FINAL PROJECT REPORT

Advancing Data Security with Iterative Encryption and Decryption Using Adversarial Generative Networks

1. Introduction

This project focuses on improving data confidentiality and integrity using machine learning models combined with classical encryption. The system automates both encryption and decryption using LSTM models and enhances dataset richness through adversarial generative networks (GANs).

The Caesar cipher serves as the base encryption method, enabling iterative enhancement through ML.

2. Problem Statement

Traditional Caesar cipher encryption is easy to break using frequency analysis and brute force.

When handling large-scale data (lakhs of sentences), manual or rule-based decryption becomes ineffective.

Hence, there is a need for automated, intelligent, and scalable encryption–decryption mechanisms.

This project aims to automate decryption using deep learning and adversarial networks while building a real-time chat communication system for demonstration.

3. Objectives

• Generate large-scale plaintext datasets using the NLTK corpus.

• Create 26 Caesar cipher datasets (keys 1–26).

• Train per-key LSTM models for decrypting ciphertext automatically.

• Use GAN-based data enhancement to strengthen model accuracy and robustness.

• Develop a ChatMessenger application supporting secure communication.

4. Dataset Description

• Plaintext generated using NLTK English words.

• 1 crore+ sentences can be generated (scalable).

• For each plaintext, 26 ciphertext versions are produced using Caesar cipher shifts.

• Data stored in CSV format containing plaintext–ciphertext pairs.

5. System Architecture

Step 1: Dataset Generation

• NLTK words → Random plaintext sentences.

• 26 Caesar cipher datasets created.

Step 2: Preprocessing

• Character-level encoding (a–z + space).

• Fixed word length padding.

Step 3: LSTM Decryption Models

• One LSTM model per Caesar shift key.

• Input: cipher word (character indices).

• Output: predicted plaintext word.

Step 4: GAN Component

• Generator creates synthetic cipher patterns.

• Discriminator distinguishes real vs. generated patterns.

• Increases dataset diversity and model accuracy.

Step 5: ChatMessenger Application

• User signup/login.

• Random shift per word encryption.

• Model-based automated decryption.

• Real-time secure communication.

6. Methodology

1. Generate plaintext dataset using NLTK corpus.

2. Apply Caesar cipher to create ciphertext datasets for keys 1–26.

3. Convert text into numerical sequences.

4. Train 26 separate LSTM models.

5. Apply GAN-based iterative enhancement.

6. Validate accuracy and integrate results into ChatMessenger.

7. Results

• Successfully trained decryption models for multiple keys.

• Automated decryption accuracy increases with GAN-enhanced datasets.

• ChatMessenger demonstrates practical secure communication.

8. Risks and Challenges

• Training 26 models requires computational resources.

• GAN networks may become unstable during training.

• Handling unseen vocabulary reduces model accuracy.

• Compatibility issues (SavedModel vs. .h5) in Keras 3.

9. Conclusion

This project modernizes classical Caesar cipher security by integrating advanced machine learning models,

LSTM-based decryption, and GAN-driven dataset enhancement. The ChatMessenger application demonstrates

a real-world implementation of automated encryption and decryption, providing improved confidentiality,

integrity, and scalability for secure digital communication.