National College of Ireland

Configuration Manual

MSc Research Project
M.Sc. in Cyber Security

Rohit puligadda
Student ID: 23223715.

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	**Rohit Puligadda**
Student ID:	**23223715**
Programme:	**M.Sc. in Cyber Security** **Year:** **2024-2025**
Module:	**Practicum**
Lecturer:	**Michael Prior**
Submission Due Date:	**12-12-24**
Project Title:	**Implementing Homomorphic Encryption for Privacy-Preserving Cloud Communication in Healthcare Systems**
Word Count:	1013 **Page Count:** 2

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: **Rohit Puligadda**

Date: **12-12-24**

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

Penalty Applied (if applicable):	

Configuration Manual

Rohit Puligadda
23223715.

1 Introduction

Welcome to the Secure Healthcare Data Management System Manual. This system allows healthcare organizations to securely upload, store, and manage sensitive patient data using industry-standard encryption techniques. The system leverages AES (Advanced Encryption Standard) for file encryption and Homomorphic Encryption via TenSEAL to ensure data privacy during processing. The uploaded files are stored in Firebase Cloud Storage, ensuring accessibility and scalability.
This manual is designed to guide users through every aspect of the system, from installation to troubleshooting.

1.1 System Requirements

Hardware Requirements:

- ✓ Processor: Intel Core i5 or equivalent (Recommended: i7 or higher)
- ✓ Memory: 8 GB RAM (Recommended: 16 GB)
- ✓ Storage: 100 GB free space (Recommended: SSD for faster processing)
- ✓ Network: Stable Internet connection for cloud storage integration

Software Requirements:

- ✓ Operating System: Linux (Ubuntu 20.04 or later), Windows 10
- ✓ Python: Python 3.8 or higher
- ✓ Django: Django 3.0 or higher
- ✓ Firebase Admin SDK: For Firebase Storage integration
- ✓ TenSEAL: For homomorphic encryption
- ✓ Cryptography Library and Pycryptodome: For AES encryption
- ✓ Web Browser: Google Chrome, Firefox, or Safari (for accessing the web interface)

1.2 Installation Guide

Step 1: Unzip the files to your working directory and open the directory in command prompt or terminal and type the following command

```
cd healthcare-encryption-system
```

Step 3: Create virtual environment

```
python -m venv venv
```

Step 4: Activate virtual environment
Windows

```
venv\scripts\activate
```

Linux

```
source venv/bin/activate
```

Step 5: Install Python Dependencies

```
pip install -r requirements.txt
```

Step 6: Setup Firebase

- ➢ Create a Firebase project in the Firebase Console.
- ➢ Enable Firebase Storage in the Firebase Console.
- ➢ Download the Firebase Admin SDK private key and save it to your project directory as firebase-key.json.

Step 7: Configure Django Settings
In the settings.py file, update the following:

```
FIREBASE_CONFIG_PATH = 'path/to/your/firebase-key.json'
FIREBASE_STORAGE_BUCKET = 'your-firebase-storage-bucket-url'
```

Step 8: Run Migrations to set up the database:

```
python manage.py makemigrations
python manage.py migrate
```

Step 9: Start the Django Development Server Run the development server to test the application:

```
python manage.py runserver
```

Step 10: Open your browser and go the

http://127.0.0.1:8000/api/upload/

2 System Architecture

The system is composed of the following components:

- Frontend (Web Interface): A simple user interface for uploading files and viewing their status.
- Backend (Django Framework): Handles HTTP requests, encryption, file uploads, and data management.
  - Patient Record Model: Stores patient information, including encrypted diagnosis data.
  - File Upload API: Handles the uploading of encrypted files to Firebase.
  - Encryption Methods:
- AES Encryption: Used to encrypt files before uploading.
- Homomorphic Encryption: Used for secure processing of sensitive data without exposing it.
- Firebase Cloud Storage: Stores the encrypted files securely in the cloud.

3 User Guide

3.1 Uploading Files

1. Navigate to the Upload Page: Open the web interface and go to the Upload File section.
2. Select a File: Click the Choose File button to select a file from your local system.
3. Encrypt the File: When a file is uploaded, it is automatically encrypted using AES encryption before being sent to Firebase.
4. Receive Confirmation: After the file is uploaded, you will receive a public URL for the file along with the encryption key and IV (Initialization Vector) used during encryption.

3.2 Viewing Uploaded Files

Once a file is uploaded, the public URL is provided to the user. This URL can be used to access the encrypted file in Firebase. However, the file can only be decrypted using the correct key and IV.

3.3 Encrypting and Decrypting Data

- Encryption: When a file is uploaded, the system uses AES encryption to encrypt the data. Additionally, the diagnosis data of a patient is encrypted using TenSEAL to perform homomorphic encryption.
- Decryption: To decrypt the file, the user must provide the key and IV used during encryption. The decryption process will reverse the AES encryption. For homomorphic encryption, the system uses TenSEAL's decryption function.

4 Security Features

4.1 AES Encryption

AES encryption is a symmetric encryption algorithm used to securely encrypt data. The system generates a random 256-bit key and 128-bit IV for each file uploaded. AES encryption in CBC (Cipher Block Chaining)

mode is used for file encryption, ensuring that even if a portion of the data is intercepted, it cannot be decrypted without the proper key and IV.

4.2 6.2 Homomorphic Encryption (TenSEAL)

Homomorphic encryption allows data to be processed while it is still encrypted. This method ensures that sensitive data, such as a patient's diagnosis, can be processed (e.g., analyzed, stored, or transmitted) without exposing it in its raw form. The system uses TenSEAL, a library that implements homomorphic encryption to protect data even during computation.

5 7. Troubleshooting

5.1 Issue 1: File Upload Fails

Solution: Ensure that the file size does not exceed the maximum allowed size in Firebase. Also, check your internet connection and ensure Firebase credentials are correct.

5.2 Issue 2: File Decryption Issues

Solution: Ensure you are using the correct key and IV for decryption. If you lost the key, it is impossible to decrypt the file.

5.3 Issue 3: System Performance Degradation

Solution: Large files can take longer to upload and decrypt due to the encryption overhead. Consider optimizing the file sizes or using a more powerful machine for better performance.

6 FAQ

1) What encryption methods are used in this system?
2) The system uses AES encryption for file data and TenSEAL for homomorphic encryption of sensitive patient data.
3) How can I decrypt an uploaded file?
4) You can decrypt the file using the key and IV provided during the upload process. This key is essential for restoring the original file content.
5) Is the data stored in Firebase secure?
6) Yes, the data is securely encrypted before being uploaded to Firebase. The encryption key and IV are not stored in Firebase, ensuring that unauthorized users cannot access the data without the correct credentials.

7 Glossary

- AES: Advanced Encryption Standard, a symmetric encryption algorithm.
- IV: Initialization Vector, used in cryptographic algorithms to ensure uniqueness.
- Homomorphic Encryption: A form of encryption that allows computation on encrypted data without decrypting it.
- Firebase: A cloud-based platform for storing and managing data, including files.