Enhancing Biometric Security using Homomorphic Encryption

Minor Project Report

Submitted by:

Vikas Kumar Saini (2023MCS2492)

Supervised by:

Dr. Vireshwar Kumar

Department of Computer Science & Engineering
IIT Delhi
DATE SUBMITTED
[June 24, 2024]

Abstract

The increasing reliance on biometric data for security and authentication has highlighted the importance of protecting such sensitive information from unauthorized access and potential leaks, especially when stored on cloud platforms. This project addresses the critical need for securing iris biometric data by leveraging the homomorphic encryption schemes.

The primary objective is to ensure that biometrics are encrypted before being stored in the cloud, thereby maintaining the confidentiality and integrity of the biometric information. By applying homomorphic encryption, this approach not only safeguards the data during storage and transmission but also enables secure processing and verification directly on the encrypted data. This eliminates the need for decryption in cloud environments, thereby minimizing the risk of exposure.

The project involves integrating the CKKS encryption scheme from the TenSEAL library into the biometric system, encrypting the iris biometric data, and evaluating the system's performance in terms of security, efficiency, and practicality for real-world applications. The successful implementation of this project will contribute to enhancing the security framework for biometric systems, providing a robust solution for protecting sensitive biometric data in cloud environments.

I. Introduction

Biometric authentication [6] systems are increasingly being adopted for their ability to provide robust and reliable security solutions. Among various biometric modalities, iris recognition stands out due to its high accuracy and resistance to forgery. However, the sensitive nature of biometric data necessitates stringent measures to protect it from unauthorized access and potential breaches, particularly when stored on cloud platforms. Traditional encryption methods, while offering some degree of protection, fall short when it comes to performing computations on encrypted data without exposing it.

To address this challenge, this project explores the use of the CKKS homomorphic encryption scheme, implemented through the TenSEAL library [3], to secure iris biometric data. Homomorphic encryption allows computations to be carried out on ciphertexts, generating an encrypted result that, when decrypted, matches the result of operations performed on the plaintext. This property is particularly advantageous for biometric data, as it enables secure processing and verification without the need to decrypt the biometric data at any stage.

By integrating CKKS encryption as a blackbox into the biometric system, this project aims to ensure that iris biometrics are encrypted before being stored in the cloud, thus maintaining the confidentiality and integrity of the data. The project's scope includes encrypting iris biometric data, storing it securely on the cloud, and evaluating the performance of the encryption scheme in terms of security, efficiency, and practicality for real-world applications.

This innovative approach not only enhances the security of biometric systems but also demonstrates the practical viability of homomorphic encryption for protecting sensitive data in cloud environments. The successful implementation of this project is expected to set a precedent for future research and development in the field of secure biometric authentication systems.

II. Background

A. Homomorphic Encryption (HE) [1]

An Encryption Scheme E, is called Homomorphic over an operation * if it supports the equation: $E(m_1) * E(m_2) = E(m_1 * m_2), \forall m_1, m_2 \in M$, where M is message space. This allows computations to be performed on ciphertexts, generating an encrypted result that, when decrypted, matches the result of operations performed on the plaintext. This enables secure data processing without exposing the underlying data.

There are mainly 3 types of HE Schemes *Table 1* on the basis of allowed operation types and their number of usages: **Partially HE (PHE), Somewhat HE(SHE)** and **Fully HE(FHE)**. Operations are typically limited to addition and multiplication operations because these operations are fundamental building blocks for more complex computations. Homomorphic encryption schemes are often built on algebraic structures like groups, rings, and fields. In these structures, addition and multiplication are well-defined and can be efficiently implemented, while other operations might not have such straightforward implementations. For example, RSA and ElGamal encryption schemes leverage multiplicative properties of certain algebraic groups. Some popular HE schemes are listed in *Table 2*.

Types	**Allowed Operation Types**	**Number of Usages**
1. PHE	One	Unlimited
2. SWHE	Both	First: Unlimited Second: Fixed
3. FHE	Both	First: Unlimited Second: Unlimited

Table 1: Types of HE Schemes

HE is characterized by 4 functions: KeyGeneration(***KeyGen***), Encryption(***Enc***), Decryption(***Dec***) and Evaluate(***Eval***). *KeyGen* generate the keys for the scheme at the client-side, *Enc* and *Dec* performs encryption and decryption and *eval* performs homomorphic operation on the ciphertext. A simple HE scenario is shown in the Figure 1.

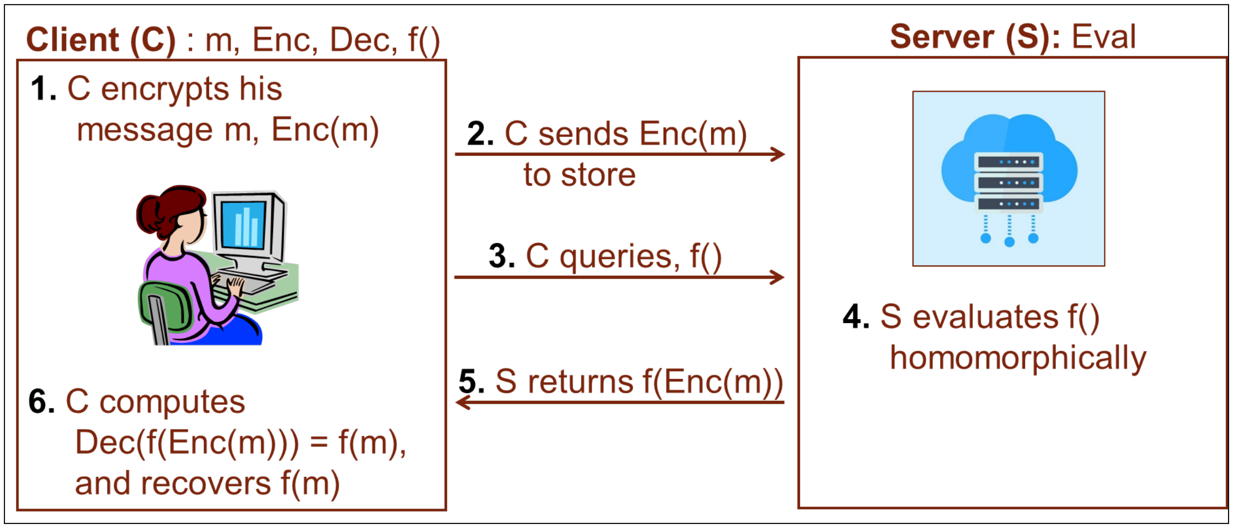

Client (C) : m, Enc, Dec, f()
1. C encrypts his message m, Enc(m)
6. C computes Dec(f(Enc(m))) = f(m), and recovers f(m)
2. C sends Enc(m) to store
3. C queries, f()
5. S returns f(Enc(m))
Server (S): Eval
4. S evaluates f() homomorphically

Figure 1: HE Client-Server Scenario depicting *Enc*, *Dec* and *Eval* on message m

HE Type	Popular Schemes
1. PHE	RSA, Paillier, Goldwasser-Micalli, EL-Gamal
2. SWHE	BGN, SYY
3. FHE	BGV, BFV, TFHE, CKKS

Table 2: Popular HE Schemes

A.1 The CKKS Scheme:

The CKKS (Cheon-Kim-Kim-Song) scheme is a SWHE scheme designed for encrypting and performing computations on encrypted data. It is specifically designed for approximating computations for applications involving real or complex numbers.

1. **Setup and Encoding:**
   - **Parameter Selection:** Similar to many lattice-based encryption schemes, CKKS necessitates parameter selection, including a cyclotomic polynomial ring of degree n and a modulus q, typically a power of 2.
   - **Encoding:** Real or complex numbers are encoded into plaintext polynomials within the chosen polynomial ring. This encoding process involves scaling by a large factor to enable approximations.

2. **Key Generation:**
   - **Secret Key:** A random polynomial with small coefficients (e.g., within a small integer range) is chosen as the secret key.
   - **Public Key:** The public key is generated using the secret key, selected parameters, and additional randomness, enabling the encryption of messages.
3. **Encryption:** The plaintext polynomial is encrypted using the public key to produce a ciphertext. Noise is intentionally introduced during encryption to enhance security, making the scheme approximate rather than exact.
4. **Homomorphic Operations:**
   - **Addition:** Encrypted numbers can be added by combining their respective ciphertexts. Unlimited number of addition operations are supported in ckks.
   - **Multiplication:** Multiplication of encrypted numbers involves multiplying their corresponding ciphertexts and subsequently performing a relinearization step to reduce size and noise in the result. Fixed number of multiplication operations are supported.
   - **Rescaling:** Post-multiplication, a rescaling step is typically applied to manage noise and maintain ciphertext integrity.
5. **Decryption:**
   - **Decrypting:** The secret key is used to decrypt the ciphertext and retrieve an approximate version of the original plaintext polynomial.
   - **Decoding:** The approximate plaintext polynomial is then decoded back into a real or complex number, reversing the initial encoding step.

A.2 The CKKS Scheme from TenSEAL library [3]:

1. **Setup and Encoding:** poly_modulus_degree = 8192 for 4096 bits of vector data,

coeff_mod_bit_sizes = [60, 40, 40, 60], global_scale = 2^{40}.

2. **Homomorphic Operations:** tenseal library has provided support for addition, multiplication, subtraction, etc. Thus during the eval of the iris encrypted data, one subtraction and one multiplication is required to perform XOR (a XOR b) = $(a - b)^2$ for 1 bit element of 4096 lenght vector.

B. Biometrics [6]

Biometrics refers to the measurement and statistical analysis of people's unique physical and behavioral characteristics. These characteristics are used to identify or verify their identity.

Types of Biometric Characteristics:

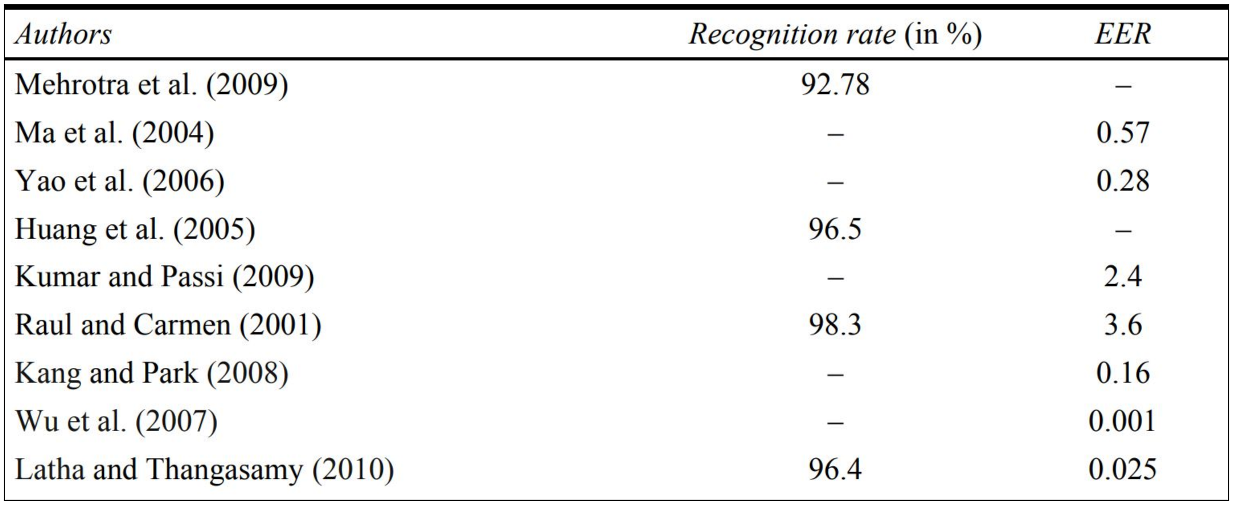
1. **Physiological Biometrics:**
   - **Fingerprint:** Analysis of unique patterns in the ridges and valleys of a person's fingerprints.
   - **Facial Recognition:** Measurement of facial features such as the distance between eyes, nose, and mouth.
   - **Iris Recognition:** Scanning of the iris for unique patterns like furrows and freckles.
   - **Retina Recognition:** Measurement of the unique patterns of blood vessels at the back of the eye.
   - **Hand Geometry:** Analysis of the shape and size of the hand.
2. **Behavioral Biometrics:**
   - **Voice Recognition:** Analysis of speech patterns, tone, and pitch.
   - **Keystroke Dynamics:** Measurement of typing rhythm and patterns.
   - **Gait Analysis:** Measurement of walking style and movement.

B.1 Iris Recognition:

Iris recognition [2 4 5] technology identifies individuals by analyzing the unique patterns visible within the iris of the eye. The iris is the colored part of the eye that surrounds the pupil. It contains intricate patterns of lines, pits, and freckles that form a unique texture. These patterns are formed randomly during fetal development and remain stable throughout a person's life, making iris recognition a reliable biometric method.

Iris Recognition Process:

1. **Image Acquisition:** Capturing a digital image of the iris.
2. **Image Processing:** Preprocessing the image(segmentation and normalization) to obtain the normalized iris image.
3. **Feature Extraction:** Specific features are extracted from the normalized iris image using 2D- Gabor wavelets.
4. **Template Creation:** The extracted features are encoded into a mathematical representation and then stored in a database.
5. **Matching and Verification:** Capture a new iris image, processes it to create a template, and then compares this template against the stored template associated with the person's identity.



Authors	*Recognition rate* (in %)	*EER*
Mehrotra et al. (2009)	92.78	–
Ma et al. (2004)	–	0.57
Yao et al. (2006)	–	0.28
Huang et al. (2005)	96.5	–
Kumar and Passi (2009)	–	2.4
Raul and Carmen (2001)	98.3	3.6
Kang and Park (2008)	–	0.16
Wu et al. (2007)	–	0.001
Latha and Thangasamy (2010)	96.4	0.025

Figure 2: Comparision table [8] for existing iris recognition

III. Problem Statement

The rapid adoption of biometric authentication systems, has introduced critical concerns regarding the security and privacy of sensitive biometric data stored on cloud platforms. Traditional encryption methods, while effective in securing data at rest, often require decryption for processing, posing inherent risks of exposure during computation. Moreover, the potential consequences of biometric data leaks, such as identity theft and unauthorized access, underscore the urgent need for advanced encryption techniques that can mitigate these risks.

If biometric data, such as iris patterns, were to be compromised and leaked, the consequences could be severe, potentially necessitating extreme measures like surgical alteration to mitigate identity theft risks. This highlights the criticality of implementing robust security measures to prevent such scenarios.

Therefore, this project aims to address these challenges by leveraging the CKKS homomorphic encryption scheme from the TenSEAL library to securely encrypt iris biometric data before storage on cloud platforms. The primary goal is to enable secure processing and verification of biometric data without the need for decryption, thereby minimizing the risk of data exposure and ensuring the integrity of sensitive information. By evaluating the performance and feasibility of CKKS encryption in real-world applications, this project seeks to provide a reliable solution for enhancing the security framework of biometric authentication systems against potential data breaches and their severe consequences.

IV. Proposed Solution

Homomorphic encryption is a sophisticated cryptographic technique that allows mathematical operations to be performed directly on encrypted data without the need for decryption. By applying homomorphic encryption to the biometric template, the data remains encrypted at all times, both during storage and during computation processes such as matching.

After encryption, the encrypted biometric templates are securely stored in the database. This step ensures that even if unauthorized access occurs, the encrypted data remains unintelligible without the appropriate decryption keys, thereby safeguarding sensitive biometric information from potential breaches and unauthorized use.

Furthermore, the matching process within the biometric system is adapted to operate directly on the encrypted templates. This approach eliminates the need to decrypt the data before performing matching operations, thereby preserving the confidentiality and integrity of the biometric information throughout the authentication process.

A. Procedure:

1. **Image Acquisition [7]:** Online available Iris Database - Phoenix - UPOL is used. The database contains 3 x 128 iris images (i.e. 3 x 64 left and 3 x 64 right). The images are: 24 bit - RGB, 576 x 768 pixels, file format: PNG. The irises were scanned by TOPCON TRC50IA optical device connected with SONY DXC-950P 3CCD camera.
2. **Segmentation [5]:** Thresholding is used initially to detect the circular frame of the image. Subsequently, another round of thresholding is employed to identify the outer boundary of the iris. Finally, thresholding is applied to the red channel of the image to detect the pupil.
3. **Normalization [2]:** Daugman's Rubber sheet model is used to obtain the normalized iris image of 64 X 512 pixel size.

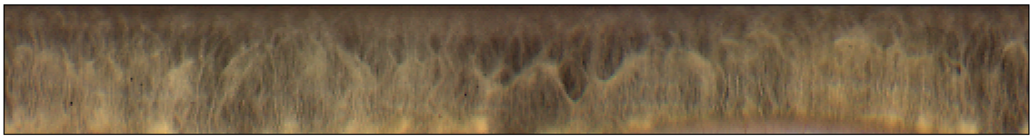


Figure 3: Normalized Iris Image 64 x 512 pixels

4. **Feature Extraction [2 4]:**

- **Haar Wavelet Decomposition:** This is used to reduce the normalized image size to 16 X 128 pixel size while preserving the potential texture information.
- **2D-Gabor Wavelets:** Real and Imaginary part of gabor wavelet is generated with kernel size = 9X9, frequency = 0.2 and orientation = $\pi/4$. These are then individually convolved with normalized image to obtain the feature vectors.

5. **Template Creation** [2]**:** Phase-Quadrant Demodulation is performed to obtain the iris template of 4096 bits.
6. **Image Registration:** CKKS HE scheme from tenseal library is used to encrypt the iris template and store it against the unique ID in the database. Procedure is shown in figure 4.

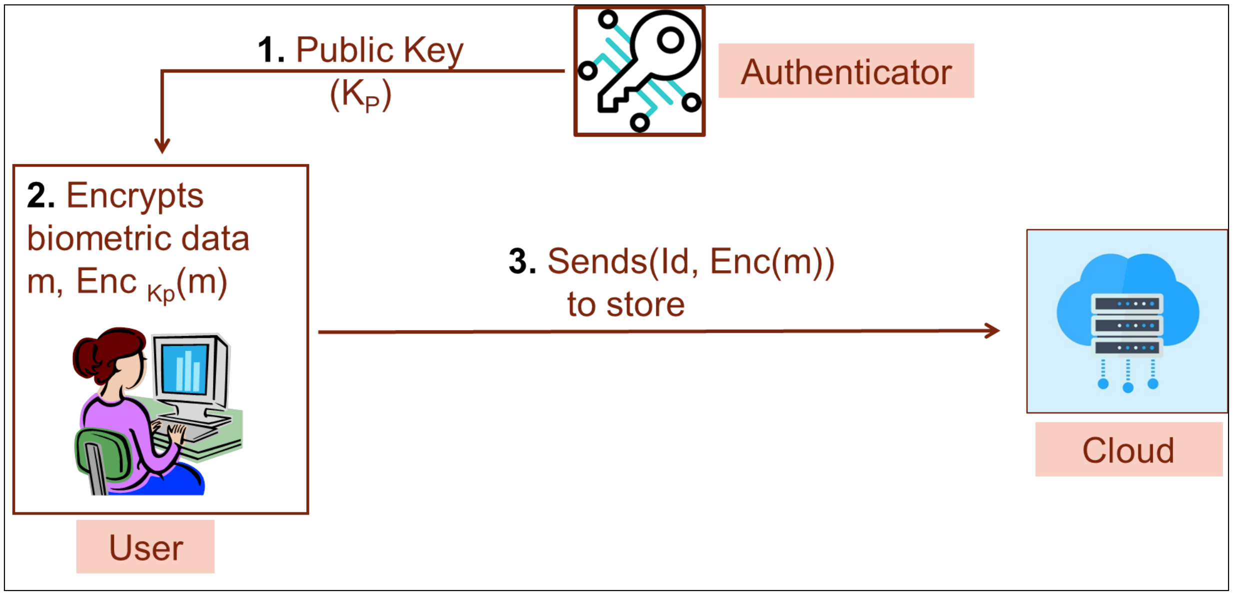

1. Public Key
(K_P)
Authenticator
2. Encrypts biometric data m, Enc $_{Kp}$(m)
3. Sends(Id, Enc(m)) to store
User
Cloud

Figure 4: Image Registration

7. **Authentication** [2 3]**:** Procedure is shown in figure 5.
   - Iris template of image to be authenticated is **encrypted** using CKKS scheme.
   - **XORing** of this template is homomorphically performed against the requested ID template in the database and resultant encrypted vector is sent back to authenticator where after **decrypting** the result **Hamming distance** is calculated.

- Resultant Hamming distance is **compared against the EER**(Equal Error rate) value and Result of Matching is declared on its basis.

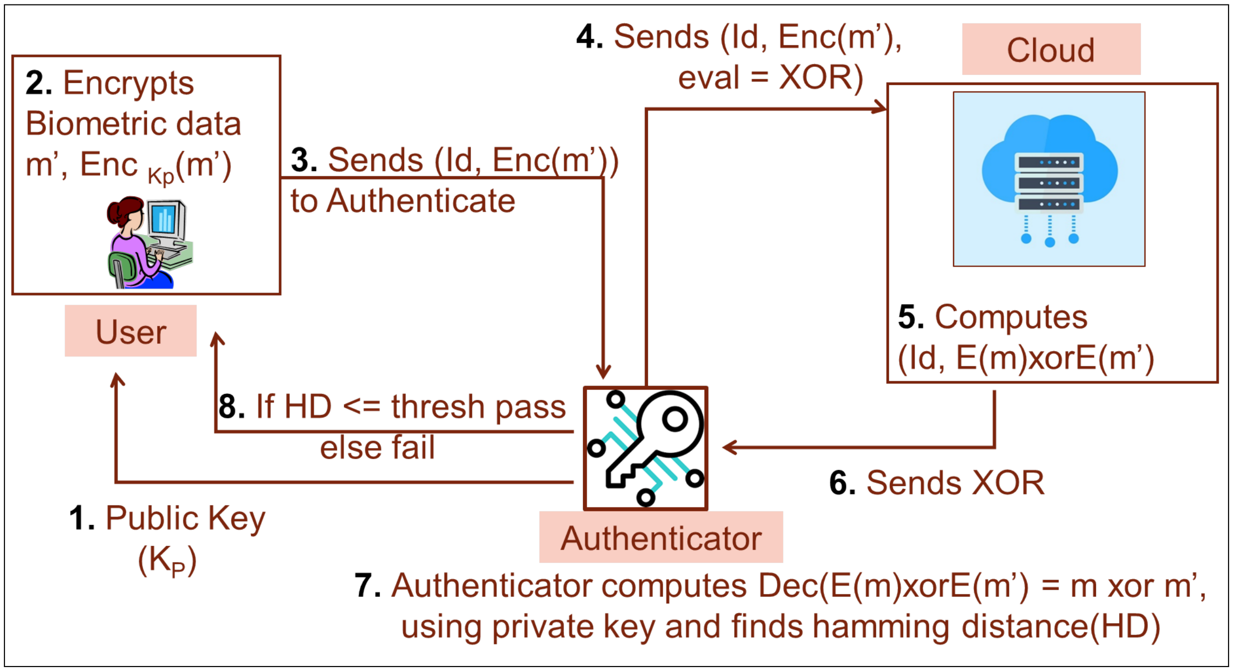

4. Sends (Id, Enc(m'), eval = XOR)
Cloud
2. Encrypts Biometric data m', Enc $_{Kp}$(m')
3. Sends (Id, Enc(m')) to Authenticate
User
5. Computes (Id, E(m)xorE(m')
8. If HD <= thresh pass else fail
6. Sends XOR
1. Public Key (K_P)
Authenticator
7. Authenticator computes Dec(E(m)xorE(m') = m xor m', using private key and finds hamming distance(HD)

Figure 5: Image Authentication

V. Evaluation [4]

Number of images used for registration and authentication is shown in $Table - 3$. Number of genuine and imposter images is shown in $Table - 4$. Parameter used for evaluation of the iris recognition system is Equal Error Rate(EER). It's value is shown in $Table - 5$. The average time required for 1 image registration and authentication is shown in $Table - 8$ (this also includes the time to read and write the vector in the text file and reading the keys from the text file). $Table - 6$ only shows the time taken by ckks schemes without reading, writing any files, while $Table - 7$ includes time with storing them on the disk. From the graph in $figure - 6$ the calculated EER value is 0.409.

Total Registered Images	$64 \times 2 \times 1 = 128$
Total images for Testing Authentication	$64 \times 2 \times 2 = 256$
Total Authentications Performed	$256 \times 128 = 32,768$

Table 3: Image tally for Registration and Authentication

Genuine	256
Imposters	$32,768 - 256 = 32,512$

Table 4: Image tally for Genuine and Imposter Authentications

Parameter	**value**
EER	0.409

Table 5: EER value

Process	Avg. time (in ms)
KeyGen	155.71
Enc	4.07
Eval	3.79
Dec	0.0014

Table 6: Ckks avg. time

Process	Avg. time (in ms)
KeyGen	155.71
Encrypt and store 1 vector	163.37
Eval vector	212.77
Dec vector	143.54

Table 7: Ckks avg. time with storing files on disk

Process	Avg. time for 1 image (in ms)
Registration	483.21
Authentication	879.52

Table 8: Avg. time

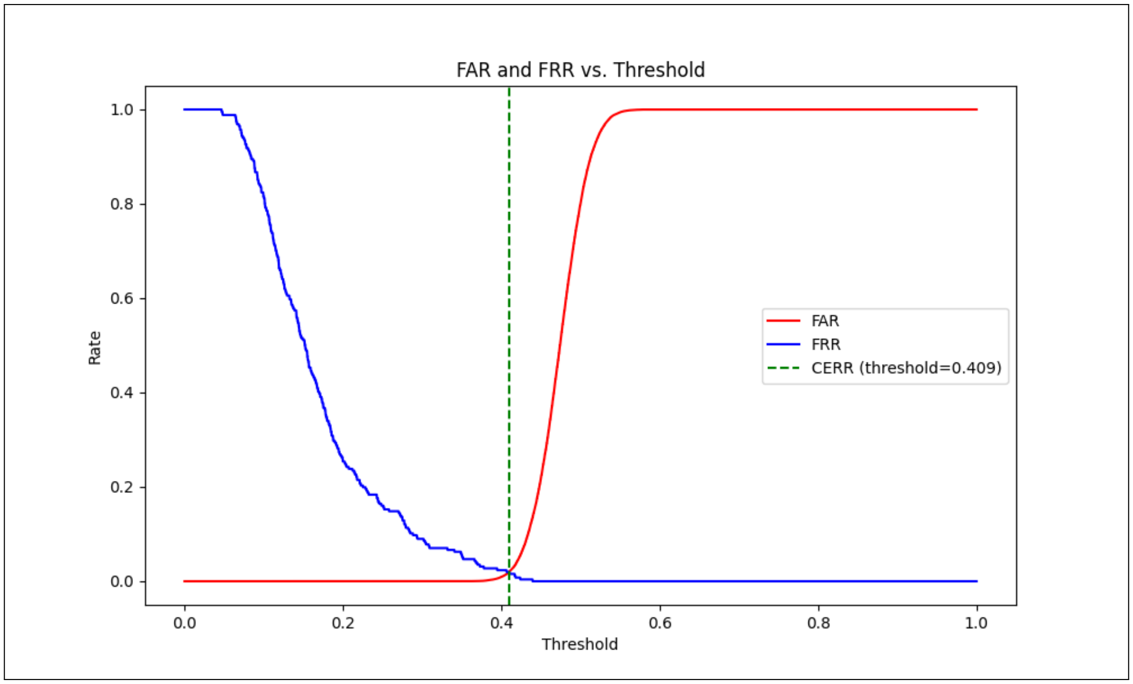


Figure 6: FAR, FRR vs Threshold and EER Graph

VII. Discussion

In the course of addressing the challenges and proposing innovative solutions outlined in this work, several considerations regarding limitations and opportunities for future research have emerged. This section aims to explore these aspects in detail, highlighting areas for improvement and potential avenues for extending the scope of our findings.

A. Limitations:

- Segmentation of iris is not modular. Current segmentation process works only for the UPOL iris dataset.
- Current EER value is high compared to other existing iris recognition techniques.
- Additional time is consumed for Homomorphic Encryption. Homomorphic operations add an overhead to the efficiency of the iris recognition system. This is a trade-off for securing the biometric data over cloud.

B. Future Work:

- Implementing modular Segmentation process for iris to make it to work on various different data sets.
- Reducing the iris template size and EER value by implementing Gabor filters for different Scales. This will improve the encryption time and EER value.
- Implementing with our own HE library with multiple schemes instead of tenseal library.
- Implementing for other Biometrics(thumb, face, etc.) also.

VII. Conclusion

Protecting biometric data, particularly iris recognition data, is crucial in today's security landscape, especially with the growing use of cloud storage. This project addresses this need by utilizing the CKKS homomorphic encryption scheme from the TenSEAL library to secure biometric templates. By encrypting iris biometric data homomorphically, we ensure that the data remains protected during both storage and computation, overcoming the limitations of traditional encryption methods that require decryption for processing.

The proposed solution integrates homomorphic encryption into the standard biometric authentication workflow. After preprocessing steps like segmentation, normalization, and feature extraction, iris templates are created and encrypted using the CKKS scheme. These encrypted templates are securely stored in the cloud, maintaining their confidentiality even if unauthorized access occurs. The authentication process operates directly on the encrypted data, preserving the integrity of the biometric information without needing decryption at any stage.

This project demonstrates the feasibility and effectiveness of securing iris biometric data with homomorphic encryption. By ensuring the biometric data remains encrypted at all times, the proposed system significantly enhances the security and reliability of biometric authentication systems. This approach provides a robust and practical solution for protecting sensitive biometric information on cloud platforms, paving the way for future advancements in secure biometric technologies.

VIII. References

[1] A. ACAR, H. AKSU, A. S. ULUAGAC and M. CONTI ”A Survey on Homomorphic Encryption Schemes: Theory and Implementation,” ACM Computing Surveys Volume 51 Issue 4 Article No.: 79, pp 1–35, 2018.

[2] J. Daugman, ”How iris recognition works,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 21-30, 2004.

[3] A. Benaissa, B. Retiat, B. Cebere, A.E. Belfedhal, ”TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption”, ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML 2021).

[4] R. Vyas, T. Kanumuri and G. Sheoran, ”Iris recognition using 2-D Gabor filter and XOR-SUM code,” 2016 1st India International Conference on Information Processing (IICIP), Delhi, India, pp. 1-5, 2016.

[5] A. Ghali, Abdulrahman & Jamel, Sapiee & Mohamad, Kamaruddin & Y. Abubakar, Nasir & M. Deris, Mustafa. (2017). A Review of Iris Recognition Algorithms. JOIV : International Journal on Informatics Visualization. 1. 175. 10.30630/joiv.1.4-2.62.

[6] S. Liu and M. Silverman, ”A practical guide to biometric security technology,” in IT Professional, vol. 3, no. 1, pp. 27-32, 2001.

[7] M. Dobeš and L. Machala, Iris Database, [Online] Available: `http://phoenix.inf.upol.cz/iris/` [Accessed: June,2024].

[8] Latha, L. & Thangasamy, Sangarappan. . ”Efficient method of person authentication based on fusion of best bits in left and right irises”, International Journal of Biometrics. 4. 203-219, 2012. 10.1504/IJBM.2012.047640.