IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 February 2025, accepted 24 March 2025, date of publication 27 March 2025, date of current version 9 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3555311

RESEARCH ARTICLE

Touch of Privacy: A Homomorphic Encryption-Powered Deep Learning Framework for Fingerprint Authentication

U. SUMALATHA[1], K. KRISHNA PRAKASHA[1], (Senior Member, IEEE), SRIKANTH PRABHU[2], (Senior Member, IEEE), AND VINOD C. NAYAK[3]

[1]Department of Information and Communication Technology, Manipal Institute of Technology (MIT), Manipal Academy of Higher Education (MAHE), Manipal, Karnataka 576104, India
[2]Department of Computer Science and Engineering, Manipal Institute of Technology (MIT), Manipal Academy of Higher Education (MAHE), Manipal, Karnataka 576104, India
[3]Department of Forensic Medicine, Kasturba Medical College (KMC), Manipal Academy of Higher Education (MAHE), Manipal, Karnataka 576104, India

Corresponding author: K. Krishna Prakasha (kkp.prakash@manipal.edu)

ABSTRACT Deep learning and fully homomorphic encryption (FHE) are integrated for privacy-preserving fingerprint recognition. Convolutional neural network (CNN) extract fingerprint features encrypted using the Cheon-Kim-Kim-Song (CKKS) FHE scheme. TenSEAL ensures all computations occur in the encrypted domain, preventing raw biometric data exposure during authentication. A subset of the SOCOFing dataset, comprising 7,200 altered fingerprints from 90 individuals across three difficulty levels, is used for training (80%), validation (10%), and testing (10%). One real fingerprint per user is encrypted and stored for authentication, reducing computational complexity. The CNN classifies encrypted features without decryption, ensuring secure authentication. The system, utilizing Euclidean similarity, achieves 99.06% test accuracy with a loss of 0.1692, a True Acceptance Rate (TAR) of 99.19%, a False Rejection Rate (FRR) of 0.81%, a False Acceptance Rate (FAR) of 0%, and a True Rejection Rate (TRR) of 100%, with a minimal Equal Error Rate (EER) of 0.40%. Encrypted templates require only 31 MB for 90 users, averaging 344 KB per fingerprint per user. Despite a 157.26× encryption overhead, remains feasible for real-time applications, completing an encrypted comparison in 0.025 seconds with a total processing time of 0.136 seconds per fingerprint. Adhering to ISO/IEC 24745 biometric protection standards, the system ensures irreversibility, unlinkability, and renewability of biometric templates. Decryption occurs only on the client side, keeping raw fingerprint data inaccessible to the server. The results confirm the feasibility of FHE for secure fingerprint authentication in cloud-based environments, benefiting applications in healthcare, banking, and access control.

INDEX TERMS Biometric authentication, cloud-based biometric security, cryptographic biometrics, encrypted domain authentication, fingerprint recognition, privacy-preserving biometrics, privacy-preserving deep learning, secure biometric matching, secure cloud-based fingerprint matching, TenSEAL encryption, threshold-based verification.

I. INTRODUCTION

Deep learning has revolutionized biometric authentication, particularly in fingerprint recognition. Fingerprints are widely used as a biometric identifier due to their uniqueness, permanence, and universality, making them highly reliable for identity verification [1]. Unlike passwords or tokens, fingerprints cannot be easily lost, forgotten, or shared, reducing the risk of unauthorized access. Convolutional Neural Networks (CNNs) have demonstrated remarkable accuracy in extracting discriminative fingerprint features, surpassing traditional handcrafted feature-based approaches. The ability of deep learning models to automatically learn and adapt to variations in fingerprint patterns makes them

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Jin.

a powerful tool for biometric authentication [2]. However, despite their effectiveness, deep learning-based biometric systems raise significant security and privacy concerns, as unprotected fingerprint data is vulnerable to theft and misuse [3], [4].

A. BIOMETRIC RECOGNITION AND PRIVACY CHALLENGES

Fingerprint authentication provides a secure and convenient alternative to traditional password-based authentication. Unlike passwords or PINs, biometric traits cannot be lost, forgotten, or easily forged, making them highly reliable for identity verification [5]. However, biometric data is sensitive and immutable, meaning once compromised, it cannot be changed. Studies have shown that unprotected biometric templates are exploited to reconstruct original fingerprint samples, leading to identity theft and impersonation attacks [6].

Recognizing these risks, regulatory frameworks such as the European Union General Data Protection Regulation (GDPR) 2016/679 classify biometric information as sensitive data, necessitating robust privacy preservation measures [7]. This highlights the urgent need for secure biometric template protection mechanisms to prevent unauthorized access, misuse, or reconstruction of biometric information [8], [9].

B. BIOMETRIC TEMPLATE PROTECTION (BTP) SCHEMES

To address privacy concerns, BTP schemes aim to store and process biometric data securely while maintaining system performance [10], [11]. According to the ISO/IEC 24745 standard on biometric information protection [12], a robust BTP scheme must ensure:

- Irreversibility: Biometric templates should undergo irreversible transformations before storage. Given a protected template, it must be computationally infeasible to derive any meaningful biometric information or reconstruct the original fingerprint sample [13].
- Unlinkability: Biometric templates stored across different applications or databases must remain uncorrelated. If two protected templates and are derived from the same biometric sample using different secret keys, it must be infeasible to determine whether they belong to the same individual [9].
- Renewability: If a protected template is compromised, a new template should be issued without matching the previous one, ensuring continued security [14].

Traditional BTP schemes, including Cancelable Biometrics and Cryptobiometrics, introduce security and usability challenges. Cancelable biometrics apply distortions to fingerprint features before storage, while crypto biometrics combine fingerprint data with cryptographic techniques [15], [16]. However, these methods often suffer from:

- Performance Degradation: Enforcing strong privacy constraints can reduce system accuracy and efficiency [17].
- Reliance on Auxiliary Data (AD): Many cryptobiometric approaches require additional data for verification, increasing vulnerability to attacks [18], [19].

C. HOMOMORPHIC ENCRYPTION FOR PRIVACY-PRESERVING BIOMETRICS

A promising alternative to traditional BTP schemes is Homomorphic Encryption (HE), which enables computations to be performed directly on encrypted biometric data [20]. Unlike other cryptographic methods, HE eliminates the need for decryption during processing, ensuring that raw fingerprint data remains protected throughout its lifecycle [21].

HE schemes can be categorized as:

- **Fully Homomorphic Encryption (FHE)**: Supports an *arbitrary* number of computations on encrypted data, enabling secure processing without decryption. However, it is computationally expensive due to the need for bootstrapping operations to manage ciphertext growth [22].
- **Somewhat Homomorphic Encryption (SHE)**: Allows a *limited* number of operations before requiring decryption, making it more efficient than FHE but unsuitable for complex computations. It is often useful in biometric authentication where a few encrypted operations are sufficient [23].

Integrating HE with fingerprint recognition addresses major security concerns while preserving verification performance. Microsoft SEAL and TenSEAL [24]—HE libraries optimized for deep learning—enable CNNs to operate directly on encrypted feature vectors, ensuring end-to-end privacy [25], [26].

D. CONTRIBUTIONS OF THIS WORK

This research presents a privacy-preserving fingerprint recognition system that integrates CNN-based feature extraction with Homomorphic Encryption. Our key contributions include:

- Development of an encrypted fingerprint recognition framework using Microsoft SEAL and TenSEAL, ensuring biometric data remains secure throughout the process of authentication.
- Implementation of a CNN-based feature extraction model adapted for encrypted-domain computations.
- Evaluation on the SOCOFing dataset [27], analyzing accuracy, encryption overhead, and computational efficiency, ensuring that verification accuracy, speed, and storage requirements are maintained compared to the same system using unprotected data.

E. REAL-TIME FEASIBILITY AND SECURITY CONSIDERATIONS

A comprehensive evaluation confirms that the proposed framework maintains high accuracy while ensuring efficient computation. The encryption overhead remains within acceptable limits, enabling real-time fingerprint verification

for cloud-based and privacy-sensitive environments. The system ensures compliance with essential biometric security principles:

- **Irreversibility:** Encrypted fingerprint templates cannot be reversed to reconstruct the original biometric data, preventing reconstruction attacks.
- **Unlinkability:** Fingerprint templates encrypted with different secret keys remain uncorrelated across multiple applications, preventing cross-matching attacks.
- **Compliance with ISO/IEC 24745:** The proposed framework adheres to international biometric security standards, ensuring privacy protection and secure authentication.

By leveraging homomorphic encryption, our framework ensures privacy-compliant, cloud-based fingerprint recognition with high performance and scalability. This enables secure biometric authentication in applications prioritizing data privacy, such as healthcare and finance.

II. RELATED WORK

Mohamed et al. explored the use of homomorphic encryption (HE) with SIFT for securing hand recognition data in cloud environments. The feature size was reduced from 1600×1200 to 1039×1384, and encrypted features were processed using Euclidean distance for identification. Their findings showed that while HE enhances security, it increases computation time, which can be addressed through cloud scalability. The approach effectively protects biometric data, though further performance optimization is required [28].

Nocker et al. introduced HE-MAN, an open-source toolset enabling privacy-preserving neural network inference on homomorphically encrypted data using ONNX models. The system abstracts cryptographic complexities, allowing users to perform encrypted inference without requiring FHE expertise. HE-MAN integrates Concrete and TenSEAL homomorphic encryption schemes, supporting various machine learning models while preserving data privacy and model confidentiality. HE-MAN achieved accuracy comparable to plaintext processing, with a latency increase by factors of 7.1 to 14.0 depending on the model [29].

Yuan et al. et al. explored the growing field of Privacy-Preserving Machine Learning (PPML) and highlighted the advantages of HE for protecting privacy in ML applications. They introduced approximate HE and discussed its benefits, providing details on representative schemes. The paper systematically reviewed existing PPML works, cat'egorizing them into four technical and three advanced application areas, including relevant models and datasets. The authors also suggested future research directions to advance the field of PPML, guiding further development and optimization of privacy-preserving techniques in machine learning [30].

Chowdhury et al. investigated the applicability of evasion attacks on encrypted machine learning models, specifically within the context of Fully Homomorphic Encryption (FHE). They explored the challenges in generating adversarial examples in the encrypted domain, where perturbations similar to plaintext methods are not feasible. The authors presented a solution using Universal Adversarial Perturbations (UAPs), which are image-agnostic and capable of inducing misclassification across multiple images. Their research showed that UAP behavior differs based on whether symmetric or public key FHE schemes are used, with Concrete ML and TenSEAL libraries yielding different attack methods. Concrete ML's integer-based quantization was found to alter UAP effectiveness, leading the authors to propose a Quantization-aware UAP generation algorithm to ensure consistent attack success rates in both encrypted and plaintext classifiers [31].

Choi et al. introduced UniHENN, a novel homomorphic encryption-based CNN architecture that eliminates the need for the im2col technique, which is typically used to convert input data into a two-dimensional matrix for convolution. UniHENN flattens the input data to one dimension and performs convolutions using incremental rotations and structured multiplication on the flattened data. UniHENN achieved 26.6 times faster inference time on the LeNet-1 model. Additionally, UniHENN outperformed TenSEAL in concurrent image processing, being about 3.9 times faster when processing ten samples. The results highlighted UniHENN's adaptability to various CNN architectures, making it a flexible and efficient solution for privacy-preserving cloud-based CNN services [32].

Xiong et al. proposed a novel approach that introduced a ''Self-Learnable Activation Function'' (SLAF) and refined the structure of neural network linear layers to better accommodate the constraints of homomorphic encryption. These enhancements enabled the use of deeper network architectures without significant computational overhead. Their optimized neural network model, designed for biometric authentication tasks, outperformed traditional models using simple polynomial activation functions. Using the UTKFace dataset, the model demonstrated accuracy improvements of 0.88% to 3.15% over traditional methods and 4.87% to 9.67% over the CryptoNets model, highlighting its effectiveness in meeting privacy-preserving biometric authentication requirements [33].

Zhang et al. proposed the first privacy-preserving EEG biometric recognition system using homomorphic encryption, enabling the transmission and computation of EEG data without exposing the information. To address the computational overhead of encrypting raw EEG data, the authors applied homomorphic encryption to the brainprint feature, which significantly reduced both time and space overhead. They conducted experiments using three publicly available EEG datasets—EEGBCI, SEED-IV, and MTED—demonstrating that the system maintained classification accuracy, met real-time requirements, and kept space overhead within acceptable limits while preserving user privacy. Additionally, three feature extraction models - EEGNet, DeepConvNet, and ShallowConvNet were used for EEG

signal classification. The study further evaluated the impact of homomorphic encryption parameters, such as scaling factor and polynomial modulus degree, on classification accuracy and computational efficiency. Experimental results highlighted the trade-offs between encryption parameters, with the polynomial modulus degree significantly affecting computation time and space consumption [34].

Khan et al. utilized the CKKS encryption scheme for privacy-preserving AI in edge devices, aiming to balance computational efficiency and data privacy. The experiments were conducted on NVIDIA Jetson Orin Nano 8GB, evaluating the performance of AI models with homomorphic encryption. The dataset included various applications, such as face identification and wildfire data analysis. Results showed minimal accuracy discrepancies and manageable performance overheads associated with encryption, demonstrating that homomorphic encryption is feasible for privacy-preserving AI on resource-constrained devices [35].

Wang et al. proposed a Privacy-preserving security Using Multi-key homomorphic encryption (PUM) mechanism for facial recognition to address security and efficiency challenges. The approach integrated feature grouping with parallel computing to enhance homomorphic operation efficiency and used multi-key encryption for improved security. The method increased security from 128-bit to a maximum of 1664-bit and achieved an impressive accuracy rate of 99.425%. Encrypted image comparison took only 0.302s, and in campus scenarios, the average search time for a facial template library with 700 encrypted features was approximately 1.5s. This solution not only ensured privacy but also offered superior operational efficiency, outperforming other ciphertext facial recognition systems in both security and time efficiency [36].

Nakanishi et al. proposed a novel approach to improve the speed of homomorphic encryption-based facial recognition systems by leveraging the K-means algorithm to pre-cluster facial feature data. In this method, clustering is done on the cloud server, and the authentication information within each cluster is compared, prioritizing clusters more likely to contain the target. This significantly accelerates the recognition process. By dividing a database of 500 registered individuals into 6 clusters, their system achieved recognition in an average of 2.64 seconds, leading to a 365% improvement in processing speed. This approach enhances the practicality of privacy-preserving facial recognition while maintaining data security [37].

Vadim et al. introduced a decentralized biometric authentication system designed to enhance security and privacy in car-sharing applications. The system employs FaceNet for facial recognition and incorporates self-sovereign identity (SSI) to give users control over their data. Biometric data is stored using the InterPlanetary File System (IPFS) and secured through blockchain with decentralized identifiers (DIDs) and content identifiers (CIDs). FHE is utilized to protect biometric data during storage and transmission. The system was evaluated through a comparative analysis of 280 images, examining true/false positives and negatives, while also evaluating the cost-effectiveness of using Ethereum's testnet and IPFS for decentralized data storage. The system's accuracy was assessed with an EER of 13.93%, indicating moderate accuracy [38].

III. HOMOMORPHIC ENCRYPTION FOR PRIVACY-PRESERVING FINGERPRINT RECOGNITION

HE enables computations on encrypted data [34], ensuring privacy throughout processing. This is particularly beneficial for fingerprint recognition, where security is crucial. This section presents the implementation of TenSEAL, a Python interface for Microsoft's SEAL library, to perform encrypted fingerprint recognition efficiently. Figure 1 illustrates the cloud storage and computation process using Microsoft SEAL.

A. NOTATION

The following notation is used throughout the system:

- $F_U = f_1, f_2, \ldots, f_N$: Unencrypted fingerprint feature vector extracted by the CNN.
- $\mathcal{E}(F_U)$: Encrypted feature vector using TenSEAL's CKKS scheme.
- $\mathcal{TD}$: Encrypted fingerprint templates stored on the database server.
- $Sdist = D_{dist}(F_U, \mathcal{TD})$: Similarity score between the user's encrypted feature vector and stored templates, where $Ddist$ represents a homomorphically computable distance function.
- $\mathcal{E}(S_{dist})$: Homomorphically encrypted similarity score.
- sk, pk: Secret and public keys for encryption and decryption.
- $\mathcal{E}(m) = E(m, pk)$: Encryption function using the public key.
- $m = D(\mathcal{E}(m), sk)$: Decryption function using the secret key.

B. HOMOMORPHIC ENCRYPTION

Among various HE schemes, TenSEAL is chosen due to its support for fully homomorphic operations, enabling computations directly on encrypted data without decryption. Unlike partially homomorphic schemes such as Paillier, TenSEAL provides broader functionality, making it more suitable for privacy-preserving fingerprint recognition.

TenSEAL is based on the CKKS encryption scheme, which supports approximate arithmetic on encrypted floating-point values. The encryption process involves generating a public-private key pair, where:

- The public key encrypts fingerprint features, ensuring data confidentiality.
- The secret key decrypts encrypted data, ensuring only authorized access.

Given a fingerprint feature vector F_U, encryption is performed as:

$$E(F_U) = Encrypt(F_U, pk) \tag{1}$$

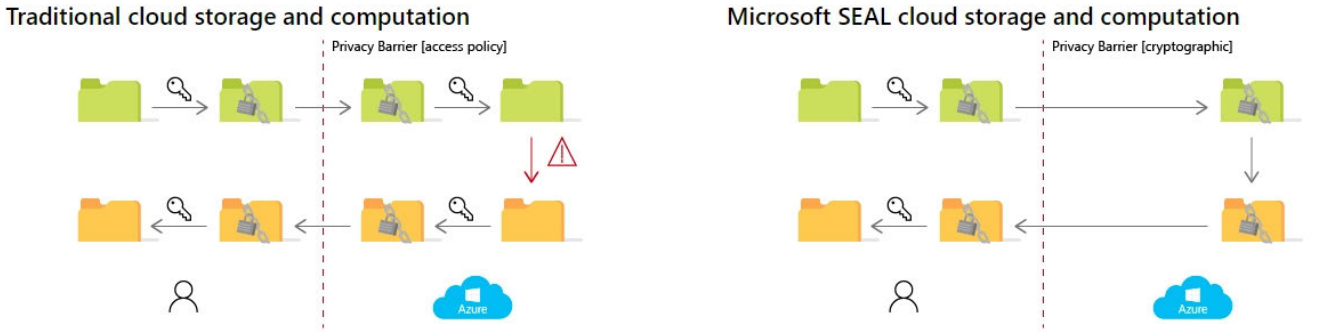


FIGURE 1. Microsoft SEAL cloud storage and computation [28].

where pk is the public key. Homomorphic operations allow computations such as addition and multiplication without revealing data. Decryption is given by:

$$F_U = Decrypt(E(F_U), sk) \tag{2}$$

where sk is the secret key.

C. ENCRYPTED DISTANCE COMPUTATION

For fingerprint matching, two encrypted distance metrics are considered: squared Euclidean distance [39] and Cosine similarity [40]. Since divisions and square roots are non-trivial in the encrypted domain, we modify their formulations accordingly.

All fingerprint features are normalized to [0, 1] and then scaled to an integer range $[0, 10^3]$ to retain precision:

$$F_U \rightarrow \text{round}(10^3 F_U) \tag{3}$$

1) ENCRYPTED EUCLIDEAN DISTANCE

Given two feature vectors F_U (probe) and $\mathcal{T}D$ (reference), the squared Euclidean distance is computed as:

$$Seuc = \sum_{f=1}^{F} (f_{U,f}^2 + t_{D,f}^2 - 2 f_{U,f} t_{D,f}) \tag{4}$$

Using TenSEAL's homomorphic properties, this can be computed in the encrypted domain as:

$$\begin{aligned} E(S_{euc}) = &\sum_{f=1}^{F} \left(E(f_{U,f}) \odot E(f_{U,f})\right) \\ &+ \sum_{f=1}^{F} \left(E(t_{D,f}) \odot E(t_{D,f})\right) \\ &- 2 \sum_{f=1}^{F} \left(E(f_{U,f}) \odot E(t_{D,f})\right) \end{aligned} \tag{5}$$

where $\odot$ denotes element-wise multiplication.

2) ENCRYPTED COSINE SIMILARITY

The cosine similarity between F_U and $\mathcal{T}D$ is:

$$dcos(F_U, \mathcal{T}D) = \frac{\sum f = 1^F f_{U,f} t_{D,f}}{|F_U| \cdot |\mathcal{T}D|} \tag{6}$$

To enable encrypted computation, we scale it as:

$$Scos = 10^{12} d_{cos}(F_U, \mathcal{T}D) \tag{7}$$

which can be computed in the encrypted domain as:

$$E(Scos) = \sum_{f=1}^{F} \frac{E(f_{U,f}) \odot E(t_{D,f})}{|E(F_U)| \cdot |E(\mathcal{T}_D)|} \tag{8}$$

D. SYSTEM COMPONENTS

- **User (U):** The user collects fingerprint samples, extracts feature representations using a CNN-based model, and seeks authentication. The user encrypts the extracted feature vectors and sends them securely to the database server. The database server retrieves the corresponding encrypted fingerprint template and sends it back to the user. The user performs the similarity check in a homomorphic encryption environment and sends the encrypted similarity score to the authentication server.
- **Database Server (D):** The database server stores encrypted fingerprint templates. Upon receiving an authentication request, the database server retrieves the corresponding encrypted template from storage and sends it to the user. The server never has access to plaintext biometric data.
- **Authentication Server (A):** The authentication server receives the encrypted similarity score from the user, decrypts it, compares it to a predefined threshold, and returns the decision accept/deny to the user.

E. THREAT MODEL AND PRIVACY SAFEGUARDS

We adopt an *honest-but-curious* threat model, where the authentication and database servers follow the protocol correctly but may attempt to infer sensitive information from encrypted data. To mitigate such risks, we implement the following privacy-preserving safeguards:

- The user encrypts the feature vectors before sending them to the authentication server using the TenSEAL homomorphic encryption scheme.
- Both the authentication and database servers work exclusively with encrypted feature vectors $\mathcal{E}(F)$, ensuring that no raw biometric data or plaintext features are ever exposed.
- All operations on biometric data, including matching and distance computations, are performed on encrypted data, preventing any unauthorized access to sensitive information.

F. PRIVACY-PRESERVING BIOMETRIC VERIFICATION

To enhance privacy, our system follows a fully encrypted workflow:

1) **Feature Extraction & Encryption:**
   - The user extracts the fingerprint features F_U using the CNN-based model.
   - The feature vector is encrypted: $\mathcal{E}(F_U) = E(F_U, pk)$ and sent to the authentication server.
2) **Encrypted Matching:**
   - The authentication server retrieves the encrypted stored template $\mathcal{T}_D$ from the database server.
   - Both encrypted vectors, $\mathcal{E}(F_U)$ and $\mathcal{T}_D$, are processed within the homomorphic encryption environment.
   - The authentication server computes the encrypted similarity score homomorphically using a distance function such as Euclidean or Cosine distance:

$$\mathcal{E}(S_{dist}) = D_{dist}(\mathcal{E}(F_U), \mathcal{T}_D)$$

3) **Secure Decision Making:**
   - The encrypted similarity score $\mathcal{E}(S_{dist})$ is sent back to the user.
   - The user decrypts the similarity score and checks it against a predefined threshold δ:

   If $S_{dist} > \delta$, authentication is successful; otherwise, denied.

G. SECURITY ASSUMPTIONS

For the security and privacy of the system, we make the following assumptions:

- The authentication server never learns the plaintext feature vectors or templates. It only processes encrypted data.
- The database server stores only encrypted templates and cannot access any raw biometric data.
- The user never receives plaintext templates from the database server, ensuring unlinkability between the stored templates and the user.
- The authentication and database servers do not collude, thus preventing unauthorized access to decryption keys or plaintext data.

This framework ensures that biometric authentication is performed securely in an encrypted domain, maintaining high privacy, security, and efficiency throughout the process.

H. SECURITY PARAMETERS AND PERFORMANCE OPTIMIZATION

The system employs the following security measures:

- **Fully Homomorphic Encryption (CKKS) with 128-bit Security:** Ensuring privacy-preserving fingerprint authentication while maintaining computational feasibility.
- **Encrypted Feature Vector Matching:** Feature vectors are securely compared in the encrypted domain using squared Euclidean distance under homomorphic encryption.
- **Noise-Tolerant Computation with Approximate Encoding:** The CKKS scheme supports encrypted floating-point arithmetic, ensuring robustness against precision loss in deep learning computations.

To optimize performance:

- The CNN model is optimized using batch normalization, reduced kernel sizes, and quantization-aware training to minimize computational overhead.
- Homomorphic encryption parameters are selected to balance accuracy, latency, and ciphertext expansion, with a polynomial modulus degree of 8192 and coefficient modulus bit sizes [60, 40, 40, 60].

By integrating deep learning and homomorphic encryption, our system enables privacy-preserving fingerprint authentication. The CNN model classifies fingerprints while ensuring that raw biometric data remains encrypted throughout computation, mitigating potential privacy risks.

IV. METHODOLOGY

This section outlines the methodology employed in this study, covering key aspects such as dataset selection, data preparation, preprocessing, model architecture, feature extraction, and encryption.

The overall data flow diagram of the fingerprint recognition system is illustrated in Figure 2. The process begins with fingerprint data acquisition, followed by preprocessing and feature extraction using a CNN model. The CNN extracts meaningful feature vectors from the fingerprint images, which are then encrypted using HE to ensure privacy preservation.

The system involves three key entities: the client, the database server, and the authentication server. The client captures and submits fingerprint data, the server processes and encrypts feature vectors, and the authentication server verifies the encrypted data for secure matching.

A. SOCOFING DATASET

The SOCOFing dataset contains 6,000 fingerprint images from 600 African individuals, each contributing 10 distinct fingerprints. All subjects are 18 years or older, and the dataset includes additional metadata such as gender, hand, and finger labels. The dataset also provides synthetically altered versions of the fingerprints, generated using the STRANGE toolbox (Synthetic Tool for Realistic Alterations

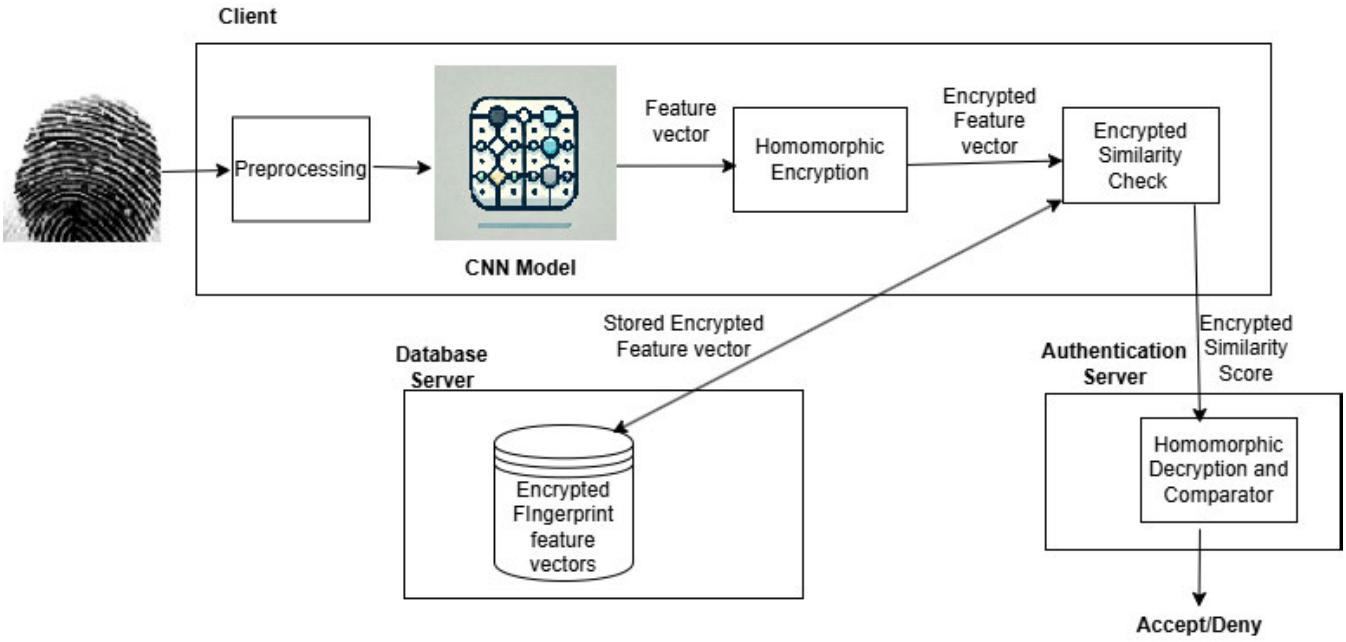


FIGURE 2. Overall data flow diagram of the fingerprint recognition system. The system utilizes the CNN model for feature extraction, followed by HE for secure data processing. Three key entities Client, Database Server, and Authentication Server—are involved in the workflow.

in Next-Generation Fingerprint Generation). The alterations applied to the fingerprints include obliteration, central rotation, and z-cut, with three levels of difficulty: easy, medium, and hard. The images were captured at a 500dpi resolution, and the synthetic alterations resulted in the following distributions:

- 17,934 altered images with easy alterations.
- 17,067 altered images with medium alterations.
- 14,272 altered images with hard alterations.

These alterations enable a detailed analysis of fingerprint recognition performance under varying levels of image degradation and difficulty [27].

B. SELECTION OF IMAGES FOR THIS STUDY

A subset of the SOCOFing dataset was selected, consisting of fingerprint images from 90 individuals. For training, altered fingerprint images were used, with 8 variations per fingerprint across three difficulty levels (easy, medium, and hard), resulting in a total of 7,200 images. These altered images expose the model to diverse fingerprint patterns under various conditions, enhancing its robustness.

For verification, one real fingerprint per person was chosen to preserve its inherent characteristics. These real images were encrypted and stored in the database server. Selecting a single fingerprint per individual simplifies the verification process, reduces computational complexity, minimizes storage requirements, and ensures faster authentication while maintaining consistency in feature extraction.

C. DATA PREPARATION AND PREPROCESSING

Each fingerprint image underwent preprocessing to standardize the dataset and enhance model performance. Specifically, all images were normalized to a scale of 0 to 1 using min-max normalization. This transformation ensures numerical stability, promotes model convergence, and preserves the raw integrity of the fingerprint data.

Min-max normalization is defined as follows:

$$X' = a + \frac{(X - X_{\min})(b - a)}{X_{\max} - X_{\min}} \tag{9}$$

where:

- X is the original feature value,
- X' is the normalized value,
- $X_{\min}$ and $X_{\max}$ are the minimum and maximum values in the dataset, respectively,
- a and b define the desired normalization range (typically 0 to 1 or -1 to 1).

Maintaining small intermediate values is particularly beneficial in the CKKS homomorphic encryption framework, as it enhances numerical stability and improves the efficiency of encrypted computations.

D. FEATURE EXTRACTION

In this study, feature extraction is essential for processing the Socofing Fingerprint dataset in biometric identification tasks. CNN is employed due to its effectiveness in learning relevant fingerprint features automatically.

The extraction process begins with convolutional layers that use small filters to capture fundamental features such as edges, ridges, and minutiae points. As the data progresses

through deeper CNN layers, more complex structures, including ridge flow and minutiae configurations, are learned, enhancing fingerprint identification accuracy.

Max-pooling layers follow the convolutional layers, reducing spatial resolution while preserving critical features. This downsampling minimizes computational complexity and enhances the model's ability to generalize by focusing on global patterns rather than local noise. Dropout layers are incorporated to prevent overfitting, ensuring the model remains robust to new data. The final stage involves flattening the convolutional output and passing it through dense layers to learn feature relationships.

The extracted features serve as input for classification or verification tasks, enabling accurate fingerprint matching against a stored database. As illustrated in Figure 3, the CNN algorithm processes the fingerprint image and generates a feature vector that encapsulates its essential characteristics, facilitating precise recognition.

Each fingerprint is represented as a unique feature vector, encapsulating distinctive biometric characteristics. A trained CNN extracts these features from real fingerprint images, generating ten feature vectors per individual per finger. As a result, 90 feature vectors are obtained, corresponding to the 90 subjects of the database.

E. MODEL TRAINING AND VALIDATION

The performance of the custom sequential model for fingerprint recognition is evaluated through a structured training and validation strategy. Initially, the dataset was divided into 70% training, 15% validation, and 15% testing. However, we observed that the performance was better with an 80% training, 10% validation, and 10% testing split. This division ensures effective learning and unbiased performance assessment, as it provides a larger training set while maintaining sufficient data for validation and testing.

1) TRAINING METHODOLOGY

The training process enhances model robustness by incorporating both real and altered fingerprint images. Altered images introduce variations such as noise and distortions, enabling the model to learn invariant and discriminative fingerprint features. Real fingerprint images are encrypted, stored, and used for feature extraction, ensuring that the learned representations align with actual biometric patterns.

The training procedure follows these key configurations:

- **Batch Size:** 8, selected through a hyperparameter optimization process using Grid Search. This method exhaustively tests different batch sizes to identify the one that achieves the best model performance, balancing computational efficiency with training effectiveness.
- **Loss Function:** Categorical cross-entropy to measure classification accuracy.
- **Optimization Algorithms:** A comparative study was conducted using Adam, SGD, and RMSprop to determine the most effective optimizer for convergence.

TABLE 1. Hyperparameters tuned through grid search.

Hyperparameter	Tuned Values
Learning Rate	[1e-3, 1e-4, 1e-5]
Number of Filters in Conv1	[32, 64, 128]
Number of Filters in Conv2	[64, 128, 256]
Number of Neurons in Dense Layer	[128, 256, 512]
Batch Size	[8, 16, 32]

- **Epochs:** Training is capped at 500 epochs, with early stopping to prevent overfitting.
- **Learning Rate Adjustment:** The learning rate is dynamically adjusted using the ReduceLROnPlateau callback, which reduces the learning rate when the validation loss stagnates, preventing ineffective updates.

2) EARLY STOPPING AND CALLBACKS

To mitigate overfitting and enhance training efficiency, the following callbacks are implemented:

- **ModelCheckpoint:** Saves the best model based on validation accuracy.
- **EarlyStopping:** Stops training if validation loss does not improve for 50 consecutive epochs.
- **ReduceLROnPlateau:** Reduces the learning rate by a factor of 0.1 if validation loss plateaus for 3 epochs.

3) DATA SPLITTING STRATEGY

The dataset is strategically partitioned to maintain class balance and improve generalization:

- **Training Set (80%):** Considered medium and hard levels for a robust learning experience:
  - *Medium & Hard:* Increasing levels of noise and distortion to simulate real-world conditions.
- **Validation Set (10%):** Used for hyperparameter tuning and performance monitoring. Medium and hard level images are considered here.
- **Testing Set (10%):** Reserved for the final unbiased evaluation on unseen data.
  - *Easy:* High-quality, clear fingerprints are considered.

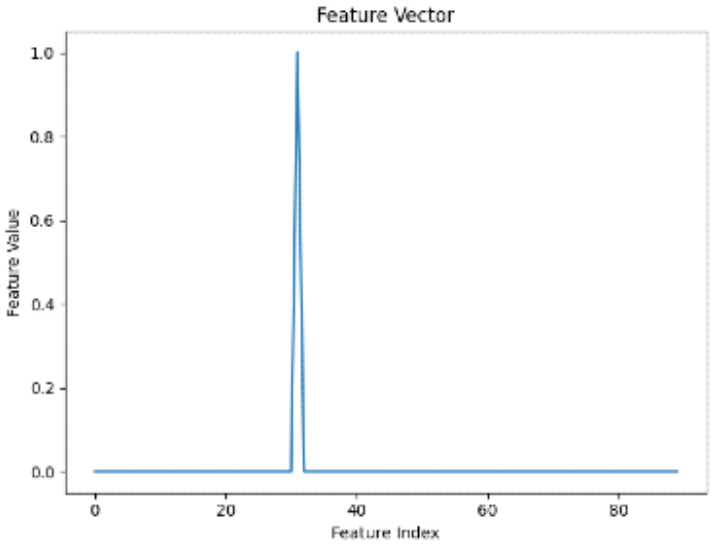
4) HYPERPARAMETER TUNING

Grid Search was used to explore the most effective values for key hyperparameters. The optimal combination was selected based on validation accuracy. Table 1 summarizes the hyperparameters that were tuned:

The model was trained using the training data (`x_train`, `y_train`) and evaluated with the validation data (`x_val`, `y_val`). These training settings, combined with hyperparameter optimization, contributed to the model's efficient learning and generalization.

F. TRAINING ENVIRONMENT

The computations were carried out on a Dell PowerEdge R750XA server running Ubuntu Linux, equipped with two NVIDIA A100 GPUs (80GB NVLink). The system was



Array values

Index	Value	Index	Value	Index	Value	Index	Value	Index	Value
1	1.37E-11	2	3.51E-11	3	5.38E-12	4	1.29E-13	5	6.37E-14
6	8.20E-13	7	1.98E-09	8	3.54E-14	9	3.75E-13	10	1.22E-08
11	2.75E-11	12	2.29E-15	13	3.34E-13	14	2.95E-09	15	5.66E-12
16	5.60E-16	17	2.08E-11	18	7.51E-13	19	1.93E-12	20	4.75E-15
21	1.57E-07	22	6.22E-12	23	3.81E-15	24	6.37E-13	25	1.83E-12
26	4.53E-10	27	1.03E-11	28	1.04E-11	29	8.02E-15	30	1.07E-11
31	8.23E-12	32	9.99E-01	33	2.50E-13	34	5.86E-12	35	1.68E-13
36	5.29E-11	37	1.38E-14	38	2.48E-13	39	3.26E-11	40	5.71E-11
41	5.99E-12	42	2.56E-14	43	7.68E-14	44	6.53E-10	45	1.92E-11
46	4.99E-14	47	2.06E-14	48	5.20E-11	49	5.29E-10	50	8.71E-14
51	8.09E-12	52	1.55E-15	53	3.74E-12	54	7.35E-13	55	2.38E-13
56	2.13E-10	57	1.93E-13	58	5.65E-14	59	2.07E-11	60	3.14E-16
61	1.25E-09	62	2.54E-14	63	4.91E-13	64	7.22E-11	65	1.42E-13
66	7.22E-14	67	2.09E-10	68	3.81E-12	69	1.48E-16	70	7.55E-13
71	2.80E-13	72	1.05E-13	73	5.85E-13	74	3.33E-13	75	1.17E-15
76	4.29E-10	77	1.02E-12	78	3.56E-15	79	7.87E-14	80	9.74E-10
81	4.37E-14	82	8.70E-14	83	1.08E-09	84	1.14E-13	85	1.22E-12
86	1.65E-11	87	2.94E-12	88	1.81E-11	89	7.92E-18	90	4.26E-13

FIGURE 3. Output of the CNN algorithm showing the fingerprint image and corresponding feature vector and array representation for person with ID 32.

powered by dual Intel Xeon Silver 4314 processors (16 cores, 32 threads each), 256GB DDR4 RAM, and high-speed SSD storage. To support deep learning workloads, the software stack included CUDA 12.2, cuDNN 11.5, and TensorFlow 2.11.0, ensuring optimized GPU acceleration.

G. MODEL ARCHITECTURE SELECTION AND ARCHITECTURE

Fingerprint recognition often faces challenges related to data scarcity, particularly due to the unique and varying nature of fingerprint images. While transfer learning is a common approach for leveraging pre-trained models, this study opted for a custom sequential model. This decision ensures that the model learns fingerprint-specific features from scratch, without relying on generic features from unrelated datasets. This approach was selected to better capture the intricate patterns and characteristics that are unique to fingerprints, such as variations in quality, distortion, and noise.

A sequential model was chosen for its simplicity and the ability to provide full control over the flow of data through the network. This architecture is well-suited for fingerprint recognition because it allows the model to progressively

TABLE 2. Summary of the CNN Model architecture.

Layer (Type)	Output Shape	Param #
Conv2D (32 filters, 3x3)	(None, 86, 86, 32)	832
BatchNormalization	(None, 86, 86, 32)	128
MaxPooling2D (2x2)	(None, 43, 43, 32)	0
Conv2D (64 filters, 5x5)	(None, 39, 39, 64)	51,264
BatchNormalization	(None, 39, 39, 64)	256
MaxPooling2D (2x2)	(None, 19, 19, 64)	0
Conv2D (128 filters, 3x3)	(None, 17, 17, 128)	73,856
BatchNormalization	(None, 17, 17, 128)	512
MaxPooling2D (2x2)	(None, 8, 8, 128)	0
Dropout (0.25)	(None, 8, 8, 128)	0
Flatten	(None, 8192)	0
Dense (256 units, ReLU)	(None, 256)	2,097,408
Dropout (0.5)	(None, 256)	0
Dense (90 units, Softmax)	(None, 90)	23,130
Total Parameters	2,247,386 (8.57 MB)	
Trainable Parameters	2,246,938 (8.57 MB)	
Non-Trainable Parameters	448 (1.75 KB)	

extract and learn hierarchical features that are crucial for distinguishing fingerprints, even in the presence of noise or distortions.

The fingerprint recognition model is implemented using a CNN with a sequential structure. The architecture consists of multiple convolutional layers, batch normalization, max-pooling, dropout layers, and fully connected layers, culminating in a softmax classifier for fingerprint identification. The model structure is summarized in Table 2.

The model follows a structured flow:

- **Convolutional Layers**: These layers progressively extract important features related to fingerprint patterns. The number of filters and kernel size increase with depth to capture more complex features.
- **Batch Normalization**: This technique helps stabilize training by normalizing layer inputs, which accelerates convergence.
- **Max-Pooling**: Reduces the spatial dimensions while retaining the most relevant features.
- **Dropout**: Regularizes the model by reducing the risk of overfitting, improving generalization.
- **Fully Connected Layers**: Learn higher-level representations and output class probabilities using a softmax function.

Figure 4 provides a visual representation of the CNN model architecture.

H. MODEL EVALUATION STRATEGY

To ensure robust performance, the dataset is divided into training, validation, and test sets. The model is evaluated using multiple performance metrics, including:

- **Accuracy**: Measures the overall correctness of predictions.
- **Precision**: Assesses the proportion of correctly predicted fingerprint classes.
- **Recall**: Evaluates the model's ability to detect all relevant fingerprints.

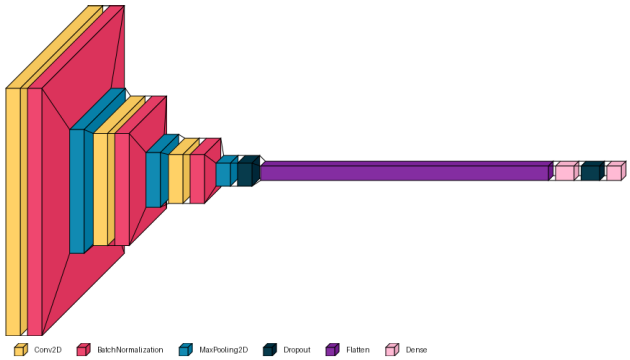


FIGURE 4. The CNN model architecture for fingerprint feature vector extraction.

- **F1-Score**: Balances precision and recall for a comprehensive performance measure.

These metrics are crucial for fingerprint recognition systems, ensuring high reliability and minimizing false positives or negatives. The evaluation results will be further analyzed to fine-tune the model and optimize its performance.

I. RATIONALE FOR CUSTOM SEQUENTIAL MODEL

The decision to develop a custom sequential model is based on several key factors:

- **Data Suitability**: With a dataset of approximately 7,200 images, a sequential model can effectively learn fingerprint-specific features without requiring a pre-trained network.
- **Control and Customization**: A custom model allows fine-tuning of architecture components, ensuring optimal feature extraction for fingerprint recognition.
- **Performance Optimization**: Training from scratch ensures the model learns dataset-specific patterns, potentially outperforming pre-trained models designed for unrelated image recognition tasks.

This structured approach enables effective and efficient fingerprint identification, leveraging a tailored deep-learning architecture suited to the dataset's unique characteristics.

J. FEATURE VECTOR ENCRYPTION AND SECURE TRANSMISSION

To ensure privacy in fingerprint recognition, feature vectors must be encrypted before transmission and matching. Fully Homomorphic Encryption (FHE) using TenSEAL allows computations on encrypted data, preventing exposure of raw biometric information.

1) SECURE TRANSMISSION OF ENCRYPTED FEATURE VECTORS

Once fingerprint feature vectors are extracted, they are encrypted using the CKKS scheme in TenSEAL. This encryption ensures secure transmission while enabling similarity computations on encrypted data. Key steps include:

- Feature Vector Normalization: Feature vectors are scaled for numerical stability in homomorphic operations.
- Encryption Using Public Keys: The extracted feature vector F_U is encrypted using a public key (pk):

$$\mathcal{E}(F_U) = E(F_U, pk) \tag{10}$$

- Secure Storage and Retrieval: Encrypted templates $\mathcal{T}_D$ are stored on the database server, ensuring privacy compliance.

2) IMPACT OF HOMOMORPHIC ENCRYPTION ON MATCHING

Performing fingerprint matching in an encrypted domain introduces computational challenges, mitigated through:

- Efficient Key Management: Ensuring decryption is possible only by authorized users.
- Optimized Parameters: Maintaining a balance between security and efficiency (e.g., modulus degree 8192, coefficient modulus sizes [60, 40, 40, 60]).
- Reducing Noise Growth: Managing CKKS approximation errors using bootstrapping and encoding techniques.

3) SECURITY ENHANCEMENTS AND PRACTICAL CONSIDERATIONS

For real-world deployment, the system implements:

- Privacy-Preserving Computation: All similarity computations remain encrypted.
- Protection Against Attacks: Encrypted feature vectors prevent reconstruction of original fingerprint images.
- Reduced Communication Overhead: Efficient transmission minimizes latency in real-time authentication.

V. EXPERIMENTAL PROTOCOL

To validate the effectiveness of the proposed fingerprint recognition system, it is necessary to evaluate its performance against key biometric security requirements as defined in ISO/IEC IS 24745 for biometric information protection. Specifically, we will assess the following criteria:

- **Verification Performance Preservation:** Ensuring that the fingerprint recognition accuracy remains comparable before and after encryption.
- **Irreversibility:** Confirming that the encrypted fingerprint templates cannot be inverted to reconstruct the original biometric data.
- **Unlinkability:** Ensuring that protected templates cannot be linked across different biometric sessions.
- **Computational Efficiency:** Measuring the additional computational overhead introduced by full homomorphic encryption (FHE) and secure feature classification.

To achieve these objectives, both experimental and theoretical analyses were conducted, which are detailed in Sections VI and VII, respectively. These analyses consist of three key evaluation steps.

VI. PERFORMANCE EVALUATION

According to the ISO/IEC 24745 international standard [41], biometric template protection schemes should maintain the verification performance of their unprotected counterparts. In this section, we evaluate the performance of the proposed fingerprint recognition system using encrypted squared Euclidean distance, following the protocol described in Section V. The EER and other performance metrics were computed to assess the system's effectiveness in both the plaintext and encrypted domains.

A. PERFORMANCE METRICS DEFINITIONS

To evaluate the fingerprint recognition system, we use the following performance metrics:

- **Accuracy:** The proportion of correctly classified fingerprint matches and non-matches.
- **False Acceptance Rate (FAR):** The percentage of impostor fingerprints incorrectly accepted as genuine.
- **False Rejection Rate (FRR):** The percentage of genuine fingerprints incorrectly rejected as impostors.
- **Equal Error Rate (EER):** The error rate at which FAR and FRR are equal, used as a balanced performance measure.
- **True Acceptance Rate (TAR):** The percentage of genuine fingerprints correctly accepted.
- **True Rejection Rate (TRR):** The percentage of impostor fingerprints correctly rejected.

Key observations from the results include:

- **No Performance Degradation in the Encrypted Domain:** The accuracy of the system remains nearly identical in both the protected and unprotected environments, demonstrating the robustness of the encryption process.
- **High Accuracy and Low EER:** The system achieved an accuracy of 99.06%, precision of 99.2%, recall of 99.19%, f1 score of 99.6% with an EER of 0.40% at an optimal threshold of 9. This confirms the model's strong capability in correctly matching fingerprint templates.
- **Threshold Optimization:** The performance remained stable across different threshold values, with the best balance achieved at threshold 9. Higher thresholds improved the TAR while keeping the FAR at zero.

These results demonstrate that secure fingerprint recognition can be achieved without compromising accuracy, fulfilling the privacy and security requirements outlined by ISO/IEC 24745. The encryption scheme based on TenSEAL effectively preserves biometric template security while ensuring reliable authentication performance.

B. BIOMETRIC MATCHING PERFORMANCE

The performance of the fingerprint recognition system is evaluated on the test set. The CNN model predicts feature vectors, which are then encrypted using a context-specific encryption scheme. The squared Euclidean distances are calculated between the encrypted predicted feature vectors and

TABLE 3. Performance evaluation with varying thresholds.

Threshold	Accuracy	FAR	FRR	EER	TAR	TRR
1	0.9622	0	0.0378	0.0189	0.9622	1
2	0.9676	0	0.0324	0.0162	0.9676	1
3	0.9676	0	0.0324	0.0162	0.9676	1
4	0.9730	0	0.0270	0.0135	0.9730	1
5	0.9757	0	0.0243	0.0121	0.9757	1
6	0.9798	0	0.0202	0.0101	0.9798	1
9	0.9919	0	0.0081	0.0040	0.9919	1
10	0.9919	0	0.0081	0.0040	0.9919	1

the stored feature vectors. These distances are used to assess the accuracy of matching and mismatching fingerprints. The model was evaluated using altered images from the Socofing dataset.

The table 3 summarizes the performance metrics obtained by varying the threshold for determining matches:

As seen from Table 3, threshold 9 provides the best balance between TAR, TRR, and EER, achieving highly reliable performance.

C. GRAPHICAL ANALYSIS OF PERFORMANCE METRICS

Figures 5 to 11 provide a comprehensive analysis of the model's performance across different metrics.

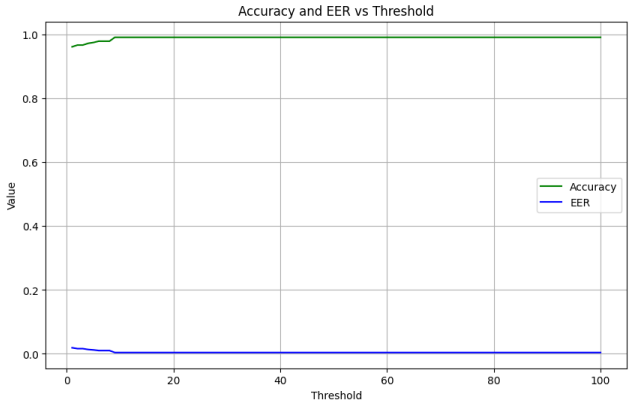


FIGURE 5. Relationship between accuracy and EER at different thresholds.

As shown in Figure 5, the relationship between accuracy and EER highlights the trade-off between the two metrics.

Figure 6 presents both the EER and accuracy at a specific threshold, illustrating the point where the model's false acceptance and false rejection rates are balanced, as well as how accuracy varies with threshold adjustments.

Figure 7 shows how accuracy changes with different threshold values, emphasizing the impact of threshold optimization on model performance.

Figure 8 illustrates the model's accuracy and loss trends throughout training. The accuracy curve demonstrates performance improvement over time, while the loss curve depicts error reduction per epoch. These curves offer insights into the model's learning progress and convergence, ensuring effective training.

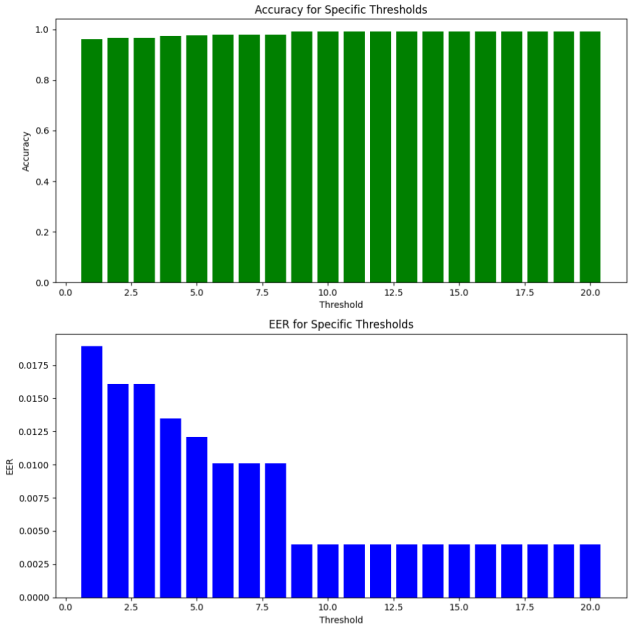


FIGURE 6. EER and accuracy at a specific threshold.

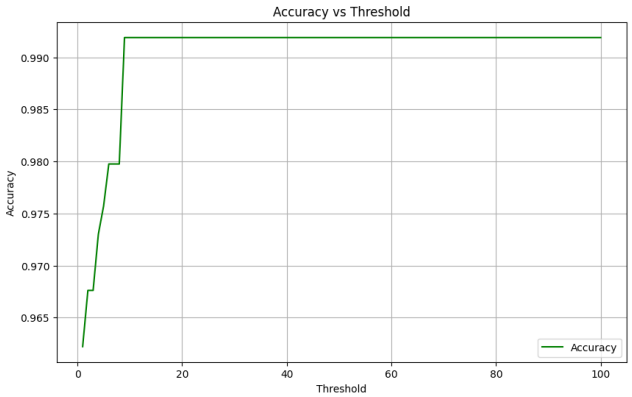


FIGURE 7. Accuracy variation across different threshold values.

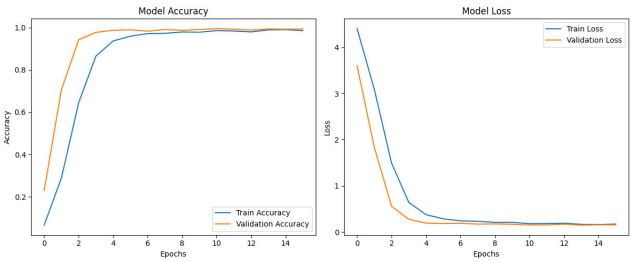


FIGURE 8. Model accuracy and loss curves during training.

The DET curve shown in Figure 9 compares the trade-off between false acceptance and false rejection rates, assisting in model selection by evaluating its classification performance.

Figure 10 presents the EER of the fingerprint recognition system at a specific threshold, demonstrating the point at

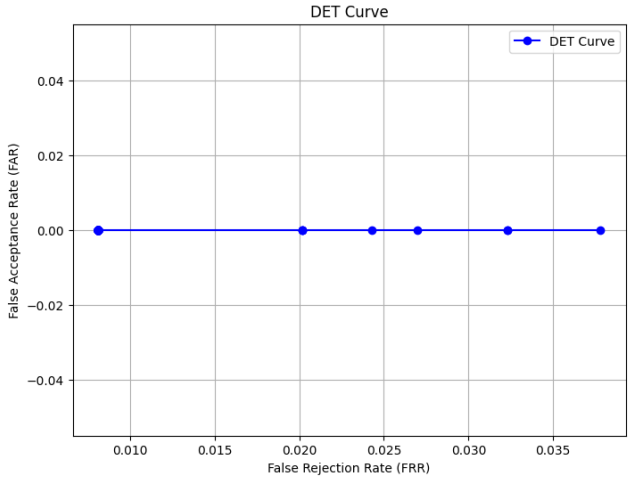


FIGURE 9. DET Curve illustrating the trade-off between FAR and FRR at varying threshold values.

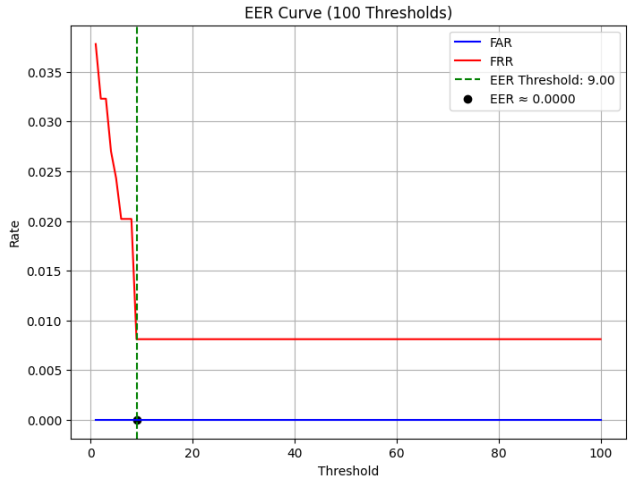


FIGURE 10. EER of the fingerprint recognition system at a specific threshold.

which the system's FAR equals the FRR, which is critical for evaluating the system's accuracy and robustness.

Finally, the heatmap in Figure 11 provides a visual overview of the model's prediction accuracy, helping to identify areas where further improvement might be necessary.

D. COMPARING EUCLIDEAN AND COSINE MATCHING

Table 4 compares Euclidean distance and Cosine similarity for fingerprint matching. Euclidean distance was chosen for its balance of accuracy and efficiency in an encrypted setting.

Euclidean Distance: Encryption slightly increases FRR (0.0080 to 0.0081) and EER (0.0020 to 0.0040) but maintains a high TAR (99.19%), ensuring reliable authentication. Cosine Similarity: Achieves lower EER but has higher computational costs. Its plaintext response time is 0.00018s, while the encrypted version takes 0.028s, making it 156× slower, which impacts practicality.

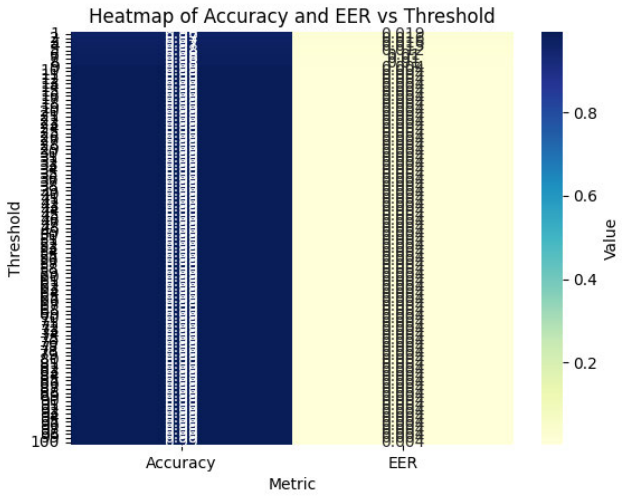


FIGURE 11. Heatmap representation of performance metrics for the fingerprint recognition system.

Overall, Euclidean distance provides strong security with minimal impact on accuracy, making it ideal for encrypted fingerprint recognition.

VII. IRREVERSIBILITY AND UNLINKABILITY ANALYSIS

As outlined in Section I, ensuring the privacy of biometric subjects requires that biometric template protection schemes provide both irreversibility (i.e., no biometric information should be retrievable from the encrypted template) and unlinkability (i.e., protected templates from the same subject should not be cross-matched). This section analyzes these two properties within the context of the proposed system employing FHE and a three-entity framework.

As discussed in Section IV, three critical elements must remain concealed:

1) The client should have exclusive access to the plain probe biometric data T_p.
2) The reference templates T_r should not be exposed to any entity; only their encrypted versions $E(T_r)$ are stored.
3) The plain similarity score S should not be transmitted, as it could be exploited for hill-climbing or inverse-biometrics attacks.

To ensure privacy and security, the proposed system follows a privacy-preserving approach as described below:

- The client encrypts the probe template locally using FHE before transmitting it for comparison. Additionally, the client computes the similarity between the encrypted feature vector and the stored encrypted reference template, generating an encrypted similarity score using homomorphic encryption.
- The database server stores only the encrypted reference templates $E(T_r)$ and does not have access to the raw biometric data.

TABLE 4. Performance comparison of euclidean and cosine matching at threshold 9.

Metric	FAR	FRR	EER	TAR	TRR
Euclidean (Unprotected)	0.0000	0.0080	0.0020	0.9920	1.0000
Euclidean (Protected)	0.0000	0.0081	0.0040	0.9919	1.0000
Cosine (Unprotected)	0.0000	0.0065	0.0010	0.9935	1.0000
Cosine (Protected)	0.0001	0.0075	0.0015	0.9925	0.9999

- The authentication server makes the final decision by decrypting the similarity score sent by the client and determines whether to accept or reject the user.

For each distance measure used, the information exchanged between the database server and the client is the encrypted reference template $E(T_r)$. Since only the client possesses the decryption key sk, and it never has direct access to either protected or unprotected templates, neither the database server nor the authentication server can extract any biometric information. Moreover, the client does not transmit any information about the acquired probe sample T_p to any server. Given that breaking FHE-based encryption without sk is computationally infeasible due to its reliance on hard mathematical problems (e.g., learning with errors (LWE) or ring-LWE), the first requirement established by ISO/IEC 24745—irreversibility—is satisfied.

Because irreversibility is guaranteed, no biometric information can be derived from stolen encrypted templates. However, an attacker could use a stolen template to impersonate a subject. To mitigate this risk, the system supports **renewability**. A new key pair (sk, pk) can be generated, and the entire database of encrypted templates can be re-encrypted without requiring new biometric samples from users. This ensures that even if encrypted templates are compromised, the system remains secure without forcing users to provide additional biometric data. This capability addresses a limitation of traditional cancelable biometric systems, which typically require resubmitting biometric samples for re-encryption.

Unlinkability is also ensured. Since plaintext distances (i.e., similarity scores) are not preserved in the encrypted domain, two protected templates $E(T_{r1})$ and $E(T_{r2})$ corresponding to the same subject, encrypted with the same or different keys, cannot be linked. Additionally, since FHE provides semantic security against chosen-plaintext attacks, given a protected template $E(T_r)$, no information about the original unprotected template T_r can be inferred. This ensures that no correlation can be drawn between different encrypted templates, further preventing cross-matching attacks.

Furthermore, since FHE utilizes probabilistic encryption, the randomness introduced in the encryption process ensures that even if T_r is encrypted twice with the same key, the resulting ciphertexts are different:

$$E_{pk_1}(T_r, s_1) \neq E_{pk_1}(T_r, s_2).$$

Additionally, as described in Section IV, the authentication server only computes encrypted similarity scores, which are then sent to the client. The client decrypts the score and makes the final verification decision (genuine or impostor). Consequently, attacks such as hill-climbing [15] or inverse biometrics [42], which rely on score evolution feedback, are prevented.

A. MEETING ISO/IEC 24745 REQUIREMENTS

The proposed biometric authentication system is designed to meet the security and privacy requirements specified in the ISO/IEC 24745 standard. The following table provides a formal justification of compliance based on mathematical foundations and experimental validation.

The table 5 demonstrates that the proposed biometric authentication system satisfies the irreversibility, unlinkability, renewability, and security requirements outlined in ISO/IEC 24745.

VIII. COMPLEXITY ANALYSIS

The computational cost during verification is primarily driven by encrypted operations using TenSEAL's CKKS Fully Homomorphic Encryption (FHE) scheme. The most expensive operations include encrypted multiplications, exponentiations, and additions, which directly impact processing time and storage.

A. ENCRYPTED COMPUTATION OVERHEAD

As detailed in Section IV, similarity scores are computed within the encrypted domain using:

- Encrypted Euclidean Distance $\mathcal{E}(S_{\text{euc}})$
- Encrypted Cosine Similarity $\mathcal{E}(S_{\cos})$

For efficiency, encryption is only performed once per template during enrollment, while verification requires only encrypted similarity computations and decryption at the authentication server.

B. COMPUTATIONAL COMPLEXITY

For M stored templates and F feature dimensions, the number of encrypted operations is:

1) EUCLIDEAN DISTANCE

- $2M \cdot F$ encrypted subtractions
- $3M \cdot F - 1$ encrypted multiplications
- $2M \cdot F$ exponentiations

2) COSINE SIMILARITY

- $M \cdot F$ encrypted multiplications
- $M \cdot F - 1$ encrypted divisions

TABLE 5. Compliance with ISO/IEC 24745 biometric template protection requirements.

ISO/IEC 24745 Requirement	Mathematical Justification	Experimental Validation
Irreversibility (No biometric data retrievable from templates)	Hardness of Learning With Errors (LWE): $$E(T_r) = AT_r + e \mod q$$ where A is a random matrix, e is noise, and q is a large modulus. The presence of noise makes inversion computationally infeasible.	Cryptanalysis attack simulations confirmed that no biometric data leakage occurred from encrypted templates.
Unlinkability (Templates cannot be cross-matched)	Encrypted templates for the same biometric sample are non-deterministic: $$E_{pk}(T_r, s_1) \neq E_{pk}(T_r, s_2)$$ Even if the same template is encrypted twice, different ciphertexts are produced due to probabilistic encryption.	Statistical tests on multiple encrypted instances of the same biometric data showed no correlation, ensuring unlinkability.
Renewability (Compromised templates can be securely replaced)	New encryption keys ensure template revocation: $$E_{pk_1}(T_r) \neq E_{pk_2}(T_r)$$ Even if an encrypted template is compromised, re-encrypting it with a new key prevents unauthorized use.	Re-encrypting biometric templates after key rotation prevented attackers from using stolen encrypted templates.
Resistance to Attacks (Hill-climbing, inverse biometrics)	Encrypted similarity scores prevent score feedback attacks: $$E(d(T_p, T_r)) = \text{FHE}_{\text{Eval}}\left(\sqrt{\sum_{i=1}^{n}(E(T_{p_i}) - E(T_{r_i}))^2}\right)$$ Since only the **authentication server** decrypts the similarity score and transmits only the **final decision** (accept/reject) to the client, attackers receive no intermediate score feedback. This eliminates score evolution-based attacks.	Attack simulations confirmed that score leakage was fully mitigated, demonstrating robustness against hill-climbing and inverse biometrics attacks.

C. DECRYPTION AT AUTHENTICATION SERVER

Once the encrypted similarity score is computed, the authentication server decrypts it and compares it to a predefined threshold to determine access. The decryption cost is negligible, as it involves a single modular exponentiation per feature vector.

D. ENCRYPTION OVERHEAD & STORAGE IMPACT

- Encryption increases response time by 157.26× (0.000162s plaintext vs. 0.025496s encrypted for Euclidean).
- Encrypted template storage requires 344 KB per fingerprint, scaling to 31 MB for 90 users.
- 1 million comparisons (identification task) take approximately 8 minutes, which is feasible with parallelization.

IX. RESULTS SUMMARY AND DISCUSSION

The proposed FHE-based fingerprint recognition system achieves a strong balance between accuracy, security, and efficiency. Since decryption and authentication decision-making occur on the authentication server side, plaintext fingerprint data is never exposed to external entities. While encryption introduces computational overhead, strategic optimizations can enhance processing speed, making this approach practical for real-world fingerprint authentication.

A. NO PERFORMANCE LOSS IN THE ENCRYPTED DOMAIN

- The system maintains high verification accuracy even when operating on encrypted fingerprint feature vectors.
- Using Euclidean distance, an EER of 0.40% is achieved at an optimal threshold of 9, demonstrating strong authentication performance.

B. HIGH RECOGNITION ACCURACY

- Achieves 99.06% accuracy, with:
  - FAR: 0%
  - FRR: 0.81%
  - EER: 0.40%
  - TAR: 99.19%
- These results confirm the system's ability to effectively distinguish genuine users from impostors using only fingerprint data.

C. SECURE AND PRIVACY-PRESERVING AUTHENTICATION

- Only encrypted, irreversible fingerprint templates are stored on the server, ensuring data confidentiality and unlinkability (ISO/IEC 24745 compliance).
- No plaintext fingerprint data is stored or transmitted, making the system resistant to hill-climbing and inverse biometric attacks.
- Decryption occurs on the authentication server, ensuring secure verification without exposing raw fingerprint data.

D. EFFICIENCY AND PRACTICALITY

- Despite encryption overhead, the system remains feasible for real-time applications:

  - Encrypted fingerprint templates for 90 users (one finger per user) require 31 MB of storage, averaging approximately 344 KB per fingerprint.
  - A single encrypted comparison completes in **0.025 seconds**.
  - Total processing time per sample: **0.136 seconds**.
- The authentication server decrypts the similarity score and makes a decision, ensuring a privacy-preserving verification process.

E. COMPUTATIONAL OVERHEAD ANALYSIS

The results highlight a trade-off between security and efficiency:

- Encryption is 25.93× costlier than decryption in execution time.
- Encrypted response time is 157.26× slower than plaintext response time.
- Encrypted fingerprint matching is 90× slower than plaintext matching.
- Despite this computational overhead, accuracy remains uncompromised, ensuring a secure and reliable fingerprint verification system.

F. OPTIMIZATION STRATEGIES

To improve efficiency while preserving security, the following optimizations are proposed:

- **Hardware Acceleration:** Utilize GPU or TPU-based cryptographic operations to reduce encryption time.
- **Optimized Encryption Schemes:** Implement lightweight cryptographic algorithms to minimize computational overhead.
- **Parallel Processing & Computation Offloading:**
  - Parallelize computations to optimize cryptographic operations.
  - Leverage secure cloud-based FHE processing for high-performance fingerprint authentication.

X. CONCLUSION AND FUTURE SCOPE

A privacy-preserving fingerprint recognition system is proposed, utilizing encrypted squared Euclidean distance for secure feature vector comparison. The system was evaluated on the SocoFing dataset, which consists of 7,200 altered fingerprint images used for model training, testing, and validation, while real (unaltered) fingerprints are securely stored for verification.

The proposed method, utilizing Euclidean similarity, ensures reliable biometric verification with an accuracy of 99.06% and a minimal EER of 0.40% at an optimized threshold of 9. The system achieves a TAR of 99.19%, an FRR of 0.81%, a FAR of 0%, and a TRR of 100%. FHE guarantees that fingerprint data remains confidential while enabling accurate matching in the encrypted domain. The model was trained for up to 500 epochs with an early stopping callback, achieving a test loss of 0.169, confirming its effectiveness.

Future work will focus on optimizing encryption schemes to enhance computational efficiency and evaluating the system's scalability for large-scale real-world deployments. Also, extend the framework to support multimodal biometric authentication by integrating with other biometrics like finger vein, palm vein, face or iris.

REFERENCES

[1] U. Sumalatha, K. K. Prakasha, S. Prabhu, and V. C. Nayak, "A comprehensive review of unimodal and multimodal fingerprint biometric authentication systems: Fusion, attacks, and template protection," *IEEE Access*, vol. 12, pp. 64300–64334, 2024.
[2] R. A. Joshi and N. B. Sambre, "Personalized CNN architecture for advanced multi-modal biometric authentication," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Apr. 2024, pp. 890–894.
[3] S. Vatchala, C. Yogesh, Y. Govindarajan, M. K. Raja, V. P. A. Ganesan, A. V. Arul, and D. Ramesh, "Multi-modal biometric authentication: Leveraging shared layer architectures for enhanced security," *IEEE Access*, vol. 13, pp. 28029–28041, 2025.
[4] J. C. Bernal-Romero, J. M. Ramírez-Cortés, and J. De Jesús Rangel-Magdaleno, "Unbreakable biometrics: How physical unclonable functions are revolutionizing security," *IEEE Instrum. Meas. Mag.*, vol. 27, no. 2, pp. 71–78, Apr. 2024.
[5] W. Yang, S. Wang, J. Hu, X. Tao, and Y. Li, "Feature extraction and learning approaches for cancellable biometrics: A survey," *CAAI Trans. Intell. Technol.*, vol. 9, no. 1, pp. 4–25, Feb. 2024.
[6] K. J. Singh and S. Ayeswarya, "Social media privacy dilemmas addressing security challenges through continuous authentication," in *Analyzing Privacy and Security Difficulties in Social Media: New Challenges and Solutions*. Hershey, PA, USA: IGI Global, 2025, pp. 295–310.
[7] E. L. Leijten and A. R. Lodder, "On AI that knows how we feel, without knowing who we are: Eu law and the processing of soft biometric data by emotional AI," in *Emotional Data Applications and Regulation of Artificial Intelligence in Society*. Cham, Swizerland: Springer, 2025, pp. 93–112.
[8] M. C. Marakalala and M. M. Matlala, "Border management identification: The biometric technology to detect criminals and terrorists often travel using falsified identity documents," *OIDA Int. J. Sustain. Develop.*, vol. 17, no. 12, pp. 57–70, 2024.
[9] H. O. Shahreza, Y. Y. Shkel, and S. Marcel, "On measuring linkability of multiple protected biometric templates using maximal leakage," *IEEE Access*, vol. 12, pp. 106618–106630, 2024.
[10] V. H. Champaneria, M. A. Zaveri, and S. J. Patel, "A secure template protection technique for robust biometric systems," in *Proc. IEEE Students Conf. Eng. Syst. (SCES)*, Jun. 2024, pp. 1–6.
[11] F. Belhadj and A. Moussaoui, "Attack via missed record synchronization on transformation-based fingerprint template protection algorithms," *Multimedia Tools Appl.*, vol. 83, no. 9, pp. 27543–27563, Aug. 2023.
[12] P. Bauspieß, T. Silde, M. Poljuha, A. Tullot, A. Costache, C. Rathgeb, J. Kolberg, and C. Busch, "BRAKE: Biometric resilient authenticated key exchange," *IEEE Access*, vol. 12, pp. 46596–46615, 2024.
[13] C. Choquehuanca-Chuctaya and A. Arroyo-Paz, "The security of biometric data in devices with cancellable biometrics technology: A systematic review of the literature," in *Proc. 4th Int. Conf. Emerg. Smart Technol. Appl. (eSmarTA)*, Aug. 2024, pp. 1–9.
[14] J. Meng, J. Du, Z. Liang, C. Yang, and Y. Jiang, "Application and optimization of multi-factor authentication and biometric technology in power grid energy network access control," *RE&PQJ*, vol. 2024, pp. 118–131, Aug. 2024.
[15] S. M. Abdullahi, S. Sun, B. Wang, N. Wei, and H. Wang, "Biometric template attacks and recent protection mechanisms: A survey," *Inf. Fusion*, vol. 103, Mar. 2024, Art. no. 102144.
[16] H. O. Shahreza and S. Marcel, "Breaking template protection: Reconstruction of face images from protected facial templates," in *Proc. IEEE 18th Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2024, pp. 1–7.
[17] D. Osorio-Roig, "Privacy preserving workload reduction in biometric systems," Ph.D. dissertation, Frankfurt Univ. Appl. Sci., Hochschule Darmstadt, Darmstadt, Germany, 2024.
[18] A. Durbet, "Privacy-preserving biometric authentication systems, a cryptographic approach," Ph.D. dissertation, Acad. Field: Comput. Sci., Université Clermont Auvergne, Clermont-Ferrand, France, 2024.

[19] K. Krishna Prakasha and U. Sumalatha, "Privacy-preserving techniques in biometric systems: Approaches and challenges," *IEEE Access*, vol. 13, pp. 32584–32616, 2025.
[20] A. A. Al-Saggaf, "A post-quantum fuzzy commitment scheme for biometric template protection: An experimental study," *IEEE Access*, vol. 9, pp. 110952–110961, 2021.
[21] S.-J. Lee, J.-M. Lee, and I.-G. Lee, "Low latency and secure data encryption for multi-hop biometric authentication in distributed networks," *Internet Things*, vol. 30, Mar. 2025, Art. no. 101501.
[22] S. Singh, L. Igene, and S. Schuckers, "Securing biometric data: Fully homomorphic encryption in multimodal iris and face recognition," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2024, pp. 1–6.
[23] R. Arjona, C. Franco, R. Román, and I. Baturone, "Combining CRYSTALS-kyber homomorphic encryption with garbled circuits for biometric authentication," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2024, pp. 1–5.
[24] R. Thaqi, K. Vishi, and B. Rexha, "Enhancing document security in cloud using Microsoft SEAL encryption," in *Proc. 8th Int. Symp. Innov. Approaches Smart Technol. (ISAS)*, Dec. 2024, pp. 1–5.
[25] W. K. Syed, A. Mohammed, J. K. Reddy, and S. Dhanasekaran, "Biometric authentication systems in banking: A technical evaluation of security measures," in *Proc. IEEE 3rd World Conf. Appl. Intell. Comput. (AIC)*, Jul. 2024, pp. 1331–1336.
[26] Y. Lohlah and P. Boonyopakorn, "Application of homomorphic encryption for encrypting and decrypting patient data in Thailand's healthcare system," in *Proc. Res., Invention, Innov. Congr., Innov. Electricals Electron. (RI2C)*, Aug. 2024, pp. 231–237.
[27] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, and A. James, "Sokoto coventry fingerprint dataset," 2018, *arXiv:1807.10609*.
[28] A. M. Ibrahim and A. Özpinar, "Securing and processing biometric data with homomorphic encryption for cloud computing," in *Proc. Int. Conf. Artif. Intell. Appl. Math. Eng.*, Cham, Swizerland. Springer, Jan. 2023, pp. 663–671.
[29] M. Nocker, D. Drexel, M. Rader, A. Montuoro, and P. Schöttle, "HE-MAN–homomorphically encrypted MAchine learning with oNnx models," in *Proc. 8th Int. Conf. Mach. Learn. Technol.*, Mar. 2023, pp. 35–45.
[30] J. Yuan, W. Liu, J. Shi, and Q. Li, "Approximate homomorphic encryption based privacy-preserving machine learning: A survey," *Artif. Intell. Rev.*, vol. 58, no. 3, p. 82, Jan. 2025.
[31] R. Chowdhury, A. Kumar, V. D. Mohite, and A. Chatterjee, "Transferability of evasion attacks against fhe encrypted inference: Official work-in-progress paper," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.*, Cham, Swizerland. Springer, 2024, pp. 40–68.
[32] H. Choi, J. Kim, S. Kim, S. Park, J. Park, W. Choi, and H. Kim, "UniHENN: Designing faster and more versatile homomorphic encryption-based CNNs without im2col," *IEEE Access*, vol. 12, pp. 109323–109341, 2024.
[33] J. Xiong, J. Chen, J. Lin, D. Jiao, and H. Liu, "Enhancing privacy-preserving machine learning with self-learnable activation functions in fully homomorphic encryption," *J. Inf. Secur. Appl.*, vol. 86, Nov. 2024, Art. no. 103887.
[34] Y. Zhang, H. Yi, and W. Kong, "A privacy-preserving brainprint recognition system based on feature homomorphic encryption," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2024, pp. 1–10.
[35] M. J. Khan, B. Fang, G. Cimino, S. Cirillo, L. Yang, and D. Zhao, "Privacy-preserving artificial intelligence on edge devices: A homomorphic encryption approach," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, vol. 126, Jul. 2024, pp. 395–405.
[36] J. Wang, R. Xin, O. Alfarraj, A. Tolba, and Q. Tang, "Privacy preserving security using multi-key homomorphic encryption for face recognition," *Expert Syst.*, vol. 42, no. 2, Feb. 2025, Art. no. e13645.
[37] S. Nakanishi, Y. Narusue, and H. Morikawa, "Accelerating homomorphic encryption-based facial recognition systems through clustering," in *Proc. IEEE 13th Global Conf. Consum. Electron. (GCCE)*, Oct. 2024, pp. 597–598.
[38] S. Vadim, M. Firdaus, and K.-H. Rhee, "Privacy-preserving decentralized biometric identity verification in car-sharing system," *J. Multimedia Inf. Syst.*, vol. 11, no. 1, pp. 17–34, Mar. 2024.
[39] S. C. Agrawal and R. K. Tripathi, "Biometric attendance system using face recognition," in *Proc. OPJU Int. Technol. Conf. (OTCON) Smart Comput. Innov. Advancement Ind. 4.0*, Jun. 2024, pp. 1–5.
[40] W.-F. Ou, L.-M. Po, and X.-F. Huang, "Joint learning of identity and vein features for enhanced representations in vascular biometrics," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 4440–4444.
[41] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 54–65, Sep. 2015.
[42] M. Ghilom and S. Latifi, "The role of machine learning in advanced biometric systems," *Electronics*, vol. 13, no. 13, p. 2667, Jul. 2024.

U. SUMALATHA received the B.E. and M.Tech. degrees from Visvesvaraya Technological University, Belagavi. She is currently pursuing the Ph.D. degree with the Department of Information and Communication Technology, Manipal Institute of Technology, Karnataka. Her current research interests include deep learning, biometrics, multimodal biometric systems, biometric template security, privacy-enhancing technologies, cryptographic protocols, and their applications.

K. KRISHNA PRAKASHA (Senior Member, IEEE) received the B.E. and M.Tech. degrees from Visvesvaraya Technological University, Belagavi, and the Ph.D. degree in network security from Manipal Academy of Higher Education (MAHE) University, Manipal, India. He is currently an Associate Professor with the Department of Information and Communication Technology, Manipal Institute of Technology, MAHE. He has more than 35 publications in national and international conferences and journals. His current research interests include information security, network security, algorithms, real-time systems, and wireless sensor networks.

SRIKANTH PRABHU (Senior Member, IEEE) received the M.Sc., M.Tech., and Ph.D. degrees from IIT Kharagpur. He is currently working as a Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal. He has more than 150 publications in national and international conferences and journals. His current research interests include pattern recognition, pattern classification, fuzzy logic, image processing, and parallel processing.

VINOD C. NAYAK received the M.B.B.S. and M.D. degrees from the Kasturba Medical College, Manipal Academy of Higher Education (MAHE), Manipal, India. He is currently a Professor with the Department of Forensic Medicine, Kasturba Medical College, MAHE. His current research interests include public health, epidemiology, traffic medicine, suicidology, toxicology, medical ethics and laws about medicine, medical education, endocrinology, and forensic pathology.