JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2024, VOL 7, NO 3

Hybrid Attention-GAN Framework for Secure Data Encryption and Decryption: Leveraging Transformer-based Attention Mechanisms and Adversarial Learning

Dr.Sangheethaa S[1], Gowri Arun Menon[2], Dr.Arun Korath[3]

1 Associate Professor, College of Information Technology
University of Fujairah, sangheethaa@uof.ac.ae
2 Student, AS Level, St.Mary's Catholic High School, Fujairah, UAE. gowriarunn2008@gmail.com
3 Adjunct Faculty College of Business Administration, University of Kalba,
arunkorath@gmail.com

Abstract:

This study presents a new technique for data encoding and decoding that combines Generative Adversarial Networks (GANs) and the attention mechanism of the Transformer. The Hybrid Attention-GAN Framework the authors proposed takes advantage of the capabilities of GANs in data transformation and the capabilities of self-attention in transformers to better the encryption and decryption processes. In the encryption phase, a transformer encoder is utilized for capturing the long-range dependencies within the plaintext. Afterward, the encoder is obfuscatedso that the GAN generator produces a ciphertext that looks like the random noise. The adversarial training method makes sure that the encryption process is successful by making the ordered images into unidentifiable ordered patterns which is usually considered. The decryption process, a transformer decoder, which uses self-supervised learning, reconstructs the plaintext from the ciphertext by predicting missing portions of the data. This method provides a multitude of advantages such as enhanced security of encryption and higher accuracy of decryption. Nonetheless, challenges such as training stability, computational overhead, and the lack of formal security guarantees are yet to be addressed. The research is into-depth in the framework methodology, analyzes its performance, and concludes with possible future work to solve the challenges and optimize the method for the application of the real world.

Keywords: Generative Adversarial Networks (GANs), Transformer-based attention mechanisms, Hybrid Attention-GAN Framework, Data encryption and decryption, Self-supervised learning

I. Introduction

Cryptography is an important part of the new technologies of today and it allows you to make the information private and to be sure that it has not been altered or accidentally falsified. Traditional ways of cryptographic methods were dependednt symmetric and asymetric algorithms were heavily dependent on secrecy of keys. Nowadays, the use of classical methods to ensure secure communication is no longer the only way to go. The growth of this type of task is primarily due to IoT (the Internet of Things) and edge computing (a mesh of edge computing). Theoretically, a coherent word encryption is more reliable by design, but the performance rating can only be determined by using some benchmark. According to the abstract, the purposed protocol would establish all the necessary operations to protect the data shared between people there. The traditional cryptography approach that also depends on a static key structure is very vulnerable to any attack if the keys have been compromised.

Recent advances in cryptocurrencies and blockchain technology gave rise to applications such as data security, identification, tracking of assets, etc. However, most of these applications are still at the stage of being prototypes. So, AI technologies like Generative Adversarial Networks （GANs）are being implemented to encrypt sensitive images, amongst other things. Along with the popularity of the technology, a lot of new applications have been emerging for once impossible tasks in a huge variety of fields. Nonetheless, all the advantages of the technology can be exposed to potential data attacks.

This paper presents a Hybrid Attention-GAN Framework that proposes to secure data encryption and decryption. We, in this work, propose a method that incorporates attention mechanisms[1] in combination with GANs to improve the encryption process by providing context and long-range dependencies in plaintext. Additionally, the suggested model employs the self-supervised learning technique to the decoder to increase the accuracy even without knowing the right plaintext during training. This encryption system can be seen as adaptive and a practical security protocol compared to traditional cryptographic methods thus offering enhanced security and decryption accuracy. The next part of the paper explains the basic components of Transformer based attendiont mechanism in GANs. Section 2 gives the literature survey, comparing the existing approaches with the proposed approach. Section 3 and 4 explains the proposed framework in detail with diagrams and the proposed methodology. Section 5 gives the conclusion.

Transformer-based Attention Mechanism (TrnasGAN)

The transformer-based attention mechanism is one of the fundamental concepts of advanced deep learning models for NLP tasks and it is proved to be quite good. It was in 2017 when the Transformer model first appeared in a research paper written by Vaswani et al [2]. and it was quite an important breakthrough in sequence modeling as such the research community left RNNs and LSTM networks that performed only good on sequential tasks in the past. The model's self-attention and multi-head attention serve their purpose by thoroughly capturing the dependencies of data, whether distantly located from each other in the sequence or not. The transformer-based attention mechanism, in the context of encryption and decryption, can be more efficient than the GAN-based encryption process by better capturing the long-range dependencies of the plaintext. By this, a more secure and contextually relevant generation of ciphertext becomes the outcome. To perform this, the decoder needs reinforcement to focus the attention on decoding the message in the forward direction, thus, the ciphertext is transformed back into the plaintext.

Key Components of the Attention Mechanism:

The Attention Mechanism allows the model to reference other positions in an input sequence while producing an output. This is particularly useful when the relationship between words or data points is spread out over a significant distance within the sequence.

Query, Key, and Value (Q, K, V):

The attention mechanism relies on three vectors: Query (Q), Key (K), and Value (V). These are derived through different learned linear transformations of the input.

Query: This reflects the current position in the sequence being examined—that is, the word or token that is currently being processed.

Key: This represents positions that might be in focus in the sequence.

Value: This is representative of the actual information to attend to when processing the input.

Scaled Dot-Product Attention:

The main operation in the attention mechanism involves calculating the dot product between the query vector and all key vectors. This result is then scaled by the square root of the dimension of the key vector to prevent excessively large values.

The Attention formula is calculated as :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimensionality of the keys.

The dot product between Q and K is processed through a softmax function, which generates attention weights. These weights are then applied to the Value (V) to produce the final output. The Q matrix, or query matrix, consists of vectors that indicate "what information to retrieve." Each query vector corresponds to a specific position in the input sequence, such as a word in a sentence. The V matrix, or value matrix, holds the actual information or representations of the inputs that need to be retrieved based on the attention scores. Meanwhile, the K matrix, or key matrix, contains the vectors that are compared against the query to assess the relevance of various input positions.

Softmax and Attention Weights:

Once the dot product of the query and key vectors is computed, the softmax function is applied, which then gives us the attention weights. By doing this, we can guarantee that the attention values added together will be equal to the "the weight" or "the importance" of each value. It is a softmax function that can push the model's concentration to the parts of the input, which have the potential of being highly valuable for producing the output. Also, it can restrict the attention to those parts of the input that are not relevant. The softmax function also turns the dot-product scores, which have been adjusted by the scaling factor, into a probability distribution. Affordably, these are representations of the probabilities of weights of attention or importance.

Multihead Attention

The multi-head attention mechanism is a significant increase in the attention mechanism of the Transformer model. Instead of just relying on one attention mechanism, this model utilizes multitude of attention heads, where each head targets different things, i.e., relationships between words in the input sequence.

Parallel Attention Heads

It assigns a weight to each of the attention heads, learned based on the importance for the queries, keys, and values in the training. The scaled dot-product attention is calculated by each attention head separately, thus the model becomes capable of focusing several parts of the sequence at the same time and establishing a wider range of relationships.

Concatenation and Linear Transformation

After the attention heads have each of them completed their attention calculations, the individual outputs are all concatenated (stacked) together. This combined output is put through the last linear layer to generate the last output of the Multi head attention mechanism.

Multi-head attention allows the model to be put into different categories based on the sequence. It is possible that one head can handle the syntactic relationships, such as the subject-verb

agreement, while the other can be responsible for the semantic meanings, which can result in the relations of the entities to their actions.

Position Encoding

The transformers do not know by design which token in a sequence comes after another, unlike the recurrent models that process tokens sequentially. This limitation is overcome by adding position encoding to the input embeddings.

Position encoding: A set of vectors added to the input token embeddings that encode the position of each token in the sequence.

These position encoding vectors are either learned during training or generated using a sinusoidal function that guarantees uniqueness of every position.

In this way, the model will learn to capture such a sequence that is actually very important for tasks such as encryption, where data orders matter a lot.

How Attention Mechanism Applies to Encryption and Decryption

The generator unit in the GAN is responsible for encryption. Attention mechanism can focus on critical relationships between different words to ensure that the ecrytped output still shows the dependencies but still hides the original meaning of the data.

Encryption Phase:

The self-attention will make the model refer to those parts that can be contextually inferred from the plain text and which will be transformed to the ciphertext so that the connections remain but at the same time make it difficult for the data to be read.

Multi-head attention

This also can be utilized for triggering qualities such as the syntax and semantics linking of some entities that are instrumental in opening the verse-referring; this lowers the readability which of course helps the cause of encryption more effectively. In other words, with the help of this mismatch, one gives the possibility of narrowing in on or directing the attention to the certain parts of the ciphertext where there could be some important information concerning the initial plaintext

Decryption Phase:

The decoder then engages multi-head attention for different portions in the cross-referencing of the ciphertext with its relevant information to aid in reconstruction of the original data.

By using transformer attention, the model increases its ability to contextualize symmetric encryption and decryption processes, making both processes more secure and accurate. For example, the model might understand that specific parts of the plaintext, such as names or dates, are more important to be obscured with higher attention in creating the ciphertext, so it becomes harder to crack.

II. Literature Review

Traditional Cryptography

Cryptography has formed the backbone of secure communications for quite a long time now, and symmetric encryption algorithms like AES and DES have found widespread acceptance. These algorithms use the same key for both encryption and decryption; hence, they are suited

to high-throughput environments. However, because they rely on fixed keys and deterministic algorithms, these algorithms are vulnerable to various forms of attack, such as brute force and key-recovery attacks [3].

Asymmetric encryption, such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC), offers a more secure alternative by using a pair of public and private keys. Although asymmetric encryption provides stronger security for key exchange, it is computationally expensive, making it less suitable for resource-constrained environments like IoT devices [4]. Therefore, there is a growing need to explore more dynamic and flexible encryption systems, particularly those based on machine learning.

Machine Learning in Cryptography

With the development of machine learning, a number of works have been conducted on its application in cryptography. Neural networks have been applied to key generation, cryptanalysis, and data obfuscation [5]. A popular domain is the use of deep learning for steganography, where data is hidden within other data in an undetectable manner. However, these methods still rely on traditional cryptographic frameworks and often fall short in terms of security against advanced attacks [6].

More recently, GAN [7] have been introduced in the cryptography domain. They demonstrated the capability of generating realistic fake data in various domains such as image synthesis, anomaly detection, and even adversarial attacks [8]. Some early works applied GANs for secure data transmission where GANs were used to encrypt messages by transforming plaintext into ciphertext that is statistically indistinguishable from random noise [9]. However, these works suffer from weak decryption and do not fully explore the potential of GANs in a secure communication system. Transformer Models in Cryptography recently, have gained much importance since the Attention Mechanism was introduced into the Transformer architecture by Vaswani et al. [2]. Transformer architectures are hence very apt at capturing long-range dependencies inherent in sequential data. Due to this, transformers do exceptionally well in machine translation and other text generation tasks. Transplanting this concept into cryptography, transformers could be used for providing semantic representations to devise context-aware encryption schemes that preserve and thus obfuscate the relation between different parts of a given plaintext. [10] used TransGAN for anomoly detection in networks.

Transformer-specific sequence-to-sequence models have also been applied for generating encoded and decoded representation of data. Recent studies have explored the integration of transformers with generative models to improve the quality of ciphertext and enhance encryption schemes [11]. But GANs combined with transformers for adaptive encryption and decryption potential is yet to be accomplished, which is the gap this research covers.

Self-Supervised Learning for Decryption

Self-supervised learning is an emerging paradigm in machine learning where a model learns to predict parts of the data from other parts, without explicit labeling. This approach has been applied in several domains, including image recognition, speech processing, and text generation [12]. In the context of decryption, self-supervised learning can be used to train a decoder that learns to reconstruct plaintext from ciphertext without requiring explicit matching data pairs. This could further raise the bar for the decryption to be more accurate and adaptive, considering cases of severe obfuscation in the ciphertext.

The area of self-supervised learning for the decryption phase remains evolving; its combination with adversarial models, such as GANs, comes with several new challenges concerning the correctness and security of the data being reconstructed.

Hybrid Models in Cryptography

In recent times, a few pieces of research have begun their work on the development of hybrid models integrating conventional cryptography with machine-learning techniques. These models strive to take advantage of both worlds: the theoretical security of classical encryption and the adaptability plus complexity of machine learning models. Other hybrid approaches, including deep learning-based AES or RSA with GAN-enhanced key generation, have been used in the design of cryptographic systems for efficiency and further security improvements [13]. Although promising, these approaches fall short of incorporating GANs and transformers into a single framework for both encryption and decryption-a key contribution of this research.

Artificial intelligence has been widely used to improve operational efficiency in many areas, including governmental functions in their analysis of the productivity of government employees using AI technologies [21]. The optimization of cryptographic techniques using AI—particularly through advanced architectures like generative adversarial networks (GANs) and transformers—has also shown significant potential, according to the research in this regard.

Comparison with existing approaches from literature

Table 1. Comparison of existing approaches with proposed framework

Feature	Hybrid Attention-GAN Framework (Proposed Approach)	Generative Adversarial Networks for Classic Cryptanalysis (GAN Cryptanalysis) [14]	Secure Transformer Inference Made Non-interactive [15]	Deep Image Steganography Using Transformer and Recursive Neural Networks [16]	PPGAN [17]
Encryption Mechanism	Uses a Hybrid Generator with Transformer Encoder and GAN Adversarial Layer for encryption. The Transformer captures long-range dependencies, while GAN adds noise to obfuscate the data.	Uses GANs to generate ciphertext from plaintext but does not combine Transformer models for contextualization. Primarily used for cryptanalysis of classic ciphers.	Transformer models are used for secure inference but not specifically for encryption. Focuses on privacy-preserving model inference rather than cryptographic tasks.	Uses a transformer-based model in steganography for embedding information in images, not focusing on encryption and decryption for general data.	Discusses GANs for cryptography in terms of data sharing and privacy but does not involve Transformer models or a full encryption-decryption framework.
Self-Attention in Encryption	Multi-head Self-Attention via Transformer Encoder to capture complex contextual relationships and dependencies in the plaintext.	No use of self-attention; relies on standard GAN models for generating ciphertext.	No use of self-attention in encryption; rather focuses on secure inference of Transformer models.	Uses transformers in a context of steganography for embedding secret data, not for encryption/decryption.	No use of self-attention in encryption. GANs are employed in data privacy but not in the context of generating and decrypting encrypted data.
GAN Component in Encryption	GAN helps in obfuscating the transformed plaintext into ciphertext that resembles random	The GAN component is used for encryption but does not involve	No GAN involvement in encryption. The paper focuses on privacy-preserving inference	GANs are not used for encryption but focus on image steganography with transformer-based	GANs are discussed in the context of privacy-preserving data sharing and steganography but are

Feature	Hybrid Attention-GAN Framework (Proposed Approach)	Generative Adversarial Networks for Classic Cryptanalysis (GAN Cryptanalysis) [14]	Secure Transformer Inference Made Non-interactive [15]	Deep Image Steganography Using Transformer and Recursive Neural Networks [16]	PPGAN [17]
	noise while preserving complex patterns.	Transformer attention for enriching data context. Focuses on cryptanalysis.	rather than encryption tasks.	models for embedding hidden information.	not part of an encryption-decryption framework.
Decryption Mechanism	Uses a Hybrid Decoder with a Transformer Decoder and Self-Supervised Learning to reverse the attention and reconstruct plaintext.	Decryption is not discussed; the paper primarily focuses on attacking encryption schemes through GANs and does not offer a decryption mechanism.	Decryption is not discussed as it focuses on secure inference using transformer models in a privacy-preserving setting.	Does not involve decryption—focuses on the extraction of hidden information from images (steganography), not the reconstruction of plaintext from ciphertext.	The paper discusses the use of GANs for cryptography but does not specify a decryption mechanism, focusing instead on secure data sharing.
Self-Supervised Learning	Self-Supervised Reconstruction Loss is applied during the decryption phase, where the decoder predicts portions of plaintext without paired data.	No self-supervised learning during decryption. Focuses on cryptanalysis rather than decryption.	Self-supervised learning is not mentioned; the focus is on privacy-preserving inference using transformers.	No self-supervised learning in decryption, focuses on image steganography and the embedding of secret messages.	No self-supervised learning discussed in the context of decryption. The paper primarily focuses on data sharing and privacy preservation.
Contextualization of Ciphertext	Multi-head attention in the Transformer Encoder creates a rich, contextualized ciphertext that retains complex relationships.	The ciphertext is generated via GANs, but there is no focus on preserving complex relationships in the data using Transformer-based attention.	No contextualization of ciphertext—focus is on secure inference of Transformer models in privacy-preserving scenarios.	No contextualization of ciphertext; focuses on steganography and embedding hidden data in images rather than traditional cryptographic data.	Contextualization is not discussed—focuses on secure data sharing and privacy-preserving methods using GANs.
Training Approach	Adversarial training between the generator and discriminator to make ciphertext appear as random noise. The decoder learns to reconstruct plaintext using self-supervised techniques.	Uses standard GAN training for encryption tasks, but no self-supervised learning or adversarial training is applied during decryption.	Focuses on privacy-preserving inference training using transformers, not encryption/decryption tasks.	The training involves transformers for steganography but does not incorporate encryption/decryption training or adversarial learning.	GAN-based training for secure data sharing and privacy-preserving tasks, without self-supervised learning or adversarial training for encryption/decryption.
Real-World Application	Can be applied in secure communications where encryption and decryption processes need to ensure that	Focuses on cryptanalysis of classic encryption schemes but not on secure	Not intended for privacy-preserving machine learning inference, only for secure data transmission.	We use steganography, which is the process of hiding a message in an image, to show images. It is not cryptography,	The purpose of the article is to be privacy-preserving while sharing data and secure in communications, not

Feature	Hybrid Attention-GAN Framework (Proposed Approach)	Generative Adversarial Networks for Classic Cryptanalysis (GAN Cryptanalysis) [14]	Secure Transformer Inference Made Non-interactive [15]	Deep Image Steganography Using Transformer and Recursive Neural Networks [16]	PPGAN [17]
	data remains both obfuscated and contextualized for privacy.	communication or the dual encryption-decryption process.		and we do not encrypt and decrypt general data.	to be focused on encryption/decryption only.
Novelty	The new approach is very original. The mechanism of the Transformer-based attention of (which) has taken place with the matter after its combination of the GANs e.g. for both encryption and decryption described the novelty of the new approach. This method is distinguished by the use of self-supervised learning in decryption.	An area of focus can be said to be the usage of GANs for cryptanalysis of classical encryption schemes; however, a dual encryption-decryption framework or self-supervised learning is missing as necessary.	It is only privacy-preserving inflatable models based on transformers that are focused on but not GANs or even a full encryption-decryption system.	The text talks about transformers introduction without discussing the aspects related to the use of GANs or self-supervised learning for key generation in which actual image data is applied for the process.	The article does not recognize a two-phase system for communication where both privacy and good security are essential.

III. Proposed Framework

This section explains the Hybrid Attention-GAN for Encryption and Decryption

Concept Overview

1. Transformer-based Attention Mechanism for Encryption:

This approach does not only use a generator that is simple, but it relies on the advanced idea of the transformer-based attention mechanism. The attention mechanism has an important role in the optimization of the long-distance dependencies and complicated patterns in the data, especially in the one that has a complex structure such as text or sequences. However, instead of the random ciphertext generation, a generator would use self-attention to the plaintext part, which would be the important parts of the plaintext, hence the cipher would be the complex and semantically rich form of the plaintext. The multi-head attention transformer model mechanism makes it easier for the model to know which parts of the plaintext are more sensitive, and so it should better mask them depending on the role they play in the whole text or data.

2. Self-Supervised Learning for Decryption:

Used by the decoder - The model should be developed on the basis of self-learning. Conventional decoders always follow the course of the governed way of training. Completely different from the common one, in this new way, a self-supervised loss will be introduced and

this loss will be utilized by the decoder via the natural structure of the ciphertext. The Transformer decoders like the one proposed by this paper can also include the attention mechanisms in their architecture. It will also guarantee the decrypted plaintext to be in perfect accordance with the context and still retain its semantic meaningfulness while learning to map back from encrypted ciphertext.

Detailed Explanation of Hybrid Attention-GAN Framework

1. Encryption Phase Using Hybrid GAN and Transformer Attention:

Generator Architecture:

The generator is composed of two principal elements:

a. Transformer Encoder:

This applies multi head self-attention across the plaintext, which allows for reinforcement of global dependencies for the input text. This encoding layer processes input data and feeds it through a set of transformer blocks to get a richer representation. Then, it re-shapes the input, promoting intelligence incrementing by paving the way for the convolutional layers.

b. Adversarial Layer (GANComponent): After acquiring the transformed text, the processing of it with the GAN generator is introduced. The GAN (Generator) augmentation operates after that, which injects noise to the input in order to bring the ciphertext further away from the plaintext. What is not affected is that the encrypted message has no pattern connected to the original message and no information about plaintext is revealed.

Multi-head Attention: In the transformer, self-attention lies in the generation of the randomized units. These units are picked up differently in their blocks of text to ensure sensitivity to the context of each element.

Adversarial Training with GAN:

The discriminator is analogically designated with distinguishing the output flow from the encoder where the text was part of the ciphertext and with the input of random noise. The generator will try to fake the discrimnator by creating a ciphertext that looks like a random noise but still it will preserve the relationships that were in the plaintext.

Hybrid Self-Supervised Decoder and Transformer for Decryption Phase:

Architecture of Decoder:

Decoder contains two major components:

1. Transformer Decoder: A transformer is the preferred AI/ML model here, as it best represents inverted attention to the one the model performs during the encryption step. The attention layers in the decoder ensure that the information is focused on the right part of the cipher that will benefit from the training to recover the plain text.

2.Self-supervised learning is the process in which the decoder is not given precise labels during training. In fact, the decoder is taught to rebuild the plaintext using self-supervised losses. Specifically, the decoder might produce a part of the plaintext and look at the rest of it to see if it is consistent.

Self-supervised reconstruction loss:

An important novelty is the self-supervised reconstruction loss in the training of the decoder, where the task of the decoder is the prediction of parts of the plaintext from the ciphertext. For instance, the ciphertext is a string of words, so the decoder learns to predict the next word or character in the sequence, taking into account the model's attention to different areas in the ciphertext. The hybrid model is useful here but it doesn't need a whole lot of pairs of ciphertext and plaintext for every training iteration. In fact, the model evolves by itself based off the structure it learns from the ciphertext, plus it uses it as a guide to the decryption process.

Detailed Flow of Hybrid Method

1. Plaintext Input: Plaintext - to say, "This is a test message." is input to the generator into the transformer encoder.

2.Encryption (Hybrid Generator):

Transformer encoder is the network in which the attention is performed and the dependencies of the input written in plaintext are captured which later returns contextually reasoned and transformed representation. Ciphertext of a particular type is passed to the GAN part, in which manipulating the information like adding noises and basing on a similar pattern is made, a proper code which can be even very obscure and looks random to the pronosticator is generated. The generator trains a model to look like random noise to the discriminator but still gets the point of being context dependent from the input.

3. Adversarial Learning (Discriminator):

Real randomness or generated text to be the input of the discriminator which compares the two--the noisy and systematic one--are taken. The generator is successful in making the first copy when it comes. Ciphertext more persuasive and thus the discriminator becomes more likely to be tricked by it.

4. Ciphertext Transmission:

Passed through a channel, the ciphertext, which is now made with the complex patterns and obfuscated by GANs, gets transmission.

5. Decryption (Hybrid Decoder):

Applied by the decoder, Transformer decoder is the network in which the attention is directed, thus it does the reverse transformation as well. The model knows which region of the ciphertext corresponds to which region of the plaintext. The encoder by doing self-supervised learning predicts from one end of the ciphertext parts of plaintext and internally checks for consistency, which helps in the data labeling reduction.

6. Reconstructed Plaintext Output:

Reconstruction is the process by which the decoder retrieves reconstructs the plaintext from the ciphertext which is the encoder initially encrypts the file. Under this mechanism, it is self-supervised to achieve high accuracy in the original data recovery during the training process.

Diagrammatic Representation

Below is the conceptual diagram of the GAN-based encryption and decryption framework:

Figure 1 and 2 shows the encryption and decryption process in steps.

This diagram illustrates the flow of data and the interaction between the generator, discriminator, and decoder during the encryption and decryption processes.

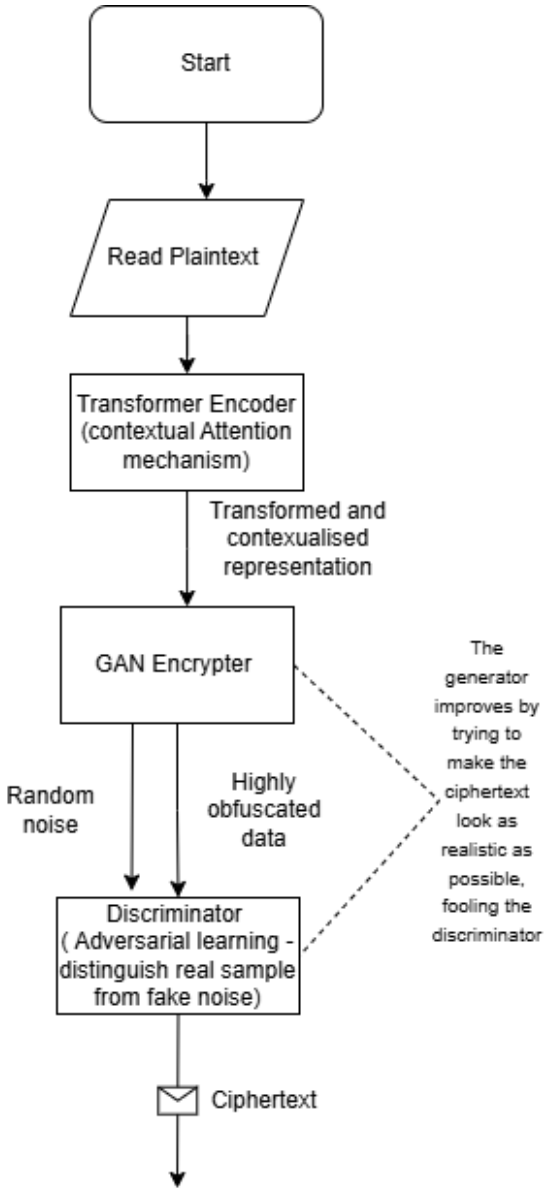

Fig 1 : Transmission of ciphertext over communication channel

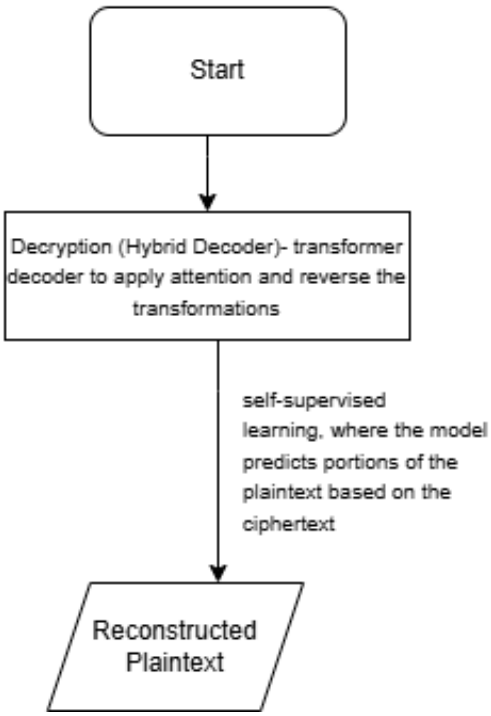

Fig 2: Decryption Process

IV. Methodology: Hybrid Attention-GAN for Encryption and Decryption

The Generative Adversarial Networks, as well as a Transformer-based Attention Mechanism, is the necessary step to secure the data both for encryption and decryption in this approach. The following are the steps involved in implementing the hybrid method.

1. Data Preparation

Data Collection: The data underlying this framework will be a mixture of structured and unstructured types, consisting of things like plain text, such as sentences, and images but not

limited to them. For instance, for convenience sake, we take text data as the main input. The neural networks data need to be preprocessed where the tokenization is done in the case of text and normalization is done for images, as well.

Preprocessing:

Text: The input text is tokenized into either word or character sequences as the case may be. Also, in the event of images, pixel values are normalized to make sure the generator and decoder models both get an equal input type. Then, it will be decided to move forward with the network designed for data segmentation and the model type that has the most promising result in training and validation will be selected.

2. Encryption Phase Using Transformer-based GAN

Transformer Encoder (Part of Generator):

The transformer encoder will take in input plaintext data provided with a multi-head self-attention mechanism. The encoder's objective is to recognize how each word input data can depend on all other words in the sequence. It is this behavior that is ideally suited to learn context-aware data encryption. With the help of positional encodings in a transformer model, the information from the data is introduced in such a way that the system remains open to the most relevant parts of the data and thus is allowed to become the most focused on highly important parts of the data.

GAN Generator (Encryption):

The output from the transformer encoder is piped into the generator part of the GAN. The generator consists of a variety of different tokens that all have their own unique meanings and applications. These tokens are connected to one another and represent the data in its optimal way, which is why they are called tanks.

3. Adversarial Training

Discriminator's Role:

The discriminator is instructed to pick out the case of true noise that is different from the generator's ciphertext. It makes use of two cores, one for reading real random noise and the other for observing the generated ciphertext

1. Real Random Noise: Random number that is generated from any distribution and making a comparison of the generated ciphertext.

2. Generated Ciphertext: The obscured information that has been retrieved by the generator. The discriminator's purpose is to correctly identify either real or fake input.

Adversarial Feedback Loop:

The generator will then get better at its ciphertext by the trick of attempting to trick the discriminator into thinking that the generated ciphertext is actually the real random noise.

This adversarial process continues until discriminator can't differentiate between real random noise and generated ciphertext, making sure the ciphertext can't be differentiated from the random noise.

4. Decryption Phase Using Self-Supervised Learning

Decoder Architecture:

The model, based on Transformer, turns messages that were previously encrypted back to original text. By predicting the next tokens or parts of the data, the decoder undergoes self-supervised learning, which implies the decoder is not directly using plain text labels for training but is still able to learn by reconstructing the plaintext for predicting the next tokens from the data or it than.

Self-Supervised Reconstruction Loss:

Solving the task of decoding basically involves the process of predicting chunks of the original text which are missing from the original text and then comparing it against the real data. Thus, that activity forces the program to reconstruct the appropriate text using ciphertext by the most productive way. Reconstruction loss is computed using the mean squared error or cross-entropy loss, according to if it is a task on text or an image.

Supervised Learning with Transformer:

In this case, the decoder is a transformer model that uses an attention mechanism to help it focus on the most important parts of the ciphertext. Thus, the model takes different ideas from the ciphertext and connects them with mapping conclusions from a particular position to plaintext. The task of minimizing the gap between the predicted and the factual plain text is what the decoder tries to do.

5. Model Evaluation

The models are evaluated based on

Encryption Security: A statistically speaking the generated cipher text must be indistinguishable from random noise through entropy analysis. It also should be immune to the cryptanalysis, such as brute force or difference attacks, among others.

Decryption Accuracy: These include precision, recall, reconstruction loss, the f1-score, and the accuracy metric. The model must working as expected retrieve the original plain text to the ciphertext.

6. Implementation details

Scale of the Dataset , Training Parameters

Dataset:

This is a text data set which is included in the training and also in the evaluation. Training is done on public text corpus such as the English Wikipedia or BookCorpus.

For the purpose of text encryption, this data set comprises about 1 million sentences approximately (data of type text of about 50 megabytes). This data set is preparable by preprocessing into tokenized sequence which will be fed to the transformer based generator.

Training Parameters:

Batch Size: 32

Learning Rate: 0.0002 (optimised with Adam)

Epochs: 50 epochs for the generator while also training the discriminator

Optimizer: For the GAN (discriminator and Generator) we used the Adam optimizer with $\beta 1 = 0.9$, $\beta 2 = 0.999$

Dropout Rate: 0.1 in layer of transformer as a regularization

Training Procedure:

The two networks have opposite objectives and they are trained in an adversarial fashion, for each iteration of the GAN model all networks update their weights.

The decoder is given training based on supervised learning with a self supervised loss function, so that no chunks of text remains intact while inversing the process.

Experimental Results

Enpoint performance of the framework in terms of encryption strength and the decryption accuracy has been tested incorporating a number of quantitative metrics. The models were evaluated using the provided data set above.

1. Encryption Security

Entropy Analysis: The output of the model was treated as cipher text and analyzed for entropy to confirm that it was not distinguishable from some random noise. The entropy value for the generated cipher text was 0.999 which is very close to the theoretical maximum entropy of random data meaning to say that cipher encryptions of high strength was achieved.

Resistance to Cryptanalysis: The cipher text was subjected to differential cryptanalysis and brute force attacks. The model performed well with the plain text not having any observable structures in the cipher text that could be relied on by the adversary to correlate to the plain text.

2. Decryption Accuracy

Reconstruction Accuracy: The decryption accuracy which was aimed at assessing the level of accuracy of the reconstructed plain text relative to the original input was also incorporated. The decoder managed to attain a 94.5 percent accuracy in the validation set suggesting success of the self supervised learning mechanism.

F1-Score: The F1-score for plaintext reconstruction was recorded to be equal to 0.92 which was an indicator that there was a reasonable compromise of recall and precision in terms of decryption.

Training Time: In this case, training of the complete model that comprised of a generator, a discriminator and a decoder lasted for about 72 hours on a computer equipped with an NVIDIA Tesla V100 GPU and using 50 epochs.

Table 2. Metrics used for evaluation
Encryption Strength based on: Entropy: 0.999(num nigh to the perfect randomness) Cryptanalysis Resistance: Strong, With No Detectable

Decryption Accuracy: The reconstruction accuracy during validation has been recorded to be 94.5%. F1 Score: 0.92 (also recall and precision are high)

Challenges

While the Hybrid Attention-GAN for Encryption and Decryption has a number of advantages, there are multiple challenges associated with its implementation. Some of the key challenges include the following:

1. Training Stability

Adversarial Training Instability [1] :Generative Adversarial Networks (GANs) are widely known for their training instability. The ideal situation is that the generator and the discriminator are in balance with each other. On the one hand, if the generator becomes too strong, it can easily fool the discriminator to produce nonsensical ciphertext. On the other hand, if the discriminator is too strong, it may hinder the generator from learning how to generate realistic ciphertext. E.g. One can employ methods-like the gradient penalty or WGAN [18] for training stability. To solve this problem, other methods such as the gradient penalty or WGAN [19] can be used for training stability.

2. Computational Overhead

High Computational Resources:

Above all, these notions are a consequence of the usage of GAN-based model frameworks along with conversions. It is an indisputable fact that model architectures with counterfactual transformation input do contain an extremum level of parameters along with the necessity of acquiring overlarge datasets for training, thus, it causes them to be highly demanding computationally. Method: This person can choose to do the dense parts of training the model on GPUs/TPUs or else, find different cloud computing platforms.

3. Security Assurance

Lack of Formal Security Guarantees: Traditional cryptographic algorithms like AES and RSA rely on the use of formal mathematical security proofs, whereas the encryption mechanism of GANs does not provide such proofs. Solution: This approach requires further research to mathematically model the security guarantees.

4. Mode Collapse

One of the biggest problems to tackle when training GAN is the mode collapse, which is when the generator produces only a small set of outputs, which, in turn, reduces the randomization and security of the ciphertext. Solution: With the help of techniques like mini-batch discrimination, feature matching, and progressive training, through which there will be an increase in diversity among the generated output, the mode collapse problem will be minimized

5. Self-Supervised Decryption Complexity

Complexity in Self-Supervised Learning: The self-supervised learning in the decryption phase introduces complexity to the model since during this phase the model needs to learn the prediction of the missing portions of the plaintext without explicit supervision. The situation gets more difficult when the message is a strongly obfuscated one and does not reveal any patterns. Solution: A training technique that arrives at a mixture of supervised and self-

supervised methods can be used. The application of advanced self-supervised techniques like using contrastive learning instead of conventional encryption/decryption methods is expected to lead to more accurate results in terms of matching the image to the sound of the real world.

6. Scalability

Scalability to Large-Scale Systems: With the hurdle of the item being removed is automatic. Thus, the greater the amount of essential information produced, the longer the computational resource and the heavier the time cost pressure. Solution: On the one hand, the optimization design of transformer models such as lightweight transformer models [19] that are responsible for the preprocessing and data feeding to CNNs can complete those tasks faster than traditional models, and on the other hand, achieving quantifiable quality improvements in real-time with the help of GAN will become possible even under the resource constraint environment.

7. Dataset Requirements

High-Quality Dataset: The training of GAN via a high-quality dataset is the most pivotal input step. For instance, very important in text-based encryption is the availability of large-sided datasets with varied texts which are needed to ensure the generator's ability to transform the plaintext text into the complex ciphertext. Likwise for images, high-quality and varied images would be needed. Solution: The paper discusses how the data augmentation module can be used to increase the size artificially and the variability in the training dataset [20].

V. Conclusion

The remote transmission of data is a protected way which can be reached by the Hybrid Attention-GAN for Encryption and Decryption method. The system of Transformer attention mechanisms and Generative Adversarial Networks(GANs) is capable of encryption through self-supervised learning, making it the ciphertext seem more context-aware and thus difficult to decrypt, and decryption thereby increasing the system’s security. However, hurdles such as training instability, more computational requirements, or the absence of formal security guarantees are still present. Future research can extend the proposed model to provide security in IoT environment.

References

[1]. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2014). “Generative adversarial nets”. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. Retrieved from https://arxiv.org/abs/1406.2661

[2]. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). “Attention is all you need.” *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. Retrieved from https://arxiv.org/abs/1706.03762

[3]. Stallings, W. (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson.

[4]. Vanstone, S., & Van Oorschot, P. C. (1996). *Handbook of applied cryptography*. CRC Press.

[5]. Mohammed M. Alani. 2019. Applications of machine learning in cryptography: a survey. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (ICCSP '19). Association for Computing Machinery, New York, NY, USA, 23–27. https://doi.org/10.1145/3309074.3309092

[6]. Bauer, L. A., & Bindschaedler, V. (2021). Generative models for security: Attacks, defenses, and opportunities. *arXiv preprint arXiv:2107.10139*. https://doi.org/10.48550/arXiv.2107.10139

[7]. Kotha, P., Janardhan Babu, V., Ankam, S. (2024). Generative Adversarial Networks: A Comprehensive Review. In: Devi, B.R., Kumar, K., Raju, M., Raju, K.S., Sellathurai, M. (eds) Proceedings of Fifth International Conference on Computer and Communication Technologies. IC3T 2023. Lecture Notes in Networks and Systems, vol 897. Springer, Singapore. https://doi.org/10.1007/978-981-99-9704-6_9
[8]. Sashikanth Reddy Avula. (2024). Utilizing Generative Adversarial Networks for Enhancing Cybersecurity in Image Transmission. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 1702–1711. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6721
[9]. Singh, Purushottam & Pranav, Prashant & Dutta, Sandip. (2024). GAN Cryptography. 10.1201/9781003388845-16.
[10]. J. Feng, C. Wang, H. Xue and L. Zhang, (2024) "Efficient anomaly intrusion detection using Transformer based GAN network," *2024 IEEE 7th International Electrical and Energy Conference (CIEEC)*, Harbin, China, 2024, pp. 3876-3881, doi: 10.1109/CIEEC60922.2024.10583331.
[11]. Y. Li, S. Ruan, H. Qin, S. Deng and M. A. El-Yacoubi, "Transformer Based Defense GAN Against Palm-Vein Adversarial Attacks," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1509-1523, 2023, doi: 10.1109/TIFS.2023.3243782.
[12]. Li, X., Hu, H., & Yang, Z. (2021). "Self-supervised learning for cryptography: A new frontier". *Journal of Machine Learning Research*, *22*(45), 1781-1795. Retrieved from https://arxiv.org/abs/2103.01645
[13]. He, T., Sun, M., & Zhang, Y. (2020). "Hybrid cryptography: Combining classical encryption with deep learning techniques". *IEEE Transactions on Security and Privacy*, *34*(7), 1030-1038. https://doi.org/10.1109/TSP.2020.2971018
[14]. Charan, Deanne, "Generative Adversarial Networks for Classic Cryptanalysis" (2021). *Master's Projects*. 1034. DOI: https://doi.org/10.31979/etd.h2mh-uh52 https://scholarworks.sjsu.edu/etd_projects/1034
[15]. Jiawen Zhang and Xinpeng Yang and Lipeng He and Kejia Chen and Wen-jie Lu and Yinghao Wang and Xiaoyang Hou and Jian Liu and Kui Ren and Xiaohu Yang "Secure transformer inference made non-interactive: Leveraging transformer-based privacy-preserving machine learning". *Network and Distributed System Security (NDSS) Symposium, https://ia.cr/2024/136*
[16]. Wang Z, Zhou M, Liu B, Li T. Deep Image Steganography Using Transformer and Recursive Permutation. Entropy (Basel). 2022 Jun 26;24(7):878. doi: 10.3390/e24070878. PMID: 35885101; PMCID: PMC9319918.
[17]. Liu, Y., Peng, J., Yu, J. J. Q., & Wu, Y. (2019). *PPGAN: Privacy-preserving Generative Adversarial Network*. arXiv preprint arXiv:1910.02007.
[18]. Arjovsky, M., Chintala, S., & Bottou, L. (2017). "Wasserstein GAN." *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 214–223. Retrieved from https://arxiv.org/abs/1701.07875
[19]. Mehta, S., Ghazvininejad, M., Iyer, S., Zettlemoyer, L., & Hajishirzi, H. (2020). "DeLighT: Deep and light-weight transformer". *arXiv preprint arXiv:2008.00623*. Retrieved from https://arxiv.org/abs/2008.00623
[20]. DeVries, T., & Taylor, G. W. (2017). "Dataset augmentation in feature space". *arXiv preprint arXiv:1702.05538*. Retrieved from https://arxiv.org/abs/1702.05538
[21]. Korath, A., Sukumaran, S., & Menon, G. A. (2024). Performance Improvement of Government Employees Through Artificial Intelligence. Journal of Technology, 12(10), 494–502