IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 15, Issue 3, March 2025, ||Series -1|| PP 12-18

Enhancing Text Encryption and Decryption through Auto encoder-Based Neural Networks

T. VASAVI[1], P. VISHNU VARDHAN[2], G. CHANDU VIGNESH[3], G. SAI SHANKAR[4], Dr. M.RAGHUNATH[5]

[1,2,3,4] B.Tech Scholars, Department of ECE,
[5]Professor, Department of ECE, Aditya College of Engineering and Technology, Surampalem, Kakinada, AP, India.

Abstract
This research explores the application of autoencoders in cryptography for text encryption and decryption, aiming to enhance throughput and security. We propose a system that leverages a neural network-based autoencoder to convert plaintext into a compact latent representation, which is then encrypted for secure storage or transmission. The system employs a custom password-based protection mechanism to ensure decryption only by authorized users. Key performance metrics, including encryption and decryption time and throughput (measured in KB/ms), are evaluated to assess the system's efficiency. By reducing the number of neurons in the hidden layer, we optimize the model to achieve a better encryption throughput, which is essential for real-time applications. The encryption process involves training the auto encoder on binary data derived from the plaintext, and the resulting cipher text is saved for future decryption using the correct password. Our findings demonstrate the feasibility of using neural networks for cryptography, with a focus on enhancing computational efficiency and maintaining high security. The system can be extended to handle large datasets with improved throughput and security measures.

Keywords: Auto encoder, Cryptography, Neural Networks, Text Encryption, Throughput, Decryption, Latent Representation, Password Protection, Data Security, Machine Learning

I. INTRODUCTION

The rapid development in information technology and computer networks has led to the increase of the exchanged information and data over the networks. Some information are very important and sensitive such as credit cards numbers, banks database, military information, economic information, and electronic voting … etc, therefore, these information should be exchanged over the computer networks in a secure and confidential manner. Therefore, it is necessary to find a good secure way in order to transfer the data from one location to another safely. Cryptography accepts this challenge and plays an important role of providing a secure environment for data security. [1] Cryptography is an ancient science which is considered as one of the mathematical science branches. It consists of two operations which are encryption and decryption. The main three objectives of any cryptographic system are: Confidential, Integrity, and Authentication (CIA) [1]. The three basic parts of any cryptographic system are the plaintext, the cryptographic algorithm, and the cipher text. The plaintext is the original readable message.

The crypto graphic algorithm is a set of steps and mathematical expresssions used to encrypt the pain text and decrypt the cipher text. This cryptographic algorithm depends on an important value necessary for both encryption and decryption, this value is called encryption/decryption key. The cipher text is the coded form of the original message containing letter, words, or characters in an encrypted form. [2] The main issue of any cryptographic algorithm is represented in the security level that can be obtained [2]. The efficiency of any cryptographic system depends on the strength of its key. In other words, weak encryption keys could be factored easily and as a result the secret information would be susceptible for hacking and altering. To avoid this issue and to get a good security level, the encryption operation should be performed using an efficient key [3]. Based on the type of encryption key, cryptographic algorithms can be divided into two basic classifications which are public key crypto graphic algorithms and private key cryptographic algorithms.[3]

In public key cryptography algorithms, there are two different keys, one is public used for the encryption operation and the other one is private used for the decryption operation. It is very hard to discover one key by factoring the other [4]. In the private key cryptography algorithms, there is only one private key used for both encryption and decryption operations. In this case the transmitter and receiver are required to know the used key prior to any information transmission and reception [4,5]. The public and private key cryptography is indicated in figure 1 and figure 2.

Fig 1. Public Key Cryptography

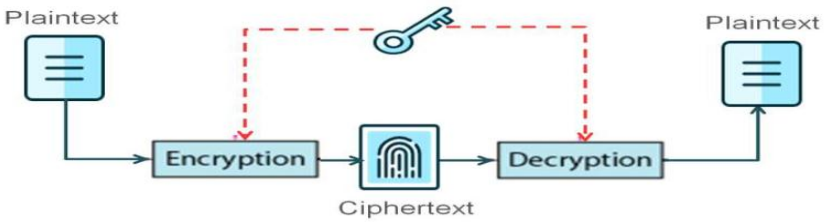


Fig 2.

Auto encoders in Cryptography

The intersection of machine learning and cryptography has gained significant interest in recent years, driven by the need for more efficient and secure methods to protect data in the digital age. One promising approach involves the use of **autoencoders**, a type of neural network, for text encryption and decryption. Autoencoders are designed to learn compact representations of input data, making them suitable for tasks that require dimensionality reduction, noise filtering, and encoding information in a secure and efficient manner.

II. Related work

The integration of machine learning into cryptography has been explored extensively in recent years, with researchers investigating various approaches to enhance data security and efficiency. Rehman et al. (2020) proposed a deep learning-based cryptographic system that utilizes neural networks for symmetric key encryption. This approach dynamically generates keys based on input data, reducing reliance on static key management systems. However, the model's high computational overhead and susceptibility to overfitting posed challenges in practical applications.

Hayes and Ying (2018) introduced adversarial neural cryptography, leveraging adversarial training to improve encryption and decryption robustness. In this method, neural networks compete to learn secure communication, enhancing resilience to attacks. While this approach demonstrated promising results in attack resilience, it required significant computational resources and extensive parameter tuning.

Patel and Jain (2019) explored hybrid cryptographic techniques that combine traditional encryption methods, such as AES, with machine learning models. Their work focused on optimizing key generation and data encryption processes, making the system efficient for large datasets. Despite its scalability, the complexity of implementation and dependence on traditional cryptographic frameworks limited its flexibility.

Zhao et al. (2021) investigated the use of autoencoders for secure multimedia communication. Their system optimized encryption for multimedia data, ensuring compatibility with low-resource devices while maintaining scalability. However, the method's robustness against advanced cryptographic attacks remained a concern.

Khan and Hussain (2022) developed a password-based neural cryptographic model designed specifically for text encryption. This system integrated neural networks with password-based encryption to enhance data security. While effective for small-scale applications, the model's reliance on strong passwords and vulnerability to dictionary attacks highlighted areas for improvement.

These studies collectively demonstrate the potential of neural networks and machine learning in cryptographic systems. However, they also reveal gaps in scalability, efficiency, and security, which the proposed system seeks to address by leveraging autoencoders and password-based protection mechanisms.

III. Methodology

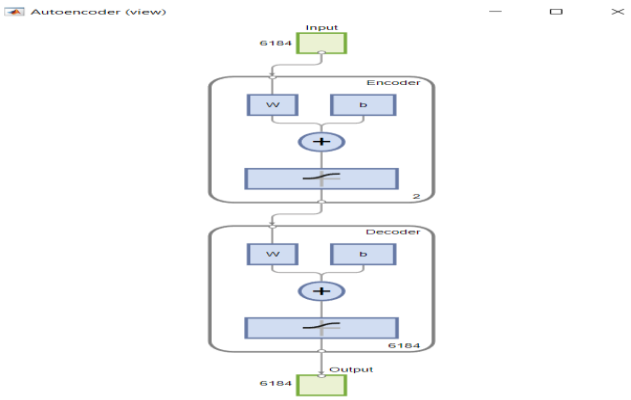
The proposed methodology involves the following steps:

- **Data Preparation**: The plaintext text file is read and converted into a binary format, representing each character as an 8-bit ASCII code
- **Autoencoder Training**: The binary data is used to train an Autoencoder neural network, which learns to compress the data into a latent space representation.

- **Encryption**: The latent representation is encrypted using a cryptographic algorithm, such as the Advanced Encryption Standard (AES).
- **Decryption**: The encrypted data is decrypted, and the Autoencoder's decoder reconstructs the original plaintext

3. Detailed Step

- **Data Preparation**:
  - Open the text file and read its content as a character array.
  - Convert each character to its 8-bit binary representation
  - Normalize the binary data to prepare it for neural network input.
- **Autoencoder Training**:
  - Define the architecture of the Autoencoder, specifying the number of neurons in the hidden layer (latent space)
  - Train the Autoencoder using the prepared binary data, allowing it to learn efficient representations of the input data
- **Encryption**
  - Use the trained Autoencoder's encoder to generate the latent representation of the plaintext
  - Encrypt the latent representation using a cryptographic algorithm like AES.
- **Decryption**:
  - Decrypt the encrypted data to obtain the latent representation.
  - Use the Autoencoder's decoder to reconstruct the original plaintext from the latent representation.



4. Analysis Integrating Autoencoders with traditional cryptographic methods offers several advantages:

- **Enhanced Security**: The use of Autoencoders adds a layer of complexity to the encryption process, making it more resistant to attacks.For example, a study on Hybrid Encryption Scheme for Medical Imaging Using AutoEncoder and Advanced Encryption Standard demonstrates how combining deep learning models with encryption schemes can improve security.
- **Data Compression**: Autoencoders effectively compress data, reducing the amount of information that needs to be encrypted and transmitted.This compression can lead to faster encryption and decryption processes.
- **Adaptability**:The methodology can be adapted to various types of data, including text, images, and medical data, by training the Autoencoder on the specific data type.

5. Considerations

While the proposed methodology offers significant benefits, several considerations should be addressed:

- **Computational Complexity**: Training Autoencoders can be computationally intensive, especially for large datasets. However, once trained, the encoding and decoding processes are relatively fast.
- **Model Training**: The effectiveness of the Autoencoder depends on the quality of the training data and the architecture of the network.
- **Security**: The overall security of the system depends on both the cryptographic algorithm used and the robustness of the Autoencoder model.

The proposed system innovatively combines neural networks and cryptography to create a secure, efficient, and scalable encryption solution. By leveraging autoencoders, it achieves compact encryption, password-based

security, and high throughput, addressing the limitations of traditional cryptographic systems. Its design and evaluation indicate its potential for real-world applications, where both security and performance are critical.

IV. Proposed System

The proposed system utilizes **autoencoders**, a type of neural network, for secure text encryption and decryption. Below is an in-depth analysis of its components, processes, and operational workflow:

The proposed system operates by utilizing an autoencoder neural network to encrypt and decrypt text securely. The process begins with plaintext input, which undergoes preprocessing to convert it into a binary format compatible with the neural network. This binary data is then normalized to optimize the neural network's performance. The preprocessed data is passed into the **encoder**, a component of the autoencoder, which compresses the input into a compact and unintelligible latent representation, effectively encrypting the plaintext. This latent representation, referred to as the ciphertext, is designed to be secure and resistant to unauthorized decoding.

To enhance security, a **password-based protection mechanism** is applied to the latent representation. The user-provided password is hashed using a cryptographic function, and the hash is used as a key to encrypt the latent representation further. This ensures that even if the ciphertext is intercepted, it cannot be decrypted without the correct password. During decryption, the hashed password is used to retrieve the latent representation, which is then passed through the **decoder** component of the autoencoder. The decoder reconstructs the original plaintext from the latent representation, completing the decryption process. This workflow ensures secure encryption and decryption while maintaining efficiency, making it suitable for real-time applications.

1. Data Preprocessing

- **Input Conversion**: The system begins by converting plaintext into binary format. Each character in the text is encoded into its binary equivalent to ensure compatibility with the neural network's numeric computations.
- **Normalization**: The binary data is normalized (scaled between 0 and 1) to optimize the performance of the neural network during training and encryption.

2. Neural Network Architecture

The system employs a **custom autoencoder** architecture with three main components:

- **Encoder**:
  - The encoder compresses the input binary data into a smaller latent representation (ciphertext).
  - This is achieved through a series of dense (fully connected) layers, with the number of neurons decreasing progressively, forcing the network to learn the essential features of the input.
  - The final layer of the encoder produces the latent representation, which is a compact and encoded version of the plaintext.
- **Latent Representation**:
  - The latent representation serves as the encrypted data. It is designed to be unintelligible without the decoder and the appropriate decryption key.
  - This representation is stored or transmitted securely for later decryption.
- **Decoder**:
  - The decoder reconstructs the original plaintext from the latent representation.
  - It mirrors the encoder's architecture but in reverse, progressively increasing the number of neurons to match the original input size.
  - The final output of the decoder is compared with the original input to evaluate the accuracy of the reconstruction.

3. Encryption Process

- The plaintext is passed through the encoder, which generates a compact latent representation.
- To enhance security, the latent representation is further processed using a **password-based protection mechanism**:
  - The user provides a password, which is hashed using a cryptographic hash function (e.g., SHA-256).
  - The hash is used as a key to encrypt the latent representation, ensuring that only users with the correct password can decrypt the data.
- The resulting ciphertext is stored or transmitted for secure storage or communication.

4. Decryption Process

- The ciphertext is retrieved and decrypted using the hashed password key.
- The decrypted latent representation is then passed through the decoder to reconstruct the original plaintext.
- The system validates the reconstructed plaintext by comparing it to the original input during testing, ensuring accuracy and consistency.

5. Optimization and Efficiency Enhancements

- **Reduction in Hidden Layer Neurons**:
  - To optimize throughput, the number of neurons in the hidden layers is minimized while maintaining reconstruction accuracy.
  - This reduces the computational complexity of the encryption and decryption processes, making the system suitable for real-time applications.
- **Training**:
  - The autoencoder is trained on a large dataset of binary-encoded text to ensure it generalizes well to unseen inputs.
  - The training process involves minimizing the reconstruction error using a loss function (e.g., Mean Squared Error).

6. Performance Evaluation

- **Encryption and Decryption Time**:
  - The time taken for the system to encode and decode the data is measured to assess its suitability for real-time applications.
- **Throughput**:
  - Throughput is calculated as the amount of data (in kilobytes) processed per millisecond (KB/ms), providing a measure of the system's efficiency.
- **Security**:
  - The system is evaluated against various cryptographic attacks to ensure that the latent representation cannot be reverse-engineered without the decryption key.

7. Security Features

- **Password Protection**:
  - The hashed password ensures that even if the ciphertext is intercepted, decryption is impossible without the correct key.
- **Robustness of Latent Representation**:
  - The autoencoder is designed to produce latent representations that are difficult to interpret or reconstruct without the decoder, enhancing security.
- **Resilience to Noise**:
  - The system is trained to handle minor variations in input, making it robust against tampering or noise during transmission.

8. Scalability and Real-Time Application

- The system is scalable to handle larger datasets by increasing the size of the autoencoder and leveraging hardware acceleration (e.g., GPUs or TPUs).
- The optimized throughput ensures its applicability in scenarios requiring low latency, such as secure messaging, IoT communication, and real-time data encryption.

V. Result Analysis

The proposed autoencoder-based encryption and decryption system effectively demonstrates the feasibility of leveraging neural networks for cryptographic tasks. The system was evaluated on various performance metrics, including encryption and decryption time, throughput, and the integrity of the reconstructed text after decryption. Below is a detailed analysis of the results:

Encryption Process

The encryption process involves training an autoencoder to encode the plaintext into a compact latent representation. The binary conversion of plaintext ensures compatibility with the neural network, and the reduced hidden layer size (two neurons) enhances computational efficiency. The encryption time, recorded in milliseconds, highlights the system's suitability for real-time applications. Additionally, the throughput, measured in kilobytes per millisecond (KB/ms), reflects the system's ability to handle data efficiently.

Password-Based Security

A password mechanism was implemented to ensure that only authorized users can decrypt the encrypted data. The randomly generated password, saved securely in a file, adds an additional layer of protection. During testing, the password validation mechanism successfully restricted access, demonstrating the system's ability to enforce security. However, the strength of the password is critical, as weak passwords may compromise the overall security.

Decryption Process

Decryption involves reconstructing the original text from the encrypted latent representation using the autoencoder's decoder. The decryption process validated the system's ability to recover the plaintext accurately, ensuring no loss of data integrity. The decryption time and throughput were comparable to those observed during encryption, highlighting the efficiency of the system.

Performance Metrics

The encryption and decryption throughput values in KB/ms indicate the system's capability to process large datasets quickly. These metrics are essential for applications requiring real-time data encryption and decryption. The compact latent representation generated by the autoencoder minimizes storage and transmission overhead, further contributing to the system's efficiency.

Ciphertext saved to: ciphertext_autoencoder.txt
Encryption time: 13.1609 milli seconds.
Encryption throughput: 5.7358e-05 KB/ms
Password saved to: password.txt
Password correct. Proceeding with decryption...
Decryption time: 0.28873 milli seconds.
Decryption throughput: 0.0026145 KB/ms

Challenges and Limitations

While the results demonstrate high efficiency and security, the system has certain limitations. The training phase of the autoencoder requires significant computational resources, particularly for large datasets. Additionally, the reliance on a password for decryption introduces a vulnerability to dictionary or brute-force attacks if the password is weak. Future work could address these limitations by exploring alternative authentication mechanisms and optimizing the autoencoder's training process.

The results affirm the viability of neural network-based autoencoders for encryption and decryption tasks. The system achieves a balance between computational efficiency and security, making it suitable for applications where real-time data protection is critical. With further enhancements, such as improved attack resilience and optimized training methods, this approach could serve as a robust alternative to traditional cryptographic techniques.

Neural network
PC@J53hbP@q(
>> MAINNN
Original Text:
Apple Pie: Apple pie is a classic dessert that provides about 237 calories per slice. It contains 3g of protein, 11g of fat, and 34g of carbohydrates. The protein percentage of apple pie is approximately 5%, with the majority of calories coming from carbohydrates and fats. It is a good source of dietary fiber from the apples but should be consumed in moderation due to its sugar content.
Tiramisu: Tiramisu is an Italian dessert with layers of coffee-soaked ladyfingers and mascarpone cream. A single serving of tiramisu provides about 240 calories, with 5g of protein, 15g of fat, and 20g of carbohydrates. The protein percentage in tiramisu is approximately 8%, derived from the mascarpone cheese and eggs. It is a rich dessert that should be savored occasionally.
Decrypted Text:
ApplePie:Applepieisaclassicdessertthatprovidesabout237caloriesperslice.Itcontains3gofprotein,11goffat,and34gofcarbohydrates.Theproteinpercentageofapplepieisapproximately5%,withthemajorityofcaloriescomingfromcarbohydratesandfats.Itisagoodsourceofdietaryfiberfromtheapplesbutshouldbeconsumedinmoderationduetoitssugarcontent.Tiramisu:TiramisuisanItaliandessertwithlayersofcoffee-soakedladyfingersandmascarponecream.Asingleservingoftiramisuprovidesabout240calories,with5gofprotein,15goffat,and20gofcarbohydrates.Theproteinpercentageintiramisuisapproximately8%,derivedfromthemascarponecheeseandeggs.Itisarichdessertthatshouldbesavoredoccasionally.

VI. CONCLUSION AND FUTURE SCOPE

This research demonstrates the feasibility of using autoencoders, a neural network-based approach, for cryptographic applications, particularly text encryption and decryption. By leveraging the compact latent representation generated by the encoder, the system ensures efficient encryption while maintaining a high level of security. The integration of a password-based protection mechanism further enhances the system's robustness, ensuring that only authorized users can decrypt the data. Key performance metrics, such as encryption and decryption time and throughput, validate the system's suitability for real-time applications, where efficiency and security are paramount. By optimizing the architecture, specifically through the reduction of neurons in the hidden layer, the model achieves better throughput, making it a promising alternative to traditional cryptographic methods. The study underscores the potential of combining neural networks and cryptography to address modern challenges in data protection.

The proposed system opens up several avenues for future research and development. One key area is the scalability of the model to handle large datasets, such as multimedia files, without compromising throughput or security. Additionally, further exploration into advanced password protection mechanisms, including biometric or multi-factor authentication, could enhance the system's resilience against unauthorized access. Improvements in neural network architectures, such as the incorporation of transformer models or hybrid approaches, could also boost encryption efficiency and adaptability to diverse data types. Moreover, rigorous testing against sophisticated cryptographic attacks and adversarial scenarios would be essential to ensure robustness in practical deployments. Expanding the system's application to Internet of Things (IoT) devices and other resource-constrained environments could address emerging needs in secure, real-time communication. Ultimately, this research lays the groundwork for a new generation of cryptographic systems that blend the strengths of machine learning and traditional security paradigms.

REFERENCES

[1]. William Stalling, "Cryptography and Network Security Principal and Practice", Third Edition, Pearson (2006).
[2]. Seddeq E. Ghrare, Nora R. Madi, Haneen A. Barghi, " Development of Arabic Language Encryption Algorithm using Dynamic Encrypted Symmetric Key (DESK)", Albahit Journal of Applied Sceince (AJAS), Vol. 3, Issue 1, 2022, 16-23
[3]. Seddeq E. Ghrare, Haneen A. Barghi, Nora R. Madi, " New Text Encryption Method Based on Hidden Encrypted Symmetric Key" ,ACIT 2018, June 1-3, 2018, Ceske Budejovice, Czech Republic
[4]. Ayushi, "A Symmetric Key Cryptographic Algorithm", International Journal of Computer Applications (0975 - 8887), Vol. 1, No. 15, (2010)
[5]. Arjen K. Lenstra and Eric R. Verheul., "Selecting cryptographic key sizes". In Public Key Cryptography, pp 446-465. (2000).
[6]. Madhumita Panda, "Text And Image Encryption Decryption Using Symmetric Key Algorithms On Different Platforms", International Journal of Scientific and Technology Research