack on AANNT would require an unrealistic amount of time and computational resources due to the need to test all possible binary encryption key combinations. This makes it infeasible within a reasonable timeframe. The effectiveness of a ciphertext-only attack on AANN depends on how robust its network architecture is. Once an attacker manages to obtain the weight matrix and understands the connections or architecture of the AANN, the encryption algorithm of AANNT is susceptible to compromise, potentially leading to a security breach at any time.

Model/key recovery attacks threaten the confidentiality of the AANNT by attempting to extract sensitive information, including network architecture, weights or training data. The compromise of this information introduces potential vulnerabilities.

A known plaintext-only attack is characterized by having pairs of both plaintext and corresponding ciphertext. In executing this attack, a deep understanding of the intricate correlations between specific segments of plaintext and their corresponding ciphertext is required. By strategically combining ciphertext pairs, patterns in the encryption process can be deduced, ultimately leading to the revelation of the encryption key.

However, LLPT offers a higher level of security. A brute force attack is nearly impossible due to the intensive process of finding the Laguerre transform of each piece of plaintext and the necessity for modular arithmetic to evaluate the ciphertext and the encryption key. Moreover, it may be challenging an attacker to invert the encryption process using standard mathematical techniques. The complexity of the LLPT encryption technique increases with the file size, making it laborious and time-consuming to crack the algorithm and find the encryption key.

A known plaintext-only attack is challenging due to the lack of a linear relationship between plaintext and ciphertext, making it difficult to break through the underlying cryptographic algorithm and deduce the secret key. LLPT is designed to withstand ciphertext–only attacks. In such attacks, adversaries are unable to discern any patterns in the available ciphertext, thereby impeding their ability to decrypt the original message.

Table 6. Security analysis of cryptographic attacks

Techniques	Possibility of brute force attack	Possibility of ciphertext-only attack	Possibility of known plaintext-only attack	Possibility of model/key recovery attack
Laplace-Laguerre Polynomial Technique (LLPT)	Impractical and very difficult	Very difficult	Difficult	Impossible
Auto-associative Neural Network (AANN)	Very infeasible	Possible	Possible	Possible
Remarks	Impractical demand for time and computational resources	Due to known architecture and weight matrix	Due to the known weight matrix and correlation between the plaintext and ciphertext	Due to known architecture, weight matrix and training data

The absence of recognizable patterns in the ciphertext serves as a protective measure, effectively thwarting attempts at model/key recovery attacks. This characteristic increases the difficulty for adversaries in deducing information about the encryption key or the LLP cryptographic model in use. The security strengths and weaknesses of the proposed LLP and AANN-based cryptographic techniques are given in Table 6.

5. Conclusions

The evolution of electronic communication and the necessity to secure data exchanged over networks has led to a significant transformation in network security. Cryptography, in particular, has witnessed substantial advancements in the past decade. This study introduces an innovative cryptographic technique based on the Laplace-Laguerre polynomial. The performance of the proposed Laplace-Laguerre Polynomial Technique (LLPT) was evaluated and compared to an existing AANN cryptographic technique in terms of encryption time, decryption time, throughput, memory usage and security. Simulation results show that AANNT outperforms the proposed LLPT in terms of processing time, throughput and memory utilization. AANNT uses a same-key cryptography technique that is dependent on network weight and architecture, making it vulnerable to attacks if these details are known. It is crucial to highlight that the integrity and confidentiality of data encrypted with AANNT are not guaranteed.

The security analysis of the LLPT is a rigorous task, as breaking the algorithm to unveil the encryption key is both time-consuming and effort-intensive. LLPT demonstrates resilience against various attacks such as brute force attacks, known plaintext-only attacks, ciphertext-only attacks and model/key recovery attacks. This overall robustness underscores LLPT's effectiveness in preserving the integrity, confidentiality and security of encrypted data, establishing a strong defence against unauthorized access. LLPT can be effectively deployed within the Local Area Network (LAN) environments. By leveraging the mathematical properties of the Laguerre and Laplace distributions, LLPT establishes a robust encryption mechanism. This implementation serves to protect sensitive data transmitted within the LAN, preventing unauthorized access and interception, thereby enhancing overall system security.

References

[1] J. Alshehri and A. Alhamed, "A review paper for the role of cryptography in network security", in 2022 IEEE 4th International, Conference on Electrical, Control and Instrumentation Engineering, 2022, pp.1-5. doi:10.1109/ICECIE55199.2022.10000338.

[2] E. Omkar and D. Mohite, "Encrypting viruses", International Journal for Research Trends and Innovation, vol. 7, no.6, 2022, pp. 2456-3315. https://ijrti.org/papers/IJRTI2206262.pdf.confidentiality.

[3] S. M. Naser, "Cryptography: from the ancient history to now, its applications and a new complete numerical model", International Journal of Statistics Studies, vol.9, no.3, pp.11-30.https://www.eajurnals.org.

[4] S. Das, A.K Balmiki and K. Mazumdar, "A review on Al-ML based cyber-physical systems security for industry 4.0", Intelligent, Cyber-Physical, System, Security for Industry. 2022, pp. 203-216. doi:10.1201/978100324134811.

[5] H. Xu, K. Thakur, A.S.Kamruzzaman and M. L. Ali, "Applications of cryptography in Database: a review", IEEE International IoT, Electronics and Mechatronics Conference, 2021, pp. 1-6. doi:10.1109/IEMTRONICS52119.2021.9422663

[6] A.P. Hiwarekar, "Application of Laplace transform for cryptographic ", International Journal of Engineering and Research, vol. 5, no. 4, 2015, pp. 129-135.

[7] E.O. Adeyefa, L.S.Akinola and O.D.Agbolade, "A new cryptographic scheme using the chebyshev polynomials", IEEE International Conference in Mathematics, Computer Engineering and Computer Science, 2020, pp.1-3. doi:10.1109/ICMCECS47690.2020.240868.

[8] K. Bhuvaneswari, "Mohand transform based cryptography technique", International Journal of food and nutrition science, vol. 11, no. 3, 2022, pp. 3294-3298.

[9] Z. E. Huma, J.U Rahman, M. Suleman and N. Anjum, "Cryptographic method based on natural-elzaki transforms", imanager's Journal on mathematics, vol.11, no.1, 2022, pp. 39-46. doi:10.26634/jmat.11.1.18511.

[10] E. A. Mansour and N. K. Meftin "Mathematical modeling for cryptography using Jafari transformation method", Periodicals of engineering and natural sciences, vol.9, no. 4, 2021, pp. 892-897.

[11] M. N. Alenezi, "A study of Z-transform based encryption algorithm", International Journal of Communication Networks and Information Security", vol.13, no.2, 2021, pp. 302-309. doi:10.17762/ijcnis.v1312.5052.

[12] N. O. Onuoha "Kamal transform technique to coupled systems of linear ordinary differential equations", IOSR Journal of Mathematics, vol.19, no. 4, 2023, pp.24-29.

[13] T. Sivakumar, Pandi Malaichamy, N. Senthilmadasamy and R. Bharathi, "An image encryption algorithm with Hermite chaotic polynomials and scan pattern", Journal of Physics Conference. Series, vol. 1767, no. 1, 2021, pp.1-14. doi:10.1088/1742-6596/1767/1/012044

[14] S. A. Osikoya, E. O Adeyefa, Jensen-based new cryptographic scheme, Journal of Nigerian Society of Physical Sciences, vol.4, 2022, pp.49-53.

[15] M. T. Gencoglu, "Cryptanalysis of a new method of cryptography using Laplace transforms hyperbolic functions", Communications in mathematics and applications, vol. 8, no. 20, 2017, pp 183-189. doi:10.26713/cma.v8i2.708

[16] M. Ayush and G. Ravindra (2019), "Kamal transformation based cryptographic technique in network security involving ASCII value", International Journal of Innovative Technology and Exploring Engineering. vol. 8, no. 12, 2019, pp. 3448-3450. doi:10.35940/ijitee.L2592.1081219.

[17] J. S. Shivaji and A. P. Hiwarekar, "Cryptographic method based on Laplace-elzaki transform", Journal of the Maharaja Sayajirao University of Baroda, vol. 55, no. 1, 2021, pp. 187-191. https://www.researchgate.net/publication/35304410_CRYPTOGRAPHIC_METHOD_BASED_ON_LAPLCE-ELZAKI_TRA

NFORM
[18] A. K. H Sedeeg, M. M. A. Mahgoub, M. A. Saif Saheed, "An Application of the New Integral "Aboodh Transform", Pure and Applied Journal, vol. 5, no. 5, 2016, pp 151-154. https://www.sciencepublishinggroup.com/article/10.11648.j.pamj.20160505.12.
[19] S. Bhusare, S.B. Thorat, S. Sandeep, S. Seema, M.V. Ramnamurth, "Symmetric encryption algorithm for data security privacy using linear convolution sum technique", International Journal of Emerging Technologies and Innovative Research, vol.6, no.6, 2016, pp. 7-10. https://www.jetir.org/ papers/JETIR1906N04.pdf
[20] B. S. Riggan, C.Reale and N .M Nasrabadi, "Coupled auto-associative neural networks for heterogeneous face recognition, in IEEE Access, vol. 3, 2015, pp. 1620-163. doi:10.1109/ACCESS.2015.2479620.
[21] M. Wang and S. Chen S, "Enhanced FM-AM based on empirical kernel map", IEEE Transaction on Neural Networks, vol. 16, no. 3, 2005, pp. 557-564. doi:10.1109/TNN.2005.847839.
[22] Z. M. Zin, "Using auto-associative neural networks to compress and visualize multidimensional data", 11th International Conference on Ubiquitous Robots and Ambient Intelligence, 2014, pp. 408-412. doi:10.1109/URAI.2014.7057451.
[23] P. Wang and C. Cox, "Study on the application of auto-associative neural network", IEEE Proceedings, 7th International Conference on Signal Processing, vol.2, 2004, pp 1570-1573. doi:10.1109/ICOSP.2004.1441629.
[24] M. Elnour, N. Meshin, and M. Al-Naemi, "Sensor fault diagnosis of multi-zone HVAC system using auto associative neural network", IEEE Conference on Control Technology and Applications, 2019, pp.118-123. doi:10.1109/CCTA.2019 8920554.
[25] W. Ha and C. Shin, "Seismic random noise attenuation in the laplace domain using singular value decomposition", in IEEE Access, vol. 9, 2021, pp. 62029-62037. doi:10.1109/ACCESS.2021.3074648.
[26] A. M. Al-Azzani, M.A.M.Rageh, G. H. Al-Gaphari, "A new cryptography scheme based on laplace transform and substitution-permutation network, International Journal of Advanced Trends in Computer Sciences and engineering", vol.10, no. 4, 2021, pp.2658-2663.
[27] M. Al-Mazmumy and A. Alsulami Aishah, "Solution of laguerre's differential equations via modified domain decomposition method", Journal of Applied Mathematics and Physics, vol. 11, no. 1, 2023, pp 85-100. doi: 10.4236/jamp.2023.111007.
[28] A.K. Shukla, I. A.Salehbhia and J. C. Prajapati, "On the laguerre transform in two variables", Integral transform and special functions, vol.20, no.6, 2009, pp.459-470. doi:10.1080/10652460802645818.
[29] S. Wolfgang,"Special functions in physics with MATLAB", March 2021, pp. 211-214. doi:10.1007/978-3-030-64232-7
[30] M. Tuma, "Application of Laguerre functions to data compression", European grant projects/result/research and development science, 2013, pp. 1-4. https://api.semanticscholar.org/CorpusID:37469761.
[31] S. Amit and K.P. Rajesh, "Laguerre polynomial based numerical method to solve a system of generalized abel integral equations", International Conference on Modeling, Optimization, and Computing. vol. 38, 2012, pp. 1675-1682. doi: 10.1016/j.proeng.2012.06.204.

Authors' Profiles

Lateef Adesola (LA) Akinyemi (Senior Member, IEEE) received the B.Sc. degree (Hons.) in ECE (computational electronics) and the M.Sc. degree in ECE from Lagos State University, Nigeria, the M.Sc. degree in EEE (communication engineering option) from the University of Lagos Akoka, Nigeria, and the Ph.D. degree in electrical engineering from the Department of Electrical Engineering, University of Cape Town, South Africa. He is a lecturer, a researcher and a scholar with the Department of Electronic and Computer Engineering, Lagos State University, Lagos. Currently, he is a postdoctoral research fellow at the Departments of Computer Science and Electrical Engineering, CSET, University of South Africa, and the Centre for Augmented Intelligence and Data Science (CAIDS), Johannesburg, UNISA, South Africa. He is also a member of the following associations and societies: COREN, NSE, IITPSA, SAAIA, and NIEEE. His research areas are wireless communications, computational electronics, modelling and simulations of quantum-inspired nanoparticles and devices, digital signal processing and vision, microwave engineering and antennas, artificial intelligence-inspired algorithms, data science, machine learning, eHealth and E-mobile and 4IR and its applications.

Bukola H. Akinwole is a lecturer and researcher in the Department of Electrical/Electronic Engineering, University of Port-Harcourt, Rivers state, Nigeria. Her research interests cover the field of wireless communication, wireless security, Artificial intelligence and bio-medical engineering. She is a registered engineer with the Council for the Regulation of Engineering in Nigeria (COREN) and a member of Nigeria Society of Engineer in Nigeria (MNSE).

How to cite this paper: Lateef A. Akinyemi, Bukola H. Akinwole, "Performance Evaluation of Laguerre Transform and Neural Network-based Cryptographic Techniques for Network Security", International Journal of Intelligent Systems and Applications(IJISA), Vol.16, No.3, pp.1-17, 2024. DOI:10.5815/ijisa.2024.03.01