International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Designing A Novel Covert Communication Medium for Secure Information Exchange

Sridhar Iyer[1*], Dr. Narendra Shekokar[2], Sharvari Patil[3]

Submitted: 25/04/2024 **Revised**: 26/05/2024 **Accepted**: 24/06/2024

Abstract: With the escalating concerns surrounding information security in today's digital landscape, the demand for covert communication channels that facilitate secure information exchange has witnessed an exponential surge. This research endeavors to design and implement a novel covert communication medium to address the challenges inherent in clandestine information transfer. The proposed medium harnesses a new age intelligent encryption technique to ensure not only the confidentiality but also the stealthiness of communication. The study commences with a comprehensive analysis and development of a concept that seamlessly integrates state-of-the-art encryption algorithms along with ways to make the communication covert. The primary objective is to strike a delicate balance between data security and the covert nature of communication, enabling information to be exchanged undetected within various digital environments. The proposed system explores the potential of leveraging neural networks and deep learning principles to encrypt data traversing insecure communication mediums. The research findings contribute significantly to the advancement of covert communication technologies, offering a robust solution for secure information exchange in environments where traditional encryption methods may prove insufficient or where a more covert approach to encryption is necessitated. The research finding also does benchmarking tests to show the amount of efforts required to break into an intelligent cryptographic algorithm to obtain the keys in totality. The implementation is evaluated on the basis of the accuracy with with the same plaintext is generated and also its ability to withstand various attacks.

Keywords: *Covert communication, Secure information exchange, Encryption, Generative Adversarial Networks, Artificial intelligence, adversarial cryptography*

1 Introduction

The inspiration for this research stems from the realization that current cryptographic approaches have been in use for a considerable time and could benefit from a reevaluation or refinement in terms of security. We all know that AES-256 is one of the most secure [1] cryptographic algorithms being used today for even the most highly sensitive information exchange. Because of its robustness against brute force attacks and other sophisticated network attacks [1], the AES Algorithm is the method of choice, but can we turn complacent and leave everything to the toughness of AES against these known attacks? The answer should be no, as we are still not aware of the hidden zero day vulnerabilities that may creep up one fine day without even any hint of its arrival.

If anything like that happens, then the entire world could come to a complete standstill. Every basic financial transaction on the internet is secured using SSL over the HTTPS which in turn uses AES for its protection. If the AES itself collapses, the algorithm gets into the public domain and within no time, the entire financial backbone worldwide collapse. Billions and Zillions of money irrespective of the currency will be at stake.

Some research suggests that although AES is considered to be the most secured, but few cryptanalysis attempts on AES were able to guess few bits of the key successfully using side channel attacks [2]. Such Power Analysis based Side Channel attacks on AES showed that although its virtually impossible to crack AES completely using the existing computational power, its not the case if an alternate or unconventional route is opted to break into the algorithm implementation rather than guessing the key. Currently AES is the reigning champion and yet to lose a match but what about ten to twenty years down the line. What if we invent a highly powerful quantum computer much powerful than the existing supercomputers come into existence? It could break the existing AES implementation in minutes if not in seconds [3] . What will happen then? We need to think out of the box and start thinking of developing some alternative approach to the existing ones, not with the intent of immediately replacing them but atleast provide a totally different approach of providing security to the data which could eventually be the method of choice. In this research work, a novel cryptographic approach is proposed, which would encrypt the plaintext using an intelligently

[1*]*Department of Computer Engineering, Mumbai, Maharashtra, India, Email: c.sridhar89@gmail.com orcid: 0000-0003-3964-2476*
[2]*Department of Computer Engineering Mumbai, Maharashtra, India, Email: narendra.shekokar@djsce.ac.in orcid: 0000-0002-2507-4140*
[3]*Department of Computer Engineering, Mumbai, Maharashtra, India, Email: sharvari.patil@djsce.ac.in orcid: 0000-0002-0721-8788*
***Corresponding Author:** Sridhar Iyer*
[*]*Department of Computer Engineering, Mumbai, Maharashtra, India, Email: c.sridhar89@gmail.com orcid: 0000-0003-3964-2476*

developed algorithm using adversarial networks, which could develop their own algorithm, generate keys and create the ciphertext and send it to the receiver for decryption. The receiver on the other hand will also be a neural network trained to retrieve the plaintext out of the cipher text.

The entire scenario will be kept completely covert [4] in the following manner:

a) The ciphertext generated will be used to generate an image intelligently and this image will be sent to the receiver as a cipher image instead of the ciphertext.

b) This cipher image will not be a stego image as generally used in steganography.Stego Images consist of the encrypted text as part of the image itself.

c) The ciphertext will not be hidden inside the image in any respect, which mightincrease the size of the image.

d) The receiver upon receiving the cipherimage will try to retrieve the ciphertextfrom the cipherimage using the knowledge obtained through the training.

This entire process is hidden from the attacker and gives rise to a completely covert communication. Even if the attacker gets access to the cipher image, until he/she gets the understanding of the reverse mapping process, the attacker might not be able to break the code. The article is systematically broken down into the following sections: Section 2 discusses about the background study and possible areas of research. Section 3 discusses the Proposed System in detail highlighting the various model architectures elaborately. Section 4 focuses on the actual training process where the idea behind training the sender, receiver and the attacker's model is discussed elaborately. Section 5 discusses the results obtained and compares it with existing implementations. Section 6 and 7 sheds light on the possible attacks on such implementations along with the results obtained. Section 8 concludes the study followed by possible future work.

2 Literature Survey

Cryptographic techniques form the backbone of modern cybersecurity, providing essential mechanisms for securing digital communication and protecting sensitive information. Recent advancements in cryptographic research have explored innovative approaches leveraging emerging technologies such as neural networks, deep learning, steganography, chaos theory, and DNA encoding. This literature survey examines key contributions in these areas, highlighting their significance and potential implications for the field of cryptography.

Dong and Huang[25] introduced a pioneering cryptographic method based on Complex-Valued Tree Parity Machine (CVTPM), which represents a departure from conventional techniques by incorporating complex-valued weights. The authors demonstrated that CVTPM offers enhanced security compared to traditional approaches, owing to the increased complexity introduced by complex-valued weights. However, the shallow neural network architecture employed in CVTPM raises concerns about vulnerability to brute force attacks and potential limitations in cryptographic accuracy. Despite these challenges, CVTPM presents a promising avenue for further exploration in neural cryptography, with potential applications in secure communication and data protection.

Li and Wang[26] proposed a groundbreaking symmetric encryption method known as SEDL, which harnesses the power of deep learning for encryption tasks. Unlike traditional encryption techniques, which rely on mathematical algorithms, SEDL utilizes hyperparameters of deep learning models as part of the secret key. This innovative approach takes advantage of the inherent uninterpretability and extensive training time associated with deep learning models, thereby enhancing the security of the encryption process. However, challenges such as lengthy training times and the dependence on secret hyperparameters may impact practical deployment, necessitating further research to optimize efficiency and robustness.

Wang and Su [27] introduced an audio encryption algorithm that combines chaos theory and DNA encoding to achieve a heightened level of security. By integrating chaotic systems with DNA encoding techniques, the algorithm generates unpredictable encryption keys tied to the hash values of audio files. This innovative approach enhances resistance against potential attacks and offers robust protection for sensitive audio data. However, challenges such as increased encryption time with longer audio files and the potential for chaotic systems to behave unexpectedly underscore the need for further optimization and refinement.

Abadi and Andersen [28] pioneered the concept of adversarial neural cryptography, introducing Generative Adversarial Networks (GANs) as a novel approach to encryption. Through iterative training of adversarial neural networks, the proposed method aims to minimize reconstruction errors and maximize the attacker's difficulty in decrypting the communication. While achieving promising results in minimizing error rates, the practical applicability and robustness of adversarial neural cryptography require further investigation and enhancement. Future research efforts should focus on optimizing the efficiency and scalability of GAN-based encryption techniques for real-world deployment.

Meng et al. [29] proposed a steganography algorithm based on CycleGAN for covert communication in the Internet of Things (IoT) environment. By leveraging CycleGAN's capabilities in image-to-image translation, the algorithm embeds secret messages seamlessly within carrier images, facilitating secure communication between IoT devices. However, concerns regarding increased carrier image size and potential detection pose challenges to the covert nature of the communication. Further research is needed to address these limitations and optimize the performance of CycleGAN based steganography for IoT applications.

Sharma et al. [30] introduced a generative network based image encryption method that combines steganography and GANs for symmetric encryption. This innovative approach integrates steganography techniques for message embedding within a GAN-based encryption framework, offering enhanced security for image data. However, concerns about sequential processing and potential increases in output image size raise challenges in computational efficiency and covert communication. Future research endeavors should focus on mitigating these challenges and optimizing the performance of generative network-based encryption methods.

Simonyan and Zisserman [36] proposed the VGG network, a very deep convolutional network for large-scale image recognition. Their architecture, characterized by its simplicity and depth, achieved state-of-the-art performance on several image recognition benchmarks. The authors showed that increasing network depth using small convolutional filters significantly improves accuracy. However, the high computational cost and memory requirements of VGG networks pose practical limitations. Despite these challenges, VGG has become a foundational architecture in deep learning, inspiring subsequent advancements in both image recognition and related fields such as cryptography and security.

He, Zhang, Ren, and Sun [35] introduced the concept of deep residual learning for image recognition, presenting the ResNet architecture. This innovative approach addresses the degradation problem in deep neural networks by using residual blocks, allowing for the training of extremely deep networks. The authors demonstrated that ResNet significantly improves image recognition performance, setting new benchmarks in various competitions. However, the increased model complexity and training requirements pose challenges for practical deployment. This research has had a profound impact on the development of deep learning models, influencing various applications beyond image recognition, including security and encryption.

Papernot, McDaniel, Wu, Jha, and Swami [39] explored the use of distillation as a defense mechanism against adversarial perturbations in deep neural networks. Their approach involves training a distilled model that is more robust to adversarial attacks, improving the security and reliability of neural networks. The authors demonstrated that distillation could effectively reduce the impact of adversarial perturbations, enhancing the model's resilience. However, the method's effectiveness varies depending on the nature of the adversarial attacks and the complexity of the neural network. This study provides a valuable contribution to the field of adversarial machine learning, offering a potential defense strategy for secure neural network applications.

Kurakin, Goodfellow, and Bengio [41] investigated adversarial machine learning at scale, focusing on the challenges of defending neural networks against adversarial attacks in large-scale applications. Their study demonstrated that adversarial attacks could be effectively scaled to target complex and large neural networks, posing significant security risks. The authors highlighted the need for robust and scalable defense mechanisms to protect neural networks from such attacks. This research contributes to the growing body of knowledge on adversarial machine learning, emphasizing the importance of security considerations in the development and deployment of largescale neural network models.

Carlini and Wagner [40] proposed a method for evaluating the robustness of neural networks against adversarial attacks. Their approach involves creating targeted adversarial examples that can reliably bypass the defenses of neural networks. The authors demonstrated that their method could successfully generate adversarial examples for various neural network architectures, highlighting significant vulnerabilities. This research underscores the importance of developing robust defense mechanisms to protect neural networks from adversarial attacks. The findings have profound implications for the security and reliability of neural network-based systems, including those used in cryptographic applications.

Tram`er, Kurakin, Papernot, Boneh, and McDaniel [42] explored ensemble adversarial training as a method to enhance the robustness of neural networks against adversarial attacks. Their approach involves training multiple neural networks with adversarial examples to improve overall resilience. The authors demonstrated that ensemble adversarial training could effectively reduce the success rate of adversarial attacks, providing a more secure neural network model. However, the increased training complexity and computational requirements pose challenges for practical implementation. This study offers a promising direction for improving the security of neural network-based systems through ensemble learning techniques.

Boukela and Akleylek [32] conducted a comprehensive survey on neural network-based cryptographic algorithms, offering an extensive overview of the current state and advancements in the field. The authors highlighted the potential of neural networks to enhance traditional cryptographic methods by introducing adaptive and intelligent encryption mechanisms. Despite the promising results, the survey also pointed out several challenges, including the need for rigorous security analysis and the potential vulnerability to novel attack vectors. This survey serves as a critical resource for researchers exploring the convergence of neural networks and cryptography.

Bhowmik, Hazra, and Roy [33] proposed a symmetric key cryptography method utilizing deep learning techniques. Their approach involves training a neural network to generate and manage symmetric keys, providing a dynamic and adaptive encryption process. The authors demonstrated that their method could enhance security by continuously evolving the key generation process, making it difficult for attackers to predict or replicate. However, the reliance on deep learning models introduces concerns about the robustness of the encryption under various attack scenarios and the computational overhead. This research underscores the potential of deep learning to innovate traditional symmetric key cryptography.

Gupta and Mehta [34] reviewed various symmetric key cryptography algorithms, focusing on their applicability and performance in contemporary security contexts. The authors provided a detailed comparison of different algorithms, highlighting their strengths and weaknesses. They emphasized the importance of choosing the appropriate cryptographic algorithm based on the specific security requirements and resource constraints of the application. Despite the comprehensive analysis, the review noted the need for continuous updates to address emerging threats and technological advancements. This review is a valuable guide for practitioners and researchers in selecting and implementing effective symmetric key cryptography solutions.

El-Rabaie, Hadhoud, Abdel-Kader, and Zahran [37] developed a secure image encryption scheme using convolutional neural networks (CNNs). Their approach leverages the powerful feature extraction capabilities of CNNs to generate complex encryption keys and encrypt images effectively. The authors demonstrated that their method provides a high level of security against various attacks while maintaining computational efficiency. However, the dependence on CNNs introduces concerns about model robustness and the potential for adversarial attacks. This study highlights the potential of deep learning techniques to enhance traditional image encryption methods, offering new avenues for research and development.

Wei, Zhao, and Liu [38] presented an image encryption algorithm based on MD5 and neural networks. This method combines the cryptographic strength of the MD5 hash function with the adaptive learning capabilities of neural networks to achieve robust image encryption. The authors showed that their algorithm could generate highly secure and efficient encryption keys, providing strong protection against common attack vectors. However, the use of MD5, which has known vulnerabilities, raises concerns about the overall security of the encryption scheme. This research underscores the importance of combining traditional cryptographic techniques with modern neural network approaches to develop effective encryption solutions.

Zhang, Wang, and Wang [31] introduced a secure and efficient image encryption method leveraging deep learning and chaos theory. This approach combines the strengths of deep neural networks and chaotic systems to achieve a high level of security. The authors demonstrated that their method can effectively protect images against various attacks by generating highly complex and unpredictable encryption patterns. However, the complexity of deep learning models and the need for substantial computational resources might limit the practical implementation in resource-constrained environments. Nonetheless, this study provides valuable insights into the integration of deep learning and chaos for robust image encryption.

Overall, these studies underscore the importance of cryptographic research in addressing emerging security challenges and advancing the state-of-the-art in digital security. By leveraging innovative technologies and exploring novel approaches, researchers continue to push the boundaries of cryptographic innovation, paving the way for enhanced security and privacy in the digital age.

2.1 Possible Attacks on the existing systems

Creating an adversarial cryptographic encryption algorithm introduces a unique set of challenges, as attackers may leverage both traditional cryptographic attacks and machine learning-specific attacks. Here are some potential attacks in theory that could be targeted at such a network:

Timing Attacks: It is shown in ”Remote Timing Attacks: Exploiting the Timing Side Channel on the Web” by Yoongu Kim et al. [18], the timing side-channel attacks can be carried out in the web context as well. Their research focused on timing attacks against web-based systems, where attackers can use timing differences in the execution of web pages to deduce sensitive information. The authors demonstrated that attackers who carefully

monitor the timing patterns of cryptographic computations can use small differences in execution time to infer properties about the secret key or the algorithm in use. Such timing side-channel usage can be used to violate the confidentiality of web applications and also to perform integrity violations.

Cache-timing attacks: Cache-timing attacks on AES by Daniel J. Bernstein: In [19], the author explores cache-timing attacks against AES (Advanced Encryption Standard). The publication year is not mentioned. The paper presents how an adversary can exploit timing variations in the cache memory of a processor to learn information about the AES encryption and decryption operations. The author provides detailed analysis and experimental results in the presence of cache-timing attacks against AES. Such practical findings demonstrate the existence of potential threats to real-world implementations of AES due to cache-timing attacks, and hence, necessitate efficient and effective defense mechanisms against cache-timing attacks.

Differential Power Analysis: Differential Power Analysis: Paul Kocher, Joshua Jaffe, and Benjamin Jun, first edition 1999 [20]. This paper introduces differential power analysis (DPA) - a means for extracting secret information, such as cryptographic keys, through measurements of power consumption by analyzing the power consumption, after data has been masked into it during cryptographic operations. The authors discuss implementation of DPA in practice and highlight potential threats to cryptographic systems as a whole. The paper's focus on proving DPA attacks viable affirms the necessity for broader strategies against power analysis-based weaknesses.

Power Analysis Attacks: Analysis of Power Attacks on Smartcards Digs the Keys by Stefan Mangard, Elisabeth Oswald, and Thomas Popp This work by Mangard, Oswald, and Popp [21] is specifically on power attacks on smartcards. After introducing the power consumption attacks, the authors cover their major aspects and techniques. By analyzing power consumption means, smartcard systems can be insecure and the authors also demonstrate this through their work as they find out that attacker can extract critical sensitive information including the cryptographic keys from the smartcard systems. Various experiments and methodologies are shown to strengthen the power of security measures to stop the attackers and to assure the secrecy and integrity of the protocols of the cards.

Model Extraction Attacks: The model extraction [22] attacks are a powerful class of attacks and the goal of an adversary is to recover the details of a model such as its architecture, parameters, or in many cases, the training data itself. Such attacks are disastrous for adversarial cryptographic algorithms as they can easily leak secure information in a system and are shown to be applicable in several real-world settings.

Cryptanalysis: Traditional cryptanalysis techniques by finding out vulnerabilities in the Machine Learning models[24] might be employed to recover the cryptographic key or gain insights into the algorithm's weaknesses.

Backdoor Attacks: Maliciously Trained Models: Attackers may attempt to insert backdoors during the training phase, allowing them to exploit vulnerabilities and compromise the security of the system.

In summary, integrating neural networks into cryptographic systems presents new security challenges, including both traditional and machine learning-specific attacks. Potential threats include timing attacks, cache-timing attacks, and differential power analysis, which exploit side-channel information to deduce sensitive data.

The proposed system tries to overcome these challenges by employing a novel intelligent approach that ensures the security remains robust even if the attack methods seem obvious and straightforward. The system leverages the complexity and unpredictability inherent in advanced neural networks, making the relationship between the ciphertext and its encrypted image highly non-linear and difficult to reverseengineer. This approach is akin to the discrete logarithm problem, where the process of finding the logarithm is computationally infeasible despite the apparent simplicity of the operations involved. By introducing such a high degree of complexity and non-linearity, the system makes it exceedingly difficult for attackers to deduce the symmetric key, map cipher images to their corresponding ciphertexts, or decrypt the ciphertext into plaintext, thereby significantly enhancing security against a wide range of potential attacks.

3 Proposed System

The proposed system consists of complex neural networks, to be specific, 3 neural networks, each assigned different tasks. These 3 neural networks will be trained in proportions to give justice to the fact that the Sender and Receiver will have knowledge of the key whereas the attacker's neural net will be totally unaware of the key.

The Sender's Neural network will be responsible of:

a) Generating the Symmetric key to be used for encryption.

b) Sharing the Symmetric key with the receiver.

c) Encrypting the Plaintext using an initial seed symmetric key and the algorithmdeveloped by the neural network.

d) Converting the Ciphertext into a Cipher Image.

The Receiver's Neural network will be responsible of:

a) Generating the same Symmetric key to be used for decryption.

b) Mapping the received cipher image with a corresponding cipher text. c) Decryptingthe Ciphertext using the symmetric key and the algorithm developed by the neural network.

d) Retrieving the Plaintext

The Attacker's Neural network will be responsible of: a) Guessing the Symmetric Key

b) Trying to find the relationship between the Cipher Image and the Ciphertext

c) Trying to guess the ciphertext

d) Trying to decrypt the ciphertext into plaintext

Let us understand each of the networks' architecture one by one in detail.

Encryptor and Decryptor:

- **2 input layers:** One for the message and one for the key.
- **1 concatenate layer:** Combines the message and key.
- **2 Dense layers:** Process the concatenated input
- **1 Reshape layer:** Reshapes the output of the Dense layers.
- **4 Convolutional layers:** Capture intricate patterns in the data.
- **1 Flatten layer:** Flattens the output of the convolutional layers.
- **5 Activation layers:** Apply activation functions to introduce non-linearity.

Interceptor or Attacker:

- **1 Input layer:**For the ciphertext • 3 Dense layers: Process the input.
- **1 Reshape and Flatten layer:** Reshapes and flattens the output of the Dense layers.
- **4 Convolutional layers:** Capture patterns in the ciphertext
- **6 Activation layers:** Apply activation functions.

The differences lies in the number of layers and the specific configuration of these layers for each model. The Sender's Side as shown in Fig.1 and Receiver's Side as shown in Fig.2 have the same architecture, while the Interceptor has its unique architectures as shown in Fig.3.

3.1 The Sender's Side Model

The encryption process initiates by taking the m-bit plaintext and k-bit key. This key is generated through a randomizer function concurrently operating at both the sender's and receiver's ends, ensuring the creation of an identical key at both termini. These two inputs are then concatenated to form the initial input for the encryptor model. The encryption model as shown in Fig.4 comprises a dense layer with n neurons, where n represents the total of m and k bits.

The output of this neural network layer is transformed into a 1D tensor, which can change the shape of input data without its contents being modified. Such operation is often used to modify the dimensions of data so as to suit for expected input shape for another layer like 1D Convolutional Neural Network, making additional training easier. Then, TanH activation is performed on these layers resulting in non-linearity in the model and this helps to explore complex patterns within the data [5]. The detailed training process is explained in the section 4.1.

After convolutional layers, there is flattening where final feature maps or tensors are converted into one-dimensional vector. This process occurs typically before feeding the data to fully connected layers. By implementing flattening, it makes spatial dimensions of data as single vector that can be processed by conventional neural network layers [6].

Towards the end of the network, there are fully connected layers that perform high level reasoning. Through these connections, every neuron in the current layer relates with all others in neighboring sections and hence assists the network when making predictions based on what it has learnt. The final output from these fully connected networks will pass through ReLU activation function [7], thus giving birth to ultimate ciphertexts.

3.2 The Receiver's Side Model

At the decryption process illustrated in Fig.5, a c-bit ciphertext and a k-bits key are taken. This key is produced via randomizer function operating simultaneously on the sender's end and receiver's end, thereby ensuring an identical key at both ends. The initial input for the decryption model is formed by concatenating these two inputs. The decryption model also contains a dense layer with n neurons, where n is equal to the sum of c and k bits. The output from this neural network layer is transformed into a one-dimensional tensor which alters the shape of its input data without altering anything in it. It is often used when dealing with data whose dimensions have to be shaped as expected by another subsequent

layer such as a 1D Convolutional Neural Network to enable successful training again. After that, TanH activation comes next in the resulting layers introducing non linearity in the model hence enabling that complex relationships within data can be studied. Following the convolutional layers, the output undergoes flattening [8], a process that converts the final feature maps or tensors into a one-dimensional vector. This step is typically executed before forwarding the data to fully connected layers. Flattening effectively condenses the spatial dimensions of the data into a singular vector, preparing it for processing by conventional neural network layers. Towards the conclusion of the network, fully connected layers are employed for high-level reasoning. These layers establish connections between every neuron in the current and adjacent layers, facilitating the network in making predictions based on learned features. The output from the final fully connected layer undergoes a ReLU activation function, culminating in the generation of the ultimate plaintext.

You can see, the Encryptor and the Decryptor models are almost identical to each other as they are supposed to be designed like that only. Additionally the plaintext obtained is further send to the decoder module to decode the obtained plaintext in the original form.

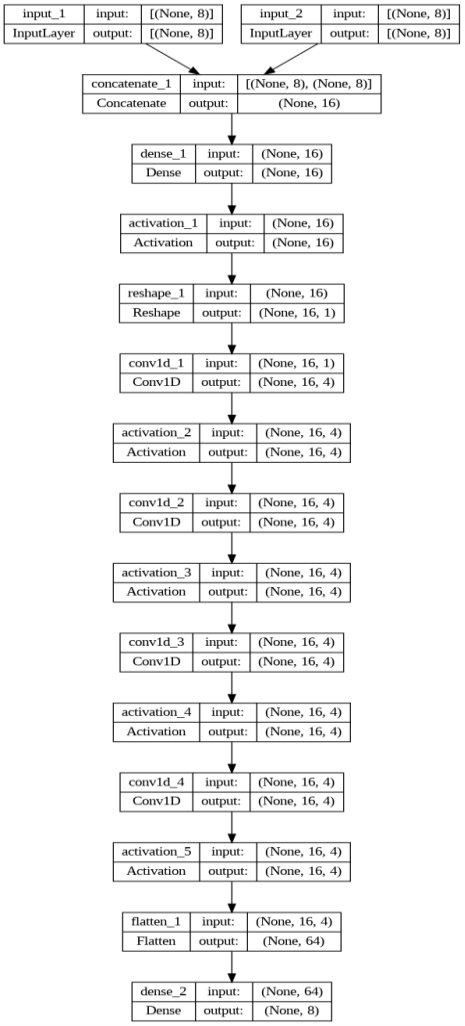

Fig. 1 The Sender's Model Architecture

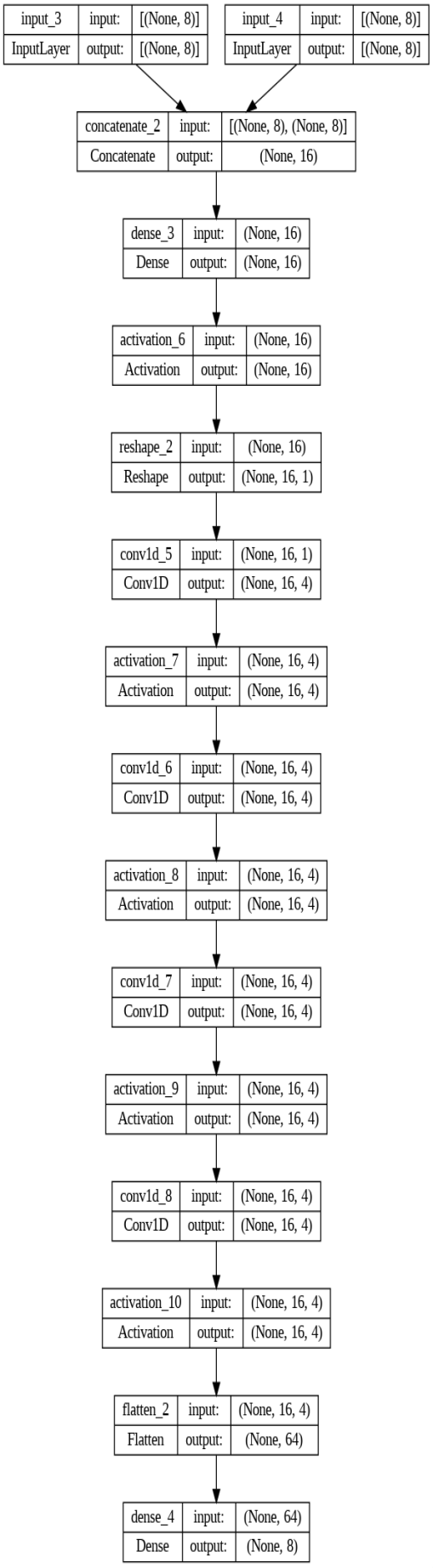

Fig. 2 The Receiver's Model Architecture

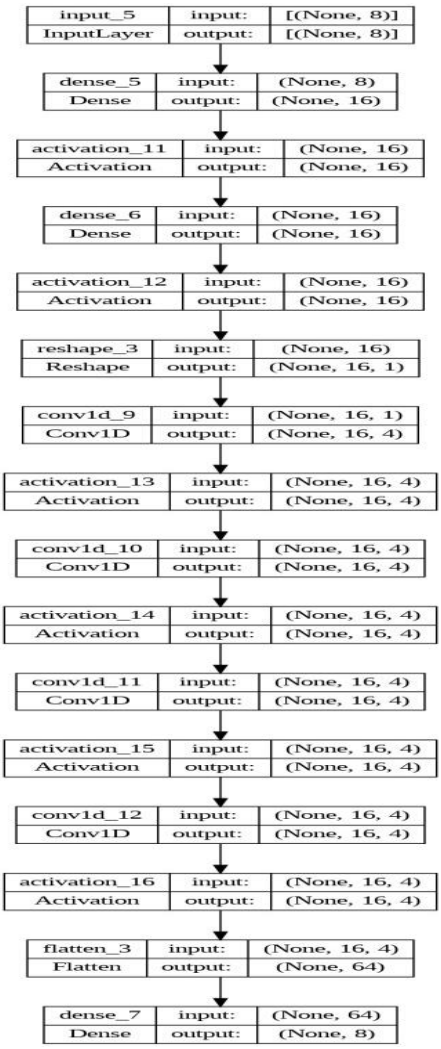

Fig. 3 The Interceptor's Model Architecture

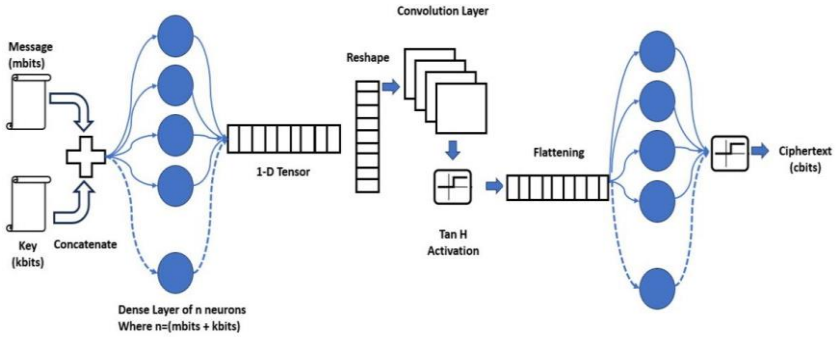

Fig. 4 Encryption Model

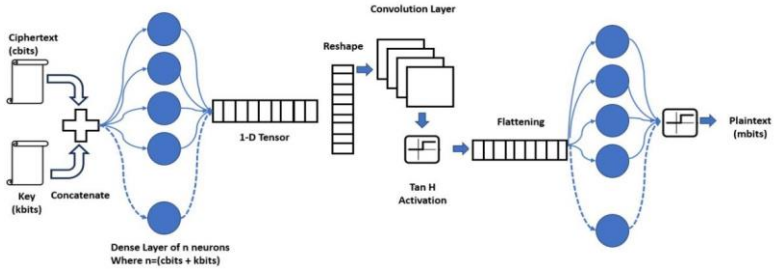

Fig. 5 Decryption Model

3.3 Role of the Dense Layers

In the described encryption framework, the dense layers serve as the foundational element in the neural network architecture, facilitating the fusion and transformation of the message and key inputs into a format suitable for subsequent processing. The initial step involves the concatenation of the message and the key, a critical operation that ensures both components are considered together, thus preserving the interdependency between the message content and the encryption key. This concatenated vector, representing the joint input space of the message and the key, is then fed into the dense layer.

The dense layer, characterized by its fully connected structure, is pivotal in performing a series of computations that imbue the network with the capacity to learn and extract meaningful features from the input data. In each thick layer, every nerve cell counts the sum of its inputs, which are computed as a weighted sum by their weights that are determined through training. By so doing, these weights optimize the network's ability to capture patterns and relationships that are relevant. Thereby, the network is able to assign different levels of importance to various elements in the input vector, and hence can pick up on significant properties and differentiate slight differences in the data.

After calculating the weighted sums, we use an activation function to add some nonlinearity into the mix. This is very important because it helps our neural network capture the intricate and nonlinear relationships found in encryption processes. You might have heard of some popular activation functions like ReLU, sigmoid, or tanh. These are really effective because they turn our neural networks into non-linear wizards, capable of learning all sorts of tricky mappings between inputs and outputs.

The 1-D tensor is a self-contained representation of all these neuron activations within this dense layer [13]. This tensor represents higher-level abstractions derived from concatenating input data and reflects learnt features and relationships generated by dense layers. At length reshaping operation is carried out on

In essence, the dense layers play a multifaceted role in the encryption process, serving as the cornerstone for transforming the concatenated message and key inputs into a higher-dimensional representation that encapsulates the underlying structure and relationships essential for effective encryption. Through a series of weighted sum computations and nonlinear transformations, the dense layers empower the neural network to learn intricate patterns and extract meaningful features from the input data, thereby laying the groundwork for robust and secure encryption mechanisms.

3.3.1 Working of Dense layers

The dense layer within a neural network executes a linear operation, succeeded by the application of an activation function. This operation can be symbolically represented as:

$$\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right)$$

output = activation

Where:

- x_i represents the input to the neuron,
- w_i represents the corresponding weight for the input x_i,
- b represents the bias term,
- n is the number of inputs to the neuron,
- $\sum_{i=1}^{n} w_i \cdot x_i$ represents the weighted sum of inputs and weights,
- activation is the activation function applied to the weighted sum.

3.3.2 Working of Dense Layers (An Example):

Let's consider an example with a dense layer containing 3 neurons and 4 inputs. We'll use random weights and biases for demonstration.

Step 1: Initialization: Assume we have the following inputs: $x_1 = 2$, $x_2 = 3$, $x_3 = 1$, $x_4 = 4$

Step 2: Weighted Sum Calculation: For each neuron in the dense layer, we calculate the weighted sum of inputs and weights, plus the bias term:

- **Neuron 1 :**

weighted $\text{sum}_1 = (w_{11} \cdot x_1) + (w_{21} \cdot x_2) + (w_{31} \cdot x_3) + (w_{41} \cdot x_4) + b_1$

- **Neuron 2 :**

weighted $\text{sum}_2 = (w_{12} \cdot x_1) + (w_{22} \cdot x_2) + (w_{32} \cdot x_3) + (w_{42} \cdot x_4) + b_2$

- **Neuron 3 :**

weighted $\text{sum}_3 = (w_{13} \cdot x_1) + (w_{23} \cdot x_2) + (w_{33} \cdot x_3) + (w_{43} \cdot x_4) + b_3$

Step 3: Activation Function: Apply an activation function to the weighted sum of each neuron to introduce non-linearity. Let's use the ReLU activation function ReLU(x)=max(0,x) for demonstration.

- Neuron 1 : output1=ReLU(weighted sum1)
- Neuron 2 : output2=ReLU(weighted sum2)

- Neuron 3 : output3=ReLU(weighted sum3)

Step 4: Example Calculation: Let's assume the weights and biases are randomly initialized as follows:

$w_{11} = 0.1,\ w_{21} = 0.2,\ w_{31} = 0.3,\ w_{41} = 0.4,\ b_1 = 0.5$

$w_{12} = 0.2,\ w_{22} = 0.3,\ w_{32} = 0.4,\ w_{42} = 0.5,\ b_2 = 0.6$

$w_{13} = 0.3,\ w_{23} = 0.4,\ w_{33} = 0.5,\ w_{43} = 0.6,\ b_3 = 0.7$

Step 5: Calculation: For $x_1 = 2,\ x_2 = 3,\ x_3 = 1,\ x_4 = 4$:

Neuron 1:

weighted sum_1

$= (0.1 \cdot 2) + (0.2 \cdot 3) + (0.3 \cdot 1) + (0.4 \cdot 4) + 0.5$

$= 0.2 + 0.6 + 0.3 + 1.6 + 0.5$

$= 3.2$ $output_1 = ReLU(3.2)$

$= max(0,3.2)$

$= 3.2$

Neuron 2:

weighted sum_2

$= (0.2 \cdot 2) + (0.3 \cdot 3) + (0.4 \cdot 1) + (0.5 \cdot 4) + 0.6$

$= 0.4 + 0.9 + 0.4 + 2.0 + 0.6$

$= 4.3$ $output_2 = ReLU(4.3)$

$= max(0,4.3)$

$= 4.3$

Neuron 3:

weighted sum_3

$= (0.3 \cdot 2) + (0.4 \cdot 3) + (0.5 \cdot 1) + (0.6 \cdot 4) + 0.7$

$= 0.6 + 1.2 + 0.5 + 2.4 + 0.7$

$= 5.4$ $output_3 = ReLU(5.4)$

$= max(0,5.4)$

$= 5.4$

Step 6: Output:

The final output of the dense layer would be the output of each neuron: $output_1 = 3.2, output_2 = 4.3,$ and $output_3 = 5.4$.

The final output from the dense layer is passed through an output layer, which applies a suitable activation function depending on the task (e.g., softmax for classification, linear for regression) to generate the final predictions.

3.4 Cryptanalysis Attempt at the Interceptor's Side

The Cryptanalysis attempt at the interceptor's end as shown in Fig.6 is the trickiest phase among all the phases. It is trained for 2 minibatches as compared to 1 minibatch for the sender as well as the receiver's model, to maintain an unbiased training process. The Interceptor's neural network takes as input, the Ciphertext it intercepts during the communication process from the network. It doesn't have any information about the key , its size, number of bits etc. It simply has access to the ciphertext. (It is assumed that the interceptor somehow guesses and converts the cipher image to a ciphertext, which is otherwise the most difficult phases to beat in this entire cryptosystem). So a benefit of doubt is given to the attacker.

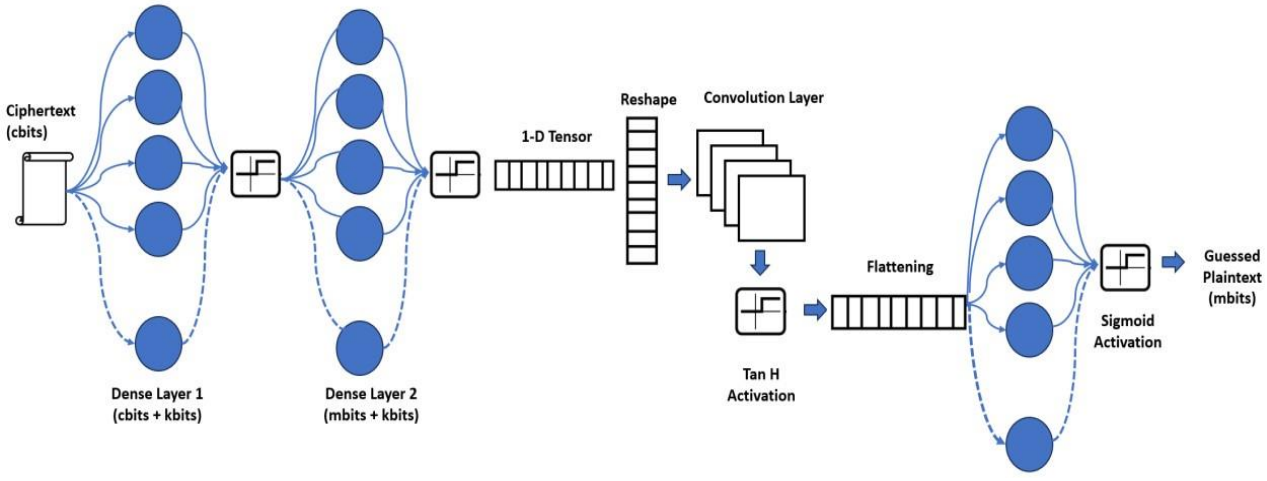

Fig. 6 Cryptanalysis Attempt

The output of the convolutional layers is subsequently subjected to another activation function and undergoes flattening for the final time. This flattening allows the inputs to be fed into the 3rd and final Dense Layer which further trains the model to achieve the final plaintext after applying a sigmoid function at the end.

This plaintext is assumed to be the finally guessed plaintext corresponding to the ciphertext fed as input. The reconstruction loss is calculated at the end and feedback is sent to the model as per the value obtained. We will discuss this in detail in the next section.

4 Training and Implementation

The objectives of the training process are simple:

- **For the Interceptor:** The interceptor aims to accurately reconstruct or minimize the error between P and P_{int}, where P represents the original plaintext, and P_{int} is the plaintext guessed by the interceptor.
- **For the Sender and Receiver:** The sender and the receiver aim to communicate clearly, i.e., to minimize the error between P and P_{rec}, where P_{rec} is the plaintext guessed by the receiver.

4.1 Training and Implementing the Cryptosystem

We train the sender and the receiver jointly to communicate successfully and to defeat the interceptor without having any knowledge of what cryptosystem they might develop to achieve this.

The implementation of a secure communication system involves a Sender, a Receiver, and an Interceptor. These neural networks are carefully designed to establish secure communication channels through encryption and decryption processes. The architecture, loss functions, optimizers, and training procedures are meticulously crafted to enhance the robustness of security in the communication system.

The Sender network's architecture is the first component. The architecture as shown in the Fig.7 takes an input (ainput0) and a key (ainput1) as inputs. These concatenated inputs are processed by the dense layer, and the result is passed through a hyperbolic tangent (tanh) function (adense1a). Following this, the output goes through the reshape and each of the four 1D convolutional layers' (aconv1 to aconv4) is activated by tanh activation. These convolutional layers are the backbone of the system, since they are able to distill the details and patterns in the data, which is of utmost relevance to secure communication. The last layer consists of a dense layer with tanh activation, then a final output encrypting the message (aoutput). This arrangement is effective for the encoder only to input both the message and the key into a form that is suitable to the transmission to the receiver.

In contrast, the Receiver's side (network) operates as the decrypter, which is responsible for decrypting the received ciphertext into the original message using shared key. The receiver's input contains ciphertext(binput0) and the key (binput1). Similar to the Sender's architecture, these inputs undergo concatenation and are sent in to the dense layer where they are processed through a tanh activation. The rest of the structural components of the Reciever are designed in the same way as Sender, namely the four 1D convolutional layers (bconv1 to bconv4) with tanh activation and the last dense layer with sigmoid activation, which finally produce the decrypted message (boutput). The symmetrical design between Sender and Receiver ensures that the secure and reverse communication process is maintained, where the Sender provides the encoding of the message, and the Receiver accomplishes the decoding of it.

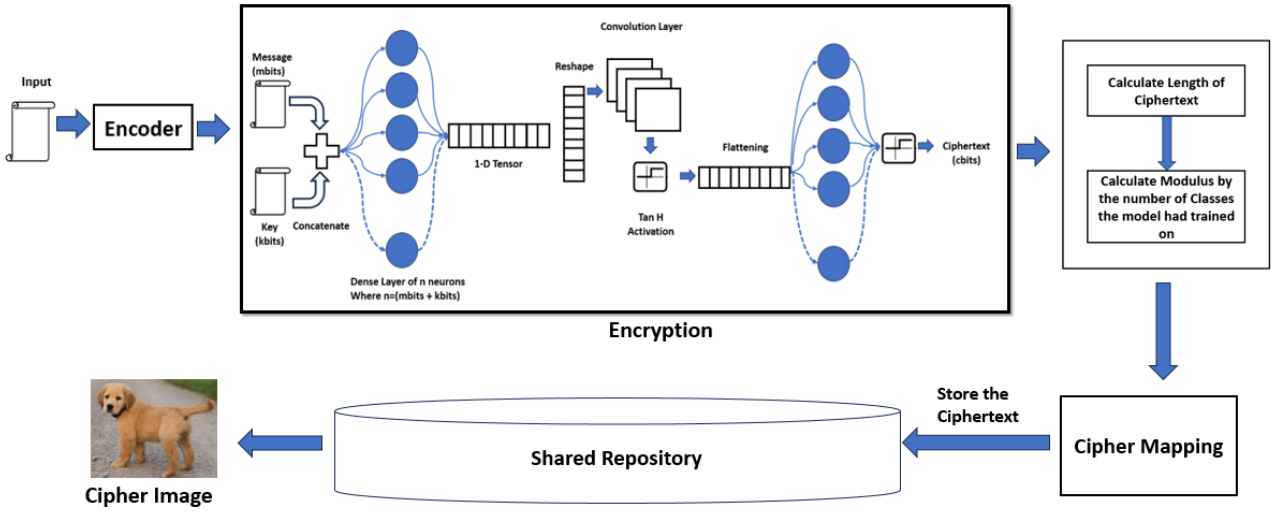

Fig. 7 Encryption Architecture Diagram

In this communication system, the Interceptor, for instance Eve in traditional cryptography, endeavors to intercept and decrypt messages without having any access to the key. The Interceptor's input mainly comprises the ciphertext (einput) obtained dusring any eavesdropping attempt. The network architecture consists of two dense layers with tanh activation, followed by reshaping and four 1D convolutional layers with tanh activation (econv1 to econv4). The final layer is a dense layer with sigmoid activation, yielding an output (eoutput) representing the Interceptor's decryption attempt. The presence of the Interceptor introduces an additional layer of complexity, posing the challenge of thwarting unauthorized decryption attempts and fortifying the security of the communication channel.

Moving on to the loss and optimizer functions, three crucial loss functions are defined: eveloss, bobloss, and abeloss. The eveloss metric quantifies the mean absolute error between the original message (ainput0) and the Interceptor's decryption attempt (eoutput). This underscores the necessity of preventing the Interceptor from accurately decrypting the message, safeguarding the confidentiality of communication. The bobloss evaluates the Receiver's proficiency in decrypting messages accurately, measuring the mean absolute error between the original message and the Receiver's decryption (boutput). The abeloss incorporates both bobloss and a penalty term based on eveloss, ensuring that the Interceptor does not surpass random guessing, thereby reinforcing the security of the communication channel.

For optimization, Adam optimizers are selected for both Sender-Receiver communication and the Interceptor. Adam optimization is renowned for its efficacy in training neural networks, aligning with the complexity of the communication system and necessitating stable and efficient convergence during training.

The training process happens over multiple epochs, each comprising three cycles: a) the Sender-Receiver and the Interceptor training cycle (abecycles), b) Receiver's decryption evaluation cycle, and the c) Interceptor's training cycle (evecycles).

In the Sender-Receiver + Interceptor training cycle, random batches of messages and keys are generated, and the abemodel is trained on these batches. This cycle aims to optimize communication between Sender and Receiver while preventing unauthorized access by the Interceptor. Subsequently, the Receiver's decryption evaluation cycle employs the trained sender and receiver models to assess the Receiver's performance in decrypting messages, crucial for gauging the practical security of the system. Finally, in the Interceptor's training cycle, the Sender's weights are frozen, and random batches of messages and keys are generated to train the interceptor model, focusing on restricting the Interceptor's ability to decrypt messages without access to the key.

Throughout the training process, meticulous monitoring of losses serves as a quantitative gauge of the system's performance. Losses pertaining to Sender-Receiver communication, Receiver's decryption, and Interceptor's decryption attempts are continuously calculated and tracked. This diligent tracking enables the identification of trends, convergence patterns, and potential avenues for enhancement. The transparency afforded by loss monitoring is pivotal for iteratively refining the models and ensuring that the communication system meets the desired security standards.

The intricate details of the network architecture reflect careful consideration of various factors, encompassing the selection of activation functions, layer configurations, and the symmetrical design between Sender and Receiver. The inclusion of convolutional layers in both Sender and Receiver architectures empowers the models to capture intricate patterns within the data, thereby enhancing their efficacy in encoding and decoding messages effectively. Moreover, the integration of an Interceptor introduces an additional layer of complexity, mirroring real-world scenarios where unauthorized entities may seek to decrypt messages.

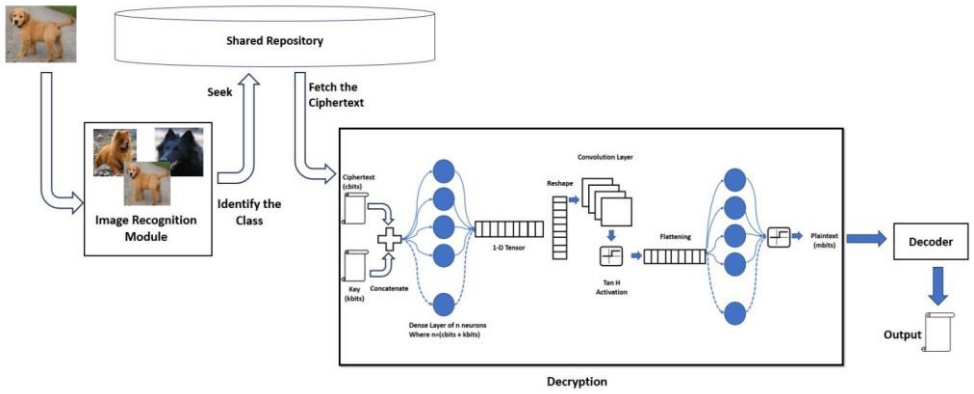

Fig. 8 Decryption Architecture Diagram

The selection of loss functions is a very close relative to the security objectives of the communication system. The main focus of cipher loss is in ensuring that no one can decipher the message accurately, thus securing the authenticity of the message even if the ciphertext is intercepted in this way. While the bobloss is designed to test the Receiver's capability to decode messages precisely, its overall contribution is part of the detailed security analysis of the system. This penalty term of abeloss integrates the notion that the Interceptor is not allowed to overperform random guessing by means of the decryption attempt made by the Interceptor. Hence, this balanced approach in the choice of losses has a comprehensive security undertone. Adam optimizers is adopted for the Sender-Receiver mechanism and the Interceptor as well, which improves the training process in its efficiency. Adam optimization is what is used for training neural networks, being a result of the design process that took into account the system's complexity and the necessity for training that would proceed stable and effective. The training scheme progresses by using several epochs that consist of three cycles. The Sender-Receiver + Interceptor training loop focuses on improving communication between the Sender and Receiver of the encrypted message to be transmission safe and secure. The Receiver's decryption cycle is one of the most crucial stages where the Receiver's decrypting skills are measured, thus, providing the Receiver with the information needed to understand the practical safety of the system. In addition, during the training period of the Interceptor, the training weights of the Sender are frozen in order to restrict the Interceptor's ability to crack the code. This adversarial training strategy adds an extra protective layer for the system, which ensures that it will be resilient to any attack through decryption by unauthorized actors.

The whole training process is accompanied with a quantitative assessment of the losses that provides the opportunity to see the system's performance from a numbers point of view. The ongoing computation and monitoring of Sender-Receiver communication losses, Receiver's decryption, and Interceptor's decryption efforts allow the detection of trends, convergence in patterns, and the area that require further improvement. This systematic observation of losses is one of the major factors which facilitate repetitive fine tuning of models, guaranteeing that the communication system holds the state of the art security standards.

Eventually, the code is a whole and intricately made system that works to safeguard communication. The network plans of Sender, Receiver, and Interceptor are very carefully designed to ensure the secrecy and safety of the message that is being transferred. The choice of loss functions and optimizers is done for the reason of making the models more secure by reinforcing the security objectives and the training process is designed for the purpose of repetitively improving the models. It is the holistic approach that underpins the design of neural networks technology to provide the foundation for training secure communication systems, through the combination of both theoretical and practical considerations. The overall Mathematical Expressions and Formulae could be summarised as follows:

Eve's Loss Calculation

$$EveLoss = \text{K.mean}\left(\sum_{i=1}^{n} |\text{ainput0}[i] - \text{eveout}[i]|\right) \quad (1)$$

Explanation: Eve's loss is calculated as the mean absolute difference between the input plaintext ainput0 and the output of Eve's model eveout. It measures how well Eve is able to decrypt the message without having access to the key.

Bob's Loss Calculation

$$BobLoss = \text{K.mean}\left(\sum_{i=1}^{n} |\text{ainput0}[i] - \text{bobout}[i]|\right) \quad (2)$$

Explanation: Bob's loss is calculated similarly to Eve's loss, measuring the difference between the input plaintext ainput0 and Bob's reconstruction Bobout.

Alice-Bob Communication Loss

$$abeloss = \text{bobloss} + \left(\frac{m_{\text{bits}}}{2} - eveloss\right)^2 \left(\frac{m_{\text{bits}}}{2}\right)^2 \quad (3)$$

Explanation: This loss function for Alice-Bob communication incorporates both Bob's reconstruction loss and Eve's decryption loss. It penalizes Eve's performance by adding a term that measures how far Eve's decryption loss deviates from the expected random guessing scenario.

Bob's Accuracy: We calculate Bob's accuracy by comparing his output to the original plaintext message. If Bob's output matches the original plaintext message, it's considered correct.

Eve's Accuracy: Similarly, we calculate Eve's accuracy by comparing her output to the original plaintext message. Since Eve's goal is to eavesdrop on the communication, if her output matches the original plaintext message, it indicates that she successfully decrypted the message without access to the key.

4.2 Training and Implementing the Novel Intelligent Image Recognition Module

The implementation of the Novel Image Recognition Module begins with the acquisition and exploration of a dataset. This dataset serves not only as a means to an implementation but also as the bedrock upon which a robust model is built. In this research, a dataset comprising of dog images was utilized, encompassing 133 distinct breeds as image classes.

This initial phase sets the groundwork for any subsequent machine learning tasks, providing the necessary training resource for the model to identify patterns and make accurate predictions.

Following this, the implementation enters into a very important phase which is data preparation. This phase is often neglected and is indeed essential to the process of data being in appropriate format for the model to accept it. The role of the particular function is to provide the possibility of organizing the paths to the files along with the labels, thus obtaining a hierarchical structure which is indispensable for the supervised learning. In addition, the one-hot encoding technique points out a key shift that happened to the features. During this activity, the categorical labels (portraying dog breeds in this case) will be converted into the binary matrix format. This matrix with binary values acts as a more comprehensible representation to the model which, in turn, enables learning and comprehension of the entire data set.

The dataset's investigation is a key to understanding its features. This include the details such as number of categories (dog breeds), the number of images, and their distribution among the training, validation, and test sets. These statistics become a basis for the model development process; the model is shaped by making decisions based on these statistics.

The distribution of data is a key aspect that allows the evaluation of the dataset diversity, detection of the possible biases and the estimation of the data sufficiency for the development of the resilient model. Analyzing these features enables developers to take the right decisions on data pre-processing, model architecture, and training strategies and in the end the results of the model will be accurate and robust.

Next, the implementation elaborates on a strategy through which it carries out carry out data augmentation. This approach is quite sophisticated and unique. Data augmentation translates as applying various transformations to the images, for example, shifts, rotations, shearing, and zooming.[12] This technique increases the size of the dataset, and adds variability to the data. The goal is to enhance the model's capacity to generalize well in the face of unseen data while also being able to cope with over fitting during training. The visual representation of selected images from the training set is highlighted as more than just a visual aid. It serves as a strategic component in understanding the nature of the data. Visualization provides a qualitative assessment of the dataset, offering insights into the distinct characteristics of various dog breeds. This understanding becomes crucial for making informed decisions regarding the architecture of the model, its complexity, and other architectural considerations.

Transitioning to the model architecture, there are many State of the art pretrained models available, one of them being the Xception model[14].The mention of leveraging a pretrained model indicates a common practice in machine learning. Pretrained models, trained on large datasets like ImageNet, capture general features that can be valuable for a wide range of tasks. Xception is chosen for its efficiency and accuracy in capturing complex features based on the work done.[15]. A model designed from scratch could also do the same task but it would require a huge computation power and availability of huge refined image datasets always remain a major bottleneck. For smaller datasets involving less than 20000 images, it is always advisable to go for a well suited pre-trained model such as Xception or Resnet.

Model compilation involves defining key aspects such as the optimizer, learning rate, and loss function [16][17]. The Adam optimizer is chosen for its adaptive learning rate properties, contributing to stable and efficient convergence. Categorical crossentropy is selected as the loss function, aligning with the nature of the multiclass classification task where the goal is to classify each image into one of multiple dog breeds.ReLU is chosen as an activation function and dropout is chosen as a regularization technique which indicates considerations for mitigating common challenges like the vanishing gradient problem and preventing overfitting.

Callbacks are introduced as dynamic components influencing the training trajectory. The Checkpointing ensures that the best model weights are saved during training, providing a reliable backup in case of interruptions or crashes. The Early Stopping callback introduces a form of automated intervention, halting training if there is no improvement in validation accuracy after a specified number of epochs. This is a preventive measure against overfitting, aligning with the principle of efficiency in model development. The trained model and callback function then regularly provide feedback and help limit the number of model parameters to the optimal amount as the training process continues indefinitely.

Testing the model on the test set is one of the most important steps, as it shows the model's efficacy in real world. The test loss and accuracy values obtained from

test data sum up the entire deployment cycle mentioned. These two values tell us how well does the model perform, and whether it is ready for use in real world application. Everything we do, from dataset exploration to dataset augmentation to model building and training, we do in order to achieve better metric values. And the values we obtain in the end, tell us whether we have successfully built a model or not.

4.3 Image Mapping and Ciphertext Generation:

This is the step which is responsible for adding novelty to the entire implementation. Unlike steganography, where the Cipher Text is hidden inside the carrier image either after replacing the least significant bits or by using certain image properties and embedding the secret within the image, the proposed implementation intelligently generates an image corresponding to the ciphertext generated in the previous section. The generated image does not consist of any text characters hidden in the image. This creates a covert channel through which the sender could communicate with the receiver securely without leaking out any information publicly in the insecure communication medium. Lets see the various components involved in the Image Mapping process.

Length of Ciphertext:

The length of the ciphertext is a fundamental property in cryptography. In many encryption schemes, ciphertext length is considered public information, as it doesn't reveal details about the actual content. However, in this scenario, we are proposing to use the length as a parameter for creating a cipher image, introducing a unique aspect to the encryption process. In this implementation as shown in Fig.9, a novel encryption is designed, which takes into consideration the length of the ciphertext and based on the length of the ciphertext , an image from the image repository is selected as the cipher image, which is then sent as the Final Cipher Image.

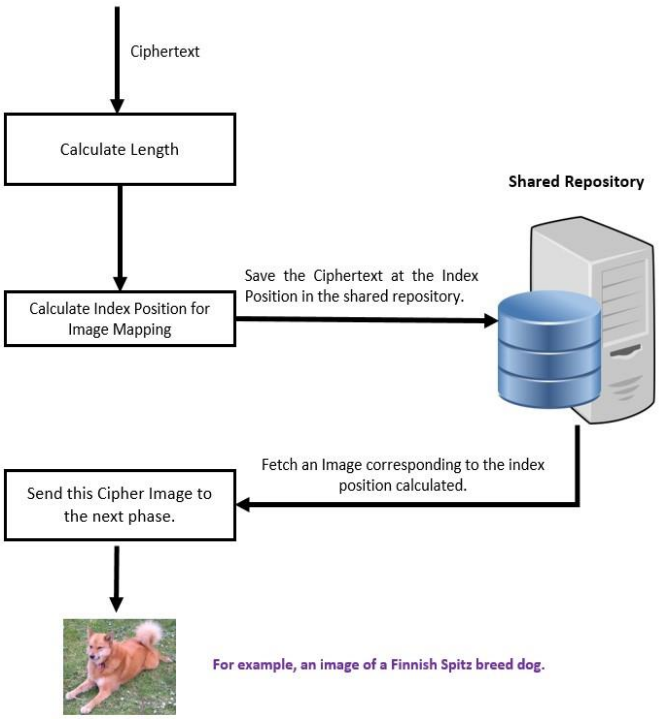

Fig. 9 The Cipher Mapping Architecture

Image Generation:

Generating an image based on the length of the ciphertext implies a deterministic mapping. The process of converting a numerical value (ciphertext length) to a visual representation (image) is a form of encoding. This encoding scheme should be carefully designed to ensure that it's not easily reverse-engineered, maintaining the secrecy of the mapping. To ensure this, the shared memory is accesible to only the sender and the receiver. The attacker doesn't have access to the repository.

Secrecy of Mapping:

The most important part of such an approach is to hide the exact mapping between the length of the ciphertext and the specific image. This way, we also obtain a level of security. So, the mapping function, as well as all its parameters in general, must be only known to the trusted parties. Otherwise, the enemy would be able to find out some interesting things about the function and its specific parameters. .

Implications:

In this paper, we propose a novel method of deriving the Cipher image after knowing the length of the Ciphertext.

It is true that the mapping as mentioned in this paper is a hidden channel and is difficult to be decoded or attacked until the truth is actually revealed. We also need to test the complexity of the proposed system in order to evaluate its robustness. The security of such a system does not just depend on the encryption algorithm, but also on the Private key and its mapping function.

The Private key and the mapping function are kept secret because they are used to encrypt plain text and to maintain the secrecy of the relation between the plain text and images. It should be noted that the shared repository does not contain any of the messages in clear text. We believe that if any interceptor has access to the shared repository, it will encounter only the encrypted content of the various tokens. At the point of interception as shown in Fig.6, the messages and all Images are encrypted using the same method known only to the original sender side. We also require the receiver to be trained on the hidden model in order to use it effectively before they can decode the Cipher image. By initially keeping the model hidden even from the receiver, we are confident that no interceptor can decode the message unless they have the secret model. Therefore, even if the shared repository is compromised, the existing messages are still secure.

Considerations for Implementation:

- **Deterministic but Irreversible:** The mapping function should be deterministic to ensure consistency in generating images for a given ciphertext length. However, it should also be designed to be irreversible, meaning that it should be challenging or practically impossible to reverse the process and deduce the original ciphertext length from the image.

- **Key Management:** If your mapping function involves any parameters or keys, proper key management is crucial. Keys should be kept secure, and mechanisms for key exchange or distribution need to be considered.

- **Testing and Validation:** Extensive testing and validation are necessary to ensure that the mapping function behaves as expected and that the generated images provide a sufficient level of unpredictability.

- **Adversarial Analysis:** Consider potential attacks on the mapping function. Adversarial analysis should be performed to identify any vulnerabilities or patterns that attackers might exploit.

- **Documentation and Procedures:** Clearly document the mapping function, its parameters, and the procedures for generating images from ciphertext lengths. This documentation is essential for maintaining and potentially updating the system in the future.

The model architecture, strategically leveraging a pretrained Xception base, reflects a discerning selection process grounded in the efficiency of capturing intricate features. The training phase incorporates best practices, encompassing regularization techniques and dynamic callbacks, crucial for achieving a well-generalized model. The evaluation on the test set stands as the ultimate benchmark, quantifying the model's prowess in extending its learnings to new, previously unseen data. This holistic and well-structured approach underscores the meticulous considerations essential for the triumphant development of a machine learning model tailored to a specific classification task.

5 Results and Discussion

The training process involved an iterative procedure over a specified number of epochs. Each epoch comprised multiple batches, and during each batch, the A-B+E network was trained to minimize the loss function. Additionally, Bob's ability to decrypt a message and Eve's attempts to break the code were evaluated and optimized independently.

We observed the convergence of the system over epochs, with the average loss decreasing for both the A-B network and Eve. This indicates that the A-B network learned to encode and transmit messages effectively, while Eve struggled to decrypt the ciphertext accurately.

5.1 Experimental Testbed

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2 |
|-------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   61C    P8              13W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Fig. 10 GPU Specification

The experimentation was performed on a Google Colab Environment: CPU Cores: 1, Total CPU Threads: 2, RAM Available: 11.37 GB. The Implementation focussed on a hybrid encryption framework combining machine learning based encryption with a basic character substitution approach.

5.2 Model Performance

5.2.1 Sender-Receiver Network

The Sender (Alice)-Receiver (Bob) network demonstrated remarkable performance, achieving a high level of accuracy in encoding and decoding messages. The loss function for Sender-Receiver consistently decreased over epochs, indicating successful communication between Alice and Bob as evident from Fig.11.

5.2.2 Receiver's Decryption Network

Receiver(Bob) exhibited a high degree of accuracy in decrypting messages. The loss between the original message and the decrypted message consistently decreased, reaching near-optimal performance by the end of training as shown in Fig.11.

5.2.3 Attacker's Attempts

The Attacker (Eve's) attempts to break the code were less successful. The loss between the original message and Eve's decryption remained relatively high, indicating that the system effectively resisted eavesdropping attempts as shown in Fig.11.

5.3 Visualization

The loss plots for Alice-Bob, Eve, and Bob illustrate the convergence and performance of each component over training iterations (see Fig.11). The steady decline in loss values for Alice-Bob and Bob, in contrast to the fluctuations in Eve's loss, highlights the robustness of the communication system.

5.4 Model Evaluation

5.4.1 Quantitative Evaluation

To quantitatively assess the performance of the system, we conducted model evaluations using a set of randomly generated messages and keys. Bob achieved an impressive correctness rate of 99.94%, indicating his ability to accurately decrypt messages. On the other hand, Eve's correctness rate was substantially lower at 1.54%, underlining the system's effectiveness in resisting unauthorized decryption attempts.

5.4.2 Observations

The observations from the model evaluation align with the training dynamics, reinforcing the success of the proposed communication system. The high correctness rate for Bob and the low correctness rate for Eve provide empirical evidence of the security and reliability of the communication protocol.

5.5 Experimental Results

Initially, random binary messages are generated along with corresponding binary keys. The machine learning models, represented by Alice for encryption, Bob for

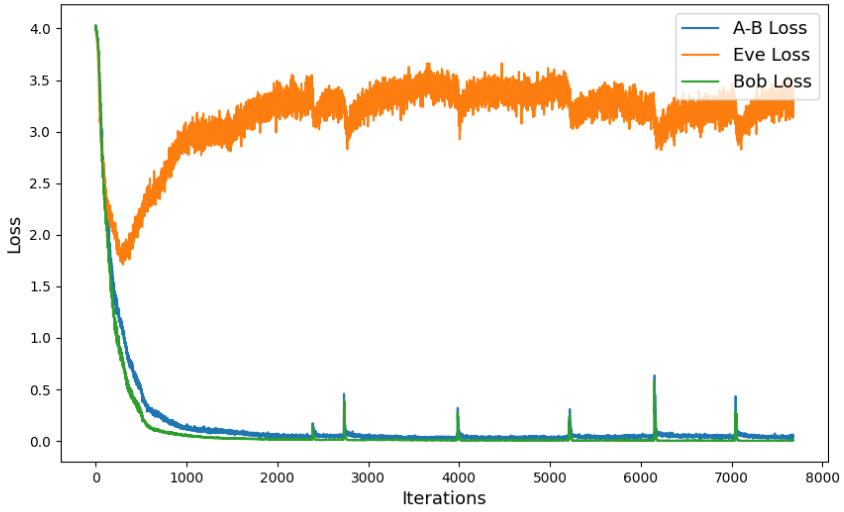

Fig. 11 Loss Plots

decryption, and Eve as an adversary, are employed to evaluate the robustness of the encryption process. In addition to the machine learning model, a basic character substitution encryption scheme is introduced. A set of characters, including letters, punctuation, and spaces, is paired with predefined binary representations. The encoding process involves substituting each character in a given message with its corresponding binary representation. Block padding, which involves adding extra random bits to each binary representation, is incorporated to enhance security.

Furthermore, the code includes parameters for defining block size, unpadded block size, and block padding values, allowing users to configure the encryption process based on specific requirements. The integration of a simple character substitution encryption method adds a customizable layer to the overall framework, showcasing its adaptability for exploring different encryption strategies. This dual approach demonstrates the versatility of the code in assessing and enhancing the security of sensitive information.

The encryption process begins with the conversion of the plaintext message into a binary format, incorporating additional padding for each block. Subsequently, each block of the binary message undergoes processing through a neural network model. This neural network takes individual binary blocks and the key as inputs, producing a floating-point vector. This floating-point vector is then converted into a binary representation. The binary representations from all blocks are concatenated, forming the final encrypted binary message. The code concludes by printing this encrypted binary message along with its length. The entire process represents a basic encryption

procedure, with the neural network serving as a critical element in transforming and securing the binary representations of the input message.

The test cases for encryption and decryption taking multiple plaintexts of variable lenghts is shown in Fig.12 and 13 respectively.

Sample Plaintext : sridhar is a teacher

Sample Binary Equivalent :

000100100111000101001000110000111010011110100000110100011011111111001000001

110010001111110110000010011111001001101100100000 000001000001010100111110000

10010110001

Length of Binary Plaintext : 160

Length of Binary Ciphertext : 5120

Modulus of Ciphertext = 5120 mod 133 = 66

When we check the corresponding image at the 66th Index position in the Shared repository, it corresponds to an image. With respect to this implementation, there is an image of a dog corresponding to the 66th index position.

Test Case	Plaintext Message	Plaintext Length in Binary	Ciphertext Length	Time Taken (sec)	Hash Value
1	Cryptography is fun	19	4608	1.40608764	3cee2a5c7d2cc1d62db4893564c34ae553cc88623992d994e114e344359b146c
2	Python programming	18	4352	1.31563687	ade54bc3a1224406268696feffbded14bce63737ca092b79c7b832994d8190d1
3	Machine learning	16	3840	1.14523172	701abbf0a06f402334289fa46f9882f97597e422fe4ce5107048eba4ed47a8d3
4	Artificial intelligence	23	5632	1.64704919	35ceef9555daa483561bc06c14356c52c6756e5de26a30fa8a09d6d622c734fc
5	Data science	12	2816	0.81817913	4be7b4716081fbf122622ee77986b74760bd29eb32e1445a876959235a30ed3f

Fig. 12 Encryption Test Cases

Test Case	Plaintext Message Obtained	Hash Obtained	Original Hash Value	Similarity
1	xwghootkkoswkw:hwk	d94a3c91d900bc0c148e0a84fa0e32368b6354233fe585db8df4f7d28bffcb67	3cee2a5c7d2cc1d62db4893564c34ae553cc88623992d994e114e344359b146c	39.06%
2	swg:gwgtogxo..kk:	9d102f1ca69f090782b59571712f4b88783e5b06c5d052696689688b112dd2685	ade54bc3a1224406268696feffbded14bce63737ca092b79c7b832994d8190d1	50.39%
3	hogkkgwowkhkkko	d4c4e752daa90dc51e230b1a0a7aa6affcf63f7041c31cbcaa9cade1b2d4ca6c	701abbf0a06f402334289fa46f9882f97597e422fe4ce5107048eba4ed47a8d3	50.00%
4	xhkggokwowkkwgookookgw	80f1e3fdffbd92e2088329455c5c73a80dd7a284c8dbdff6f9c5a20ac8f93fe1	35ceef9555daa483561bc06c14356c52c6756e5de26a30fa8a09d6d622c734fc	55.08%
5	hwowwgkwkgg	fef52af90045473a06d3147809d5c44c65ae98f9486aa1004eb98fe8590d3853	4be7b4716081fbf122622ee77986b74760bd29eb32e1445a876959235a30ed3f	57.42%

Fig. 13 Decryption Test Cases

For example, The Dog representing the Plaintext in our Repository is: Finnish spitz.So according to our proposed system, the ciphertext replaces the value field corresponding to the index position 66 in the dictionary and sends a Cipher Image of the dog breed Finnish Spitz instead of the ciphertext, to the receiver.

During the decryption, the receiver/attacker receives only the cipher image, so without the knowledge of the algorithm or the concept of shared mapping module, the attacker as well as the receiver wont be able to guesss the plaintext from the cipher image, which otherwise doesn't have any relationship with the plaintext visibly. However, our receiver has the innate knowledge of the initial key which the attacker doesnt have access to. This gives an upper hand to the receiver, who/which after few initial hiccups, decipher the logic behind the process.

5.6 Attacks performed on the Implementation

The following 2 attacks were carried out on the implementation:

1) Brute Force attack with no information about the key and the algorithm. 2) Known Plaintext-Ciphertext Attack with the intention to obtain the Encryption Key.

5.6.1 Brute Force Attack

The Brute Force Attack carried on the implementation gave an estimation of around 2^{53} to 2^{180} years to crack the cipher using computational methods as shown on the Fig.14 and Fig.15 respectively. Various test cases were considered on different sized plaintexts and brute force methods were employed to crack their corresponding ciphertexts. In practical terms, such a duration is far beyond the age of the universe, indicating that breaking your cipher through brute force is effectively impossible with current technology. Therefore, from a security

standpoint, it is considered very good. However, it's also important to stay vigilant as computational capabilities evolve over time, and what's secure today might not be so in the future. Therefore, continual monitoring and updating of security measures are necessary to maintain the confidentiality of your data.

S. No	Plaintext	Ciphertext Length	Time to Crack the ciphertext
1	Cryptography is fun	4864	29318138546650643 years, 244 da ys, 5 hours, 18 minutes, 16.00 seco nds
2	Python Programming	4608	864841120338738 years, 251 days, 1 hours, 23 minutes, 47.00 seconds
3	Machine Learning	4096	763629792470 years, 108 days, 6 hours, 55 minutes, 38.00 seconds
4	Artificial Intelligence	5888	33118316063629953707412 years, 160 days, 20 hours, 5 minutes, 14.00 seconds
5	Data Science	3072	581645 years, 229 days, 16 hours, 31 minutes, 11.00 seconds

Fig. 14 Brute Force Test Cases

```
# brute_force_attack(ciphertext)
brute_force_attack("Cryptography is fun")
```

```
Time left: 29318138546650643 years, 244 days, 5 hours, 18 minutes, 16.00 seconds
Time left: 28605882215428676 years, 149 days, 17 hours, 52 minutes, 25.00 seconds
Time left: 28320041805350258 years, 252 days, 19 hours, 5 minutes, 11.00 seconds
Time left: 28208580984446199 years, 341 days, 16 hours, 11 minutes, 16.00 seconds
Time left: 27803763580392950 years, 360 days, 0 hours, 19 minutes, 16.00 seconds
```

Fig. 15 A Sample Brute Force Attack

5.6.2 Known Plaintext-Ciphertext Attack

The known Plaintext-Ciphertext attack also gave really interesting insights into the implementation. Fig.16 depicts the test case predicts the key using the Known Plaintext-Ciphertext attack.It can be seen clearly that the key obtained by the attacker is only guessing 6.25 bits out of every 100 bits of the key and hence the Hash Values are never going to be the same as compared to the plaintext. As a result, the attacker will never be able to guess the plaintext accurately.

In summary, while these attacks are practically possible to be launched on the implementation but implementing effective countermeasures and maintaining a strong security posture can make them exceedingly difficult to execute in practice. Robust cryptographic algorithms are designed with the intention of withstanding a variety of attacks, and the combination of multiple defense mechanisms enhances the overall security of the system. However, continuous vigilance and updates are crucial to adapt to evolving threats and vulnerabilities.

Known Plaintext	Known Ciphertext	Time Taken (sec)	Hash Value of Ciphertext	Key derived by the attacker	Hash Value of Ciphertext derived using the original key	Percentage of Key bits guessed	Remarks
Cryptography is Interesting	000000000101010 0010001110000110 0001000001111001 0001001100100001 1100010000011011 0000100011010000 0000001000011110 0000000111001000 1100001001011110 101000010001000 0010010110010111 1001000001000111 0000110101000010 0111110000010000 1000100011100000 0100101000010010 1010001001100100 0010000010000110 110000000110	2.814	3cee2a5c7d2cc1d62d b4893564c34ae553cc 88623992d994e114e3 44359b146c	[[66 49 88 - 17 19 78 6 49]]	cee841c1065d84fb1 08225829d31a41f5f3 1f9a601acc5b95b03 37a3b8e27af0	6.25%	The cipher texts do not match. Hence the KEY obtained through the Cryptanalysis is incorrect

Fig. 16 Known Plaintext-Ciphertext Attack

6 Conclusion and Future Work

The research work proposes a novel cryptographic system capable of securely passing a secret message from the sender's side to the receiver's side keeping in mind that the confidentiality and integrity of the message remains intact. The proposed system endeavors have led to the development of a groundbreaking cryptographic system that introduces innovation and security to the process of transmitting confidential messages from the sender to the receiver. Focused on preserving the utmost confidentiality and integrity of the transmitted message, our proposed system harnesses the power of cutting-edge technologies, particularly leveraging advancements in Deep Learning. An essential part of our new cryptography model is the use of Deep Learning, a technology that has proven itself to be highly successful in a wide variety of domains. Deep Learning has allowed us to create a completely new, dynamic and adaptive method of securing communications. This is a break from the traditional static methods of cryptography, in which one must rely on a method or algorithm. Our approach instead leverages the adaptability of Deep Learning to learn, and get better, at protecting sensitive data. What is Deep Learning and how does Deep Learning make our cryptographical model superior? Deep Learning might seem to be very different from cryptography, but it is a very promising method to create a cryptographical model. It is almost perfect for our needs. Indeed, it shows significant advantages when used in the cryptographical model, such as the ability to self-learn and adapt to new threats. Moreover, unlike the traditional working method of cryptography, Deep Learning can learn and detect more complex patterns, allowing encryption and decryption at a higher level and thereby also maintaining efficiency. Finally, since Deep Learning lets us encrypt the message using a much more secure algorithm, we can ensure the confidentiality of the message. esteemed. We keep the message as secret as possible. We use very strong Deep Learning algorithms for encryption and decryption and these algorithms are the most secure. The algorithm adjusts itself dynamically as required according to the message and context, so it would be resistant enough to adversarial attacks.

Besides, our cryptographic system also strengthens the integrity of the transmitted message. Using Deep Learning, the system does not only encrypt the message securely but also embeds a self-validating system to verify its received version. This characteristic is intrinsic to our system and ensures that the received information is received in exactly the same way as encrypted by the sender, and hence is authentic, leaving no chance for unauthorized tempering during transit. To summarize, our proposed cryptographic system is an important step in the direction of secure communication. It is future ready, because it combines the best of both worlds - the stalwart principles of cryptography, and the adaptive and learning abilities of Deep Learning, promising to not only thwart the current alleged attack but also prepare way for a new breed of cryptographic protocols that are dynamic in nature and invincible. As we stand on the crossroads of cryptography and Deep Learning, this research gives us an opportunity to bring about a new wave of confidentiality and integrity in secure message transmission. It is a strong and immersive solution to the continuously evolving threat landscape, and certainly an area that can be improved further. On the research track, the next area is to explore the research in detail. The proposed encryption model should be subjected to a rigorous cryptanalysis, specifically targeting the attacks found in our research. The goal of this analysis was to discover the weaknesses of the refined system by examining traditional cryptographic threats. However, in the future, the system will be improved by the addition of new, sophisticated scrambling and confusion methods.

In particular, attempts will be made to investigate chaos-based algorithms as alternative methods to RSA, as well as more complex neural network architectures for the scrambling and confusion of inputs and outputs, which in turn will make the encryption process more difficult to observe and thus increase the overall strength of the system. But in order to reach this high level of security, the system has to be practically applicative. Consequently, the next step will be designing protocols for this technique to work in real-world implementation and assessing performance, scalability, and efficiency in various environments. In addition, working with specialists in mathematics, physics, and computer science will be the next phase in the development of this project. With the invaluable support of experts in these closely related areas, it will be possible to handle the challenges as early as possible and incorporate their feedback into future versions of the system. As this approach involves more than one field, effort has to be put into tuning the design and implementation of the cryptographic system. Standardization, user studies, and interoperability with existing protocols will also be further investigated, taking into account user needs. Clearly, experts in related fields will assist in designing these research components, as well as guide us on which standards to adopt. Consequently, the potential of implementing and evaluating this system in real-life, including the scalability and performance of this technique, will also be further discussed to achieve superior performance to current systems. Finally, it is important to note that these efforts are the first stage in standardizing the next generation of cryptographic systems designed to thwart next-generation attacks.

Acknowledgement

I would like to express my sincere gratitude to my co-author and Ph.D Guide, Dr. Narendra Shekokar, for his invaluable guidance, support, and scholarly contributions throughout the entire research process.

His expertise and mentorship have significantly enriched the quality of this work. I am deeply thankful to Dwarkadas J Sanghvi College of Engineering, my academic home, for providing the conducive research environment and resources necessary for the successful completion of this study. The support and encouragement from the faculty and staff have played a pivotal role in shaping my academic journey. I extend my heartfelt appreciation to my colleagues and peers at the institute for their collaborative spirit and insightful discussions, which have contributed to the refinement of the research.Lastly, I wish to acknowledge my family and friends for their unwavering support, understanding, and encouragement. Their love and encouragement have been a constant source of motivation throughout my academic pursuit.Thank you to everyone who has played a role, no matter how big or small, in making this research a reality."

References

[1] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS PUB 197, 2001. [Online]. Available: https://nvlpubs.nist. gov/nistpubs/FIPS/NIST. FIPS.197.pdf.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science, vol. 1666, pp. 388-397, 1999. [Online]. Available: https://link.springer.com/chapter/10.1007/ 3-540-48405-1 25.

[3] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1999. [Online]. Available: https://epubs.siam. org/doi/10.1137/ S0097539795293172.

[4] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, vol. 31, no. 2, pp. 26-34, 1998. [Online]. Available: https:// ieeexplore.ieee.org/document/4655281.

[5] F. Kandah and S. A. Al-Khateeb, "An Adaptive Neural Network-Based Cryptographic Technique Using Chaotic Systems," *Computers & Security*, vol. 99, p. 101955, 2020. [Online]. Available: https://www.sciencedirect.com/science/ article/pii/S0888327020307846.

[6] SuperData Science, "Convolutional Neural Networks (CNN) - Step3: Flattening," [Online]. Available: https://www.superdata science.com/blogs/ convolutional-neural-networks-cnn-step-3-flattening.

[7] J. Brown, "Rectified Linear Activation Function for Deep Learning Neural Networks," [Online]. Available: https://machinelearningmastery.com/ rectified-linear-activation-function-for-deep-learning-neural-networks/.

[8] Learn OpenCV, "Understanding Convolutional Neural Networks (CNN)," [Online]. Available: https://learnopencv.com/ understanding-convolutional-neural-networks-cnn/.

[9] MachineLearningMastery,"Adam Optimization Algorithm forDeep Learning,"[Online]. Available: https://machinelearningmastery.com/ adam-optimization-algorithm-for-deep-learning/.

[10] J. Li, "Stanford Dogs Dataset," Kaggle Dataset, [Online]. Available: https:// www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset.

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," [Online]. Available: https://arxiv. org/pdf/1703.08383.pdf.

[12] Towards AI, "An Introduction to Data Augmentation for Images Using TensorFlow's ImageDataGenerator," [Online]. Available: https://pub.towardsai.net/ an-introduction-to-data-augmentation-for-images-using-tensorflows-imagedatagenerator-45235dd0553b.

[13] S. Wang and Z. Hou, "Image encryption algorithm based on convolutional neural network," *Optics Communications*, vol. 482, 2021. [Online]. Available: https:// www.sciencedirect.com/ science/article/abs/pii/S0030401821002687.

[14] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251-1258, 2017.

[15] S. C. Iyer and N. Shekokar, "Identifying the Optimal Deep Learning Based Image Recognition Technique for Dog Breed Detection," in *14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*,2023.

[16] C. Szegedy et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[17] I. Goodfellow et al., *Deep Learning*, MIT Press, 2016.

[18] Y. Kim et al., "Remote Timing Attacks: Exploiting the Timing Side Channel on the Web," *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010. [Online]. Available: https://dl.acm.org/doi/10.1 145/ 1815961.1815966.

[19] D. J. Bernstein, "Cache-timing attacks on AES," [Online]. Available: http://cr. yp.to/antiforgery/ cachetiming-20050414.pdf.

[20] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-48658-5 1, 1999.

[21] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, 2007.

[22] F. Tram`er and D. Boneh, "Model Extraction and Zero-Knowledge Verification of Neural Networks," in *Proceedings of the 27th USENIX Security Symposium*, 2018. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/ presentation/tramer.

[23] F. Tram`er et al., "Stealing Machine Learning Models via Prediction APIs," in *Proceedings of the 25th USENIX Security Symposium*, 2018. [Online]. Available: https://dl.acm.org/doi/10.1145/3243734.3243792.

[24] T. Gu et al., "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [Online]. Available: https://openaccess.thecvf. com/content CVPR _ 2019/papers/Gu BadNets Identifying Vulnerabilities in _the Machine _Learning Model Supply Chain CVPR 2019 paper.pdf.

[25] T. Dong and T. Huang, "Neural Cryptography Based on Complex-Valued Neural Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4999-5004, Nov. 2020. DOI: 10.1109/TNNLS.2019.2955165.

[26] X. Li and P. Wang, "SEDL: A Symmetric Encryption Method Based on Deep Learning," in *Proceedings of the 12th Asia-Pacific Symposium on Internetware*, ACM, pp. 175-184, 2020. DOI: 10.1145/3457913.3457921.

[27] X. Wang and Y. Su, "An Audio Encryption Algorithm Based on DNA Coding and Chaotic System," *IEEE Access*, vol. 8, pp. 9260-9270, 2020. DOI: 10.1109/ACCESS.2019.2963329.

[28] M. Abadi and D. G. Andersen, "Learning to Protect Communications with Adversarial Neural Cryptography," arXiv preprint arXiv:1610.06918, 2016.

[29] R. Meng, Q. Cui, Z. Zhou, Z. Fu, and X. Sun, "A Steganography Algorithm Based on CycleGAN for Covert Communication in the Internet of Things," *IEEE Access*, vol. 7, pp. 90574-90584, 2019. DOI: 10.1109/ACCESS.2019.2920956.

[30] V. Sharma, M. Shukla, S. Srivastava, and R. Mandal, "Generative Network Based Image Encryption," in *Proceedings of the 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 1-5, 2020. DOI: 10.1109/ICICCS482 65.2020.9121060.

[31] Zhang, W., Wang, X., & Wang, Y. *Secure and efficient image encryption based on deep learning and chaos.* Multimedia Tools and Applications, 80, 16593–16617, 2021. https://doi.org/10.1007 /s11042-020-09885-8

[32] Boukela, L., & Akleylek, S. *Neural network-based cryptographic algorithms: A comprehensive survey.* Journal of Information Security and Applications, 55, 102649, 2020. https://doi.org/10.1016/ j.jisa.2020.102649

[33] Bhowmik, T., Hazra, R., & Roy, D. *Symmetric Key Cryptography Using Deep Learning Techniques.* In Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, 2020. DOI: 10.1109/ICCCNT49239.2020. 9225280

[34] Gupta, S., & Mehta, M. *A Review on Symmetric Key Cryptography Algorithms.* In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-5, 2020. DOI: 10.1109/ICCSEA49143.2020. 9132930

[35] He, K., Zhang, X., Ren, S., & Sun, J. *Deep Residual Learning for Image Recognition.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016. DOI: 10.1109/CVPR.2016.90

[36] Simonyan, K., & Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* In International Conference on

Learning Representations (ICLR), 2015. https://arxiv.org/abs/1409.1556

[37] El-Rabaie, S., Hadhoud, M. M., Abdel-Kader, R. F., & Zahran, O. *Secure Image Encryption Using Convolutional Neural Networks.* IEEE Access, 8, 108871108884, 2020. DOI: 10.1109/ACCESS. 2020.3001506

[38] Wei, L., Zhao, Y., & Liu, H. *Image Encryption Algorithm Based on MD5 and Neural Network.* Journal of Visual Communication and Image Representation, 65, 102693, 2020. https://doi.org/10.1016/j.jvcir.2019.102693

[39] Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks.* In 2016 IEEE Symposium on Security and Privacy (SP), pp. 582-597, 2016. DOI:10.1109/SP.2016.41

[40] Carlini, N., & Wagner, D. *Towards Evaluating the Robustness of Neural Networks.* In 2017 IEEE Symposium on Security and Privacy (SP), pp. 39-57, 2017. DOI:10.1109/SP.2017.49

[41] Kurakin, A., Goodfellow, I., & Bengio, S. *Adversarial Machine Learning at Scale.* In 5th International Conference on Learning Representations (ICLR), 2016. https://arxiv.org/abs/1611.01236

[42] Tram`er, F., Kurakin, A., Papernot, N., Boneh, D., & McDaniel, P. *Ensemble Adversarial Training: Attacks and Defenses.* In 6th International Conference on Learning Representations (ICLR), 2017. https://arxiv.org/abs/1705.07204