Homomorphically Encrypted Biometric Template Fusion and Matching

Ramin Akbari*, Luke Sperling*, Nalini Ratha *Fellow, IEEE,* Arun Ross *Senior Member, IEEE,* and Vishnu Naresh Boddeti *Member, IEEE*

***Abstract*—Biometric fusion is a promising method to elevate the recognition performance of unimodal biometric systems. Nevertheless, the exposure of feature vectors for feature-level fusion raises security concerns, as it is feasible to extract sensitive information from these vectors. This paper proposes a non-interactive, end-to-end approach to securely fuse and match biometric templates using Fully Homomorphic Encryption (FHE). For a pair of encrypted feature vectors, we perform the following operations on a ciphertext domain: i) feature concatenation, ii) fusion and dimensionality reduction through a learned linear projection, iii) an optional scale normalization to unit ℓ_2-norm, and iv) match score computation. Our method, dubbed HEFT, is custom-designed to circumvent a key limitation of FHE - the lack of support for non-arithmetic operations. From an inference perspective, we systematically explore different data packing schemes for computationally efficient linear projection and introduce a polynomial approximation for scale normalization. From a training perspective, we introduce two distinct FHE-aware algorithms to improve the learning of the projection matrix and address the challenges posed by the non-arithmetic normalization step. We demonstrate the utility of HEFT on two multimodal combinations: face and voice and face and fingerprint. For the face-voice fusion, HEFT improves verification performance by a range of 143.25% - 244.35% compared to unibiometric features. On the fingerprint-face fusion, improvements are from 13.99% to 37.99%. Code and data are available at https://github.com/human-analysis/encrypted-biometric-fusion**

***Index Terms*—Fully Homomorphic Encryption, Biometric Template Fusion, Secure Template Matching, Approximate Normalization**

I. Introduction

Feature-level fusion is commonly employed in multi-biometric recognition systems, especially in large-scale deployments as a dimensionality reduction technique. Template fusion helps overcome the limitations of uni-biometric systems in improving recognition performance and population coverage. However, utilizing multiple biometric features is associated with security risks and attacks on such systems. There is growing evidence that the templates contain sufficient information to reconstruct the raw biometric sample [1] or leak sensitive soft-biometric information [2]. Thus, in the context of fusion, it is imperative to design schemes that secure the biometric features of users across all modalities and protect user's privacy. Realizing this goal is the primary focus of this paper.

Ramin Akbari and Luke Sperling contributed equally. Ramin Akbari, Luke Sperling, Arun Ross, and Vishnu Naresh Boddeti are at Michigan State University, and Nalini Ratha is at the University at Buffalo.
Correspondence: vishnu@msu.edu

Cryptosystems based on Fully Homomorphic Encryption [3] (FHE) offer a promising solution to secure biometric templates through encryption. FHE schemes such as BFV [4], [5] and CKKS [6], allow arithmetic calculation on encrypted data without the need for decryption. Recent work [7], [8] has demonstrated that FHE is exceptionally effective and scalable for protecting biometric templates, allowing for matching and search in an encrypted domain against a gallery of 100 million.

Template-level fusion and matching typically involve the following operations: feature concatenation across various modalities, linear/non-linear projection of concatenation features to a lower dimension and a scale normalization of the resulting feature for computing cosine similarity in the matching score computation step. In existing approaches for feature-level fusion, operations are performed on plaintexts (unencrypted domain). However, performing those operations on ciphertexts (encrypted domain) in a given FHE scheme, is not straightforward due to their non-arithmetic nature. For example, division and square root required for scale normalization are not supported by FHE schemes for direct computation on ciphertexts. Furthermore, operations on ciphertexts are significantly expensive in terms of latency and memory requirements than the same operations on plaintexts.

To overcome the above challenges, we propose HEFT, a biometric template fusion and matching scheme that operates directly on encrypted templates. Given a pair of encrypted templates, HEFT performs the following operations over ciphertexts: feature concatenation, linear projection, optional scale normalization to unit ℓ_2-ball, and matching score computation.

The salient features of HEFT include i) fusion of uni-biometric templates with different dimensions , ii) dimensionality reduction through linear projection to ease the steep computational burden of downstream ciphertext matching operations, and iii) an arithmetic approximation of the ℓ_2 normalization operation through composite polynomials. Furthermore, we introduce FHE-aware learning for the projection matrix, either to bypass the normalization step or to reduce the cost of the arithmetic approximation, while maintaining matching performance.

For practical purposes, we analyzed how different design choices impact the trade-off between accuracy and efficiency (in terms of memory and latency) for biometric fusion and match score computation. These choices include data encoding schemes, matrix multiplication methods, and normalization approximations. Our analysis identifies the best options for both small-scale and large-scale settings, considering feature dimensions and gallery size.

In summary, we present a practical multi-biometric feature-level fusion and matching algorithm in the encrypted domain, utilizing FHE. Our key contributions are:

- Fusion in the encrypted domain using FHE.
- Introducing two frameworks for dimensionality reduction to address the limitations of FHE and elaboration on the theoretical motivation behind them
- Comparison of different matrix encoding schemes for efficient computation in the encrypted domain.
- Experimental evaluation on two multi-modal combinations, i) face and voice, and ii) face and fingerprint biometric templates.

Accordingly, we observe appreciable gains in matching performance over the uni-biometric counterparts. Compared to uni-biometric systems, HEFT improves matching performance by 143.25% and 224.40% for projection onto 32-dim space and 158.21% and 244.35% for projection onto 64-dim space. For the fusion of fingerprint and face dataset, improvements are 24.35% and 13.99% for projection onto 32-dim and 37.99% and 26.49% for projection onto 64-dim space. Furthermore, for a 32-dim projection it took about 1130 ms to fuse a pair of 512-dim biometric templates and 5006 ms to compute match scores against a gallery of size 1028 templates.

II. Related Work

Privacy-Preservation in Biometrics: Many methods have been devised to secure biometric templates and preserve user privacy. Early biometric cryptosystems based on image processing [9], [10] and fuzzy vaults [11] were employed for protecting both iris [12] and fingerprint [13] data. Such systems, however, suffered from a loss in matching performance. Cryptosystems such as Goldwasser-Micali encryption have also been used for authentication scenarios [14], but they do not protect the templates while matching and are, therefore, vulnerable to attacks.

Cancelable biometric schemes are a class of biometric template protection mechanisms that apply a non-invertible transformation $T_k(\cdot)$ to a biometric feature vector, where k denotes a user/system-specific key. These transformations are designed to be computationally efficient in the forward direction and infeasible to invert without knowledge of k, while still enabling comparison in the transformed space [15], [16]. A fundamental challenge in the design of cancelable biometric schemes lies in trading-off security (non-invertibility) with recognition accuracy. While prior work showed that careful parameter tuning can sometimes yield recognition performance comparable to, or even exceeding, that of unprotected systems [17], [18], a more fundamental trade-off exists within the design of the transformation itself. For example, in PolyProtect [17], increasing the *overlap* parameter enhances discriminability but simultaneously increases vulnerability to inversion under a full-disclosure threat model. Conversely, schemes such as IoM-URP [19], which prioritize stronger non-invertibility, often suffer from reduced recognition performance relative to more accurate but less secure methods like MLP-Hash [18]. These limitations are compounded by cryptanalytic vulnerabilities. Some transformations have been shown to leak soft biometric attributes even without access to the transformation key. Yalavarthi et al. [20] demonstrate that PolyProtect can leak soft biometric information; while their analysis is framed under a stated full-disclosure threat model, the specific attack they demonstrate, namely, training a classifier on protected templates, succeeds without leveraging knowledge of the user-specific keys. Similarly, Zhou et al. [21] reveal linkability vulnerabilities in Bloom filter-based encodings. Crucially, unlike cryptographic primitives like Homomorphic Encryption, cancelable biometric schemes lack formal reductions to hard problems and do not provide provable bounds on adversarial success rates. There are no theoretical guarantees on the preimage size or irreversibility of T_k, even under full- or partial-disclosure models. As a result, these schemes remain largely empirically evaluated, with no formal assurances against information leakage from the protected representation.

Homomorphic encryption (HE) is an attractive option for privacy-preserving biometrics applications because it enables computations on encrypted data without the need to decrypt, while affording strong and provable security guarantees. Early biometric systems driven by HE were based on partially homomorphic encryption (PHE) schemes [22]. They were applied to numerous biometric modalities [23], including face recognition [24], iris recognition [25]–[27] and fingerprint recognition [28]. The opportunity to design robust biometrics cryptosystems came to the fore with the development of the first fully homomorphic encryption (FHE) scheme [3]. Since then, there have been many application scenarios for biometrics exploiting the privacy afforded by FHE without substantial performance drawbacks. Gomez-Barrero et al. [29] developed a general framework for template-level fusion based on homomorphic encryption. This framework relies on performing fusion before encryption and does not support template fusion directly in the encrypted domain. Boddeti [7] demonstrated the ability to match face templates in the encrypted domain. Engelsma et al. [8] proposed an efficient way to search encrypted templates by combining a novel encoding scheme with feature compression. Using a tree search structure created by fusing similar templates, Drozdowski et al. [30] developed a faster biometric indexing and retrieval method.

In contrast to this body of work, in this paper, we leverage FHE for end-to-end template fusion and match score computation and devise an FHE-aware learning algorithm for feature projection. Unlike feature transformation methods, which face an inherent trade-off between security and matching accuracy, FHE provides strong security through encryption without impacting this accuracy. Furthermore, the primary advantage of FHE lies in its foundation on formal security proofs tied to computationally hard mathematical problems, which offers a provable guarantee of security. It is important to note that this security guarantee applies to the normal operational scenario; under a full-disclosure attack where the decryption key is known, FHE is by definition fully reversible. The primary challenge for FHE-based approaches is therefore not a trade-off with security, but rather one of performance(accuracy)-versus-complexity, due to the high computational cost of operating in the encrypted domain. In this work, our focus is to

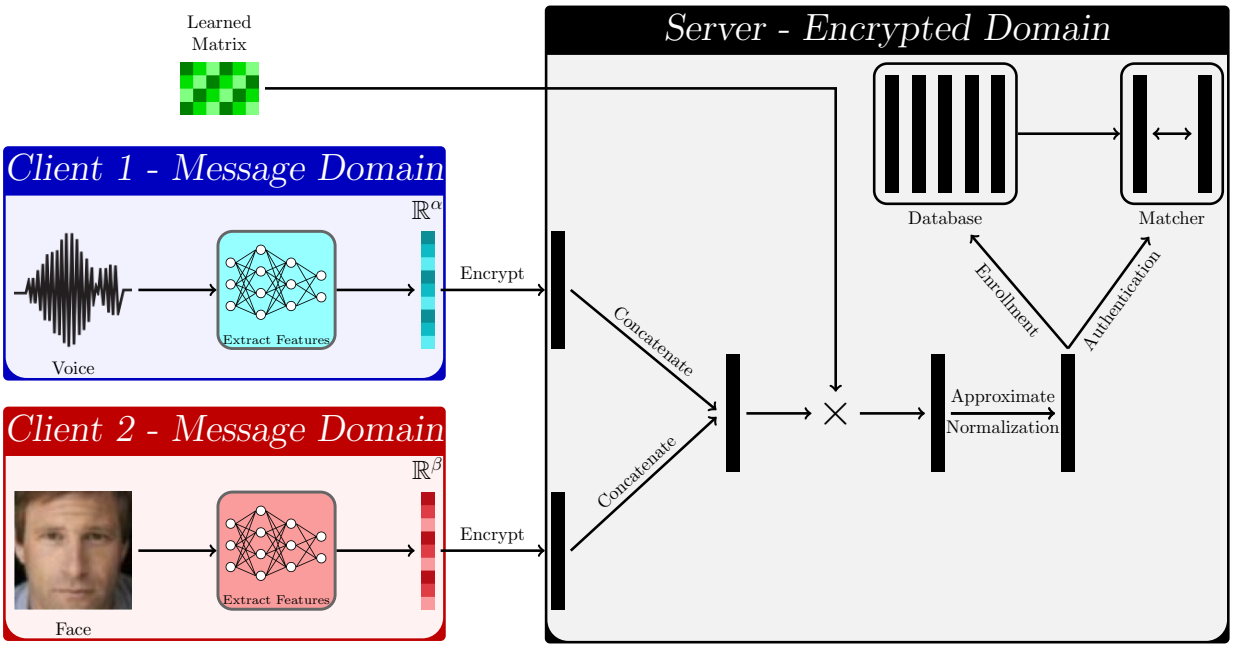


Fig. 1: End-to-end biometric template fusion and matching using fully homomorphic encryption (FHE). Given feature representations extracted from two different modalities of an individual, the clients encrypts and transmits the features to a cloud system. It will concatenate the two encrypted vectors and performs a matrix-vector multiplication with a learned plaintext projection matrix. The resulting ciphertext represents the fused encrypted vector. We optionally normalize the encrypted vector using an approximation to overcome the constraints imposed by FHE. During enrollment, this template is stored in the database of encrypted templates. During authentication, match scores are computed between the probe and templates from the encrypted database and sent to the clients for decryption and further processing.

reduce the computational complexity of FHE-based techniques while preserving accuracy.

Feature-Level Biometric Fusion: Fusion at the feature level leverages information from multiple templates to improve performance. Early techniques focused on selecting features from each template to be fused [31]. Sarangi et al. [32] combined face and ear templates by concatenating templates compressed through classical dimensionality reduction techniques. Feature-level fusion has also been performed on the face, fingerprint, and finger vein modalities [33]. Coupled mapping techniques have been devised to match samples between domains, with a maximum-margin approach [34] and a marginal fisher analysis approach [35]. Lately, learning-based approaches have been used. Silva et al. [36] performed feature selection using Particle Swarm Optimization. Tiong et al. [37] proposed a method of information fusion via extracting features from raw biometric data using a CNN and then combining them with a series of fully connected layers. Other deep learning approaches have been proposed recently [38]–[42]. Contrasting these methods, we opt for a linear projection-based approach to limit the multiplicative depth of the circuit and decrease computational complexity, which is important for creating a practical solution in FHE.

III. Approach

We propose HEFT for template fusion and matching. It is designed to maximize performance and efficiency for the biometric fusion performed on the encrypted domain. Given an encrypted multi-biometric dataset, i.e., a pair of encrypted feature vector matrices, HEFT performs the following series operations over the encrypted domain:

- *Concatenation* of the feature vectors
- *Linear projection* using a learned matrix to a new lower-dimensional feature space
- An optional *polynomial-based approximate normalization* of the features to projecting them onto a unit ℓ_2-ball
- *Match score computation* of the fused features against an encrypted gallery of fused features

These operations are illustrated in Figure 1. It is important to highlight that clients or vendors supplying the feature extractor may consist of different individuals. For example, a joint bank account could require biometric data from multiple individuals. In another scenario, a single client may want to utilize different biometric modalities where each provided by a separate vendor. Consequently, in the general case, we refer to them as distinct clients.

In addition to the aforementioned operations, HEFT is trained through an approximation-aware algorithm. This algorithm adjusts the projection matrix to address the non-arithmetic aspect of the feature normalization step. It follows one of two methods: either approximating the normalization step with a low-order polynomial or skipping this step.

A. Proposed Framework

Biometric Fusion: Consider a multi-biometric system that consists of a set of n feature vectors from two sources, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{\alpha}$ and $\boldsymbol{y}_i \in \mathbb{R}^{\beta}$. Each feature pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is fused into a new representation $\boldsymbol{z}_i \in \mathbb{R}^{\gamma}$. While we describe the fusion of two biometric feature sources, our approach extends naturally to

multiple sources. In this work, we consider a linear projection operation for feature fusion:

$$\boldsymbol{Z} = \boldsymbol{P}\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_n \end{bmatrix} = \boldsymbol{P}\boldsymbol{X}_{\oplus}$$

where $\boldsymbol{P} \in \mathbb{R}^{\gamma \times \delta}$ is the projection matrix that maps the concatenated feature space to a γ-dimensional space, and $\delta = \alpha + \beta$ represents the dimensionality of the concatenated features. The matrix $\boldsymbol{X}_{\oplus} \in \mathbb{R}^{\delta \times n}$ contains the concatenated feature vectors. The fused templates can be utilized for various downstream tasks, such as biometric matching. A commonly used metric for template matching is the cosine distance, defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = 1 - \langle \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}} \rangle,$$

where $\hat{\square}$ denotes a unit-normalized vector.

Secure Biometric Fusion: Our goal is to devise a cryptographic solution to secure the multi-biometric templates and prevent unauthorized access to private user information during the template fusion process and any desired downstream tasks. This can be achieved through a parameterized function that transforms the multi-biometric features $(\boldsymbol{x}, \boldsymbol{y})$ into an alternate space $(\mathcal{E}(\boldsymbol{x}), \mathcal{E}(\boldsymbol{y}))$ such that $\mathcal{E}(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}_{pk})$, $\boldsymbol{x} = g(\mathcal{E}(\boldsymbol{x}); \boldsymbol{\theta}_{sk})$ are encryption and decryption functions with $\boldsymbol{\theta}_{pk}$ and $\boldsymbol{\theta}_{sk}$ being the public and secret keys respectively. By executing all the fusion operations, namely, *concatenation*, *projection*, *normalization*, and *match score computation* directly over the ciphertexts, i.e., without decryption, we can prevent unauthorized access to sensitive information, and hence preserve user privacy. In this wise, FHE is a class of encryption algorithms that allows arithmetic computations directly over ciphertexts and is ideally suited to accomplish our goal.

B. Threat Model

As shown in Figure 1 HEFT involves three parties: two clients and an untrusted cloud server. Each client possesses a modality-specific biometric feature vector of a given user (e.g., face and fingerprint), while the cloud server performs the template fusion and match score computation. The clients hold the user's public and private keys[1] and are trusted to behave honestly.

Adversarial Model: We adopt a *semi-honest* (honest-but-curious) model for the cloud server. The server follows the protocol faithfully but attempts to infer sensitive information from the biometric templates it stores and processes. The server has access to (i) all ciphertexts received from clients, including modality-specific features and fused vectors, (ii) the plaintext projection matrix used in fusion (assumed public), (iii) the encrypted database of fused biometric templates, and (iii) the encrypted match scores produced during authentication. The adversary does not have access to the private FHE decryption key (securely held by the clients or a separate trusted party) or the internal biometric processing pipelines on the client side.

[1]Clients can exchange the common keys. For instance, this can be achieved by a Diffie–Hellman key exchange.

Adversarial Goals and Security: The adversary (cloud server) may attempt multiple attacks including (i) recovering raw biometric feature vectors from encrypted data, (ii) inferring relationships between modalities or deducing fused representations, and (iii) extracting identity or similarity information from encrypted match scores. However, since the adversary only has access to encrypted user data without access to the decryption keys (i.e., partial-disclosure threat model), HEFT protects against the attacks listed above.

C. Protocols: Template Fusion and Matching

We use the Cheon-Kim-Kim-Song (CKKS) scheme [6] as the underlying FHE scheme for template fusion and match score computation. We first give an overview of this scheme and then describe the enrollment and authentication protocols for template fusion.

The **CKKS encryption scheme** allows operations over encrypted vectors of complex numbers [6]. Its mathematical basis lies in modular arithmetic over polynomial rings and its security lies in the hardness of the Ring Learning with Errors problem. CKKS offers post-quantum security for an appropriate choice of encryption parameters [43]. Plaintexts are polynomials within the polynomial ring $R = \mathbb{Z}[x]/(x^N + 1)$. Therefore, complex vectors $C^{N/2}$ must be encoded into this space to perform encryption. After encoding, the plaintext polynomial is encrypted via a secret key into a set of two polynomials, $R_q^2 = \mathbb{Z}_q[x]/(x^N + 1)$ where R_q denotes polynomials of coefficients modulo q and degree less than N. This will serve as the ciphertext.

CKKS has three keys: a secret key sk, a public key pk, and an evaluation key evk for homomorphic multiplication. Its protocol comprises the following functions: i) *Key Generation:* Generates the keys, ii) *Encryption:* Given a plaintext polynomial and the public key pk, output two polynomials representing the ciphertext, iii) *Decryption:* Given a ciphertext comprised of two polynomials, apply the secret key sk and retrieve a plaintext polynomial, iv) *Addition:* A simple sum of the ciphertexts translates to homomorphic addition, v) *Multiplication:* Multiplication of ciphertexts is polynomial multiplication which results in three polynomials. To restrict the size of resultant ciphertexts, relinearization is needed, vi) *Relinearization:* Given three polynomials representing a ciphertext product, the evaluation key evk is used to reduce the size of the ciphertext from three to two polynomials, and vii) *Rotation:* Ciphertexts may be cyclically rotated using an optionally generated set of Galois keys.

Encrypted Template Fusion Protocol at Enrollment: Consider two sets of biometric templates $\mathcal{X}$ and $\mathcal{Y}$ that we seek to fuse along with their identity labels $\mathcal{I}$. Each set of templates is encrypted using the data encoding scheme requested by the cloud server. After receiving the encrypted templates, the cloud server performs the following operations: i) for each identity label c, create all pairs of templates $\mathcal{Z} = \{(\boldsymbol{x}_i, \boldsymbol{y}_j) | \forall (i,j) \in \mathcal{I}_c \times \mathcal{I}_c, \mathcal{I}_c \subseteq \mathcal{I}\}$, where $\mathcal{I}_c$ are the indices of samples belonging to identity c, ii) fuse the pairs of templates created, i.e., *concatenation*, *projection* and

an optional *normalization*, and iii) add the fused templates to the current gallery $\boldsymbol{G}$.

Encrypted Template Fusion Protocol at Authentication: clients send a sample of encrypted multibiometric templates $\boldsymbol{x} \in \mathbb{R}^{\alpha}$ and $\boldsymbol{y} \in \mathbb{R}^{\beta}$. This pair of templates is fused, i.e., *concatenation*, *projection* and optional *normalization* to create a probe template $\boldsymbol{z} \in \mathbb{R}^{\gamma}$. For identification, i.e., $1:N$ comparisons, the match score (e.g., cosine distance) is computed between the probe and the entire gallery $\boldsymbol{G}$. For verification with a claimed identity c, i.e., $1:1$ comparison, the match score is computed between the probe and the samples in the gallery $\boldsymbol{G}$ corresponding to the identity c. The encrypted scores are returned to the clients for decryption and further processing.

D. Encrypted Template Fusion and Matching

We now describe the various components of template fusion and match score computation. This includes (i) choice of the data encoding scheme, (ii) concatenating two ciphertexts, (iii) efficient ciphertext matrix-plaintext matrix multiplication for linear projection, and (iv) efficient approximate normalization.

1) Input Encoding and Vector Packing: **Input Encoding:** Before any computation on encrypted data, an encoding scheme must be selected to enable encryption and arithmetic operations on the resulting ciphertext. The efficiency of ciphertext operations is critically dependent on the encoding scheme chosen to represent features. As such, we outline two different encoding schemes for the feature vectors, each better suited for operating at a small or large scale. With regard to a set of concatenated templates as $\boldsymbol{X}_{\oplus} \in \mathbb{R}^{\delta \times n}$, these two schemes are: *Column Encoding*: Encodes each feature vector as a plaintext before encryption, thereby resulting in n ciphertexts. *Row Encoding*: Encodes each dimension of the feature vector as a plaintext before encryption, resulting in δ ciphertexts. It is worth noting that these encoding schemes were called *Dense* and *SIMD* by authors of [8] and [7]. Here, we used a different naming to avoid confusion with projection matrix encoding schemes.

Vector Packing: FHE schemes such as CKKS support arithmetic operations directly on vectors by packing multiple numbers into different slots within a single polynomial. In most practical applications, the dimensionality of feature vectors is much less than the number of available polynomial slots. Multiple feature vectors can be batched into a single polynomial in such cases. The batching allows for SIMD (single instruction multiple data) operations and helps amortize runtime across multiple feature vectors.

Suppose we wish to encode n vectors into polynomials with m slots each. In the Column encoding scheme, $\lceil \frac{n}{\lfloor \frac{m}{\delta} \rfloor} \rceil$ many polynomials are needed if rotation operations are not needed. However, ciphertext template fusion requires rotation operations. So, we pack an extra copy of each vector to simulate rotation's "wrapping" effect. Therefore, $\lceil \frac{n}{\lfloor \frac{m}{2\delta} \rfloor} \rceil$ polynomials are needed. In the SIMD encoding scheme, a single dimension of the n vectors can be packed into a single polynomial. In this scheme, $\delta \lceil \frac{n}{m} \rceil$ polynomials are needed to represent n δ-dimensional vectors.

2) Concatenating Ciphertexts: The concatenation mechanism depends on our choice of data encoding scheme. *Column Encoding:* In this case, each vector in the multibiometric dataset $[\boldsymbol{x}, \boldsymbol{y}]^T$ is zero-padded before encryption to a dimensionality of δ. Now, concatenation can be done in the encrypted domain by right-rotating each ciphertext in $\boldsymbol{y}$ by α slots and adding to the corresponding ciphertext in $\boldsymbol{x}$. (see Figure 2) *Row Encoding:* As each query dimension is packed into a single ciphertext, there is no need to concatenate the features. Instead, storing the ciphertexts in a single ordered array is sufficient in this representation.

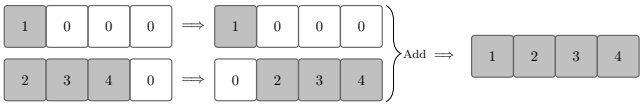


Fig. 2: Ciphertext concatenation via rotation and addition for the column encoding scheme. The second ciphertext (bottom) is right-rotated α slots and added to the first ciphertext (top).

3) Encrypted Linear Projection: Executing fusion through linear projection requires a matrix-matrix multiplication. Since we learn our projection matrix in the unencrypted domain, the multiplication is a plaintext-ciphertext multiplication, which is considerably more efficient than a ciphertext-ciphertext multiplication. Next, we outline two matrix-vector multiplication techniques, one that is better suited for small-scale datasets and the other for large-scale datasets. However, due to our ciphertext packing scheme, these methods functionally become matrix-matrix algorithms and can be treated as such. Furthermore, we note that the fused representations should be as compact as possible, i.e., γ should be small to ease the computational burden of any downstream tasks performed directly on the ciphertexts. Hence, the projection matrix $\boldsymbol{P} \in \mathcal{R}^{\gamma \times \delta}$ is rectangular.

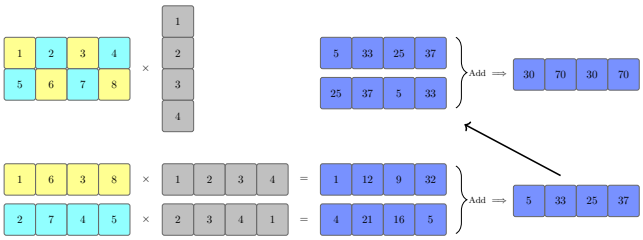


Fig. 3: **Hybrid:** The efficiency of matrix-vector multiplications can be improved through a diagonal encoding scheme for the projection matrix ($\boldsymbol{P}$). The query is rotated once and multiplied with each diagonally encoded component of $\boldsymbol{P}$. The sum of these results is rotated and added with itself to obtain the final output.

Hybrid: When the query vectors are encoded using the column scheme, the projection matrix can be encoded through a diagonal encoding scheme for efficient matrix-vector products. This scheme, shown in Figure 3, was introduced by Juvekar et al. [44] and is specialized for short and wide rectangular matrices, i.e., $\gamma < \delta$. These diagonals are multiplied by rotated versions of the query vector, and the resultant vectors can be additively combined to yield the desired matrix-vector multiplication result. This method is best suited for cases where n is small.

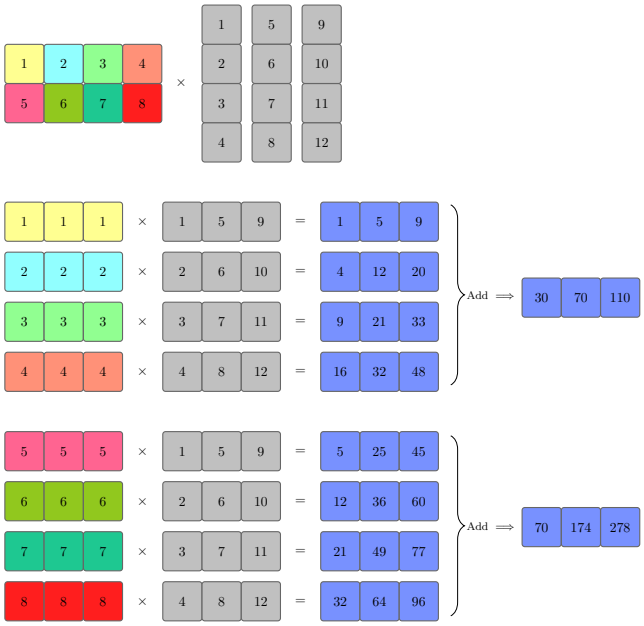
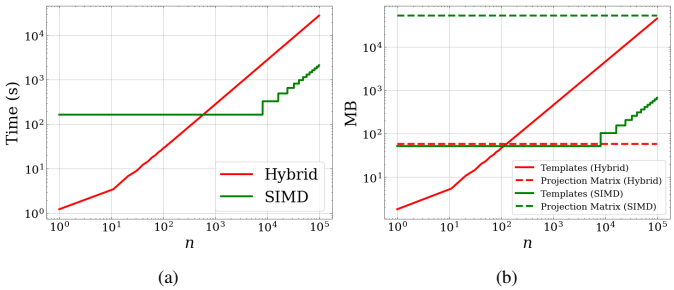


Fig. 4: **SIMD:** This method repeats and encodes each element of the projection matrix as plaintext and multiplies with row-encoded query vectors. The result is a single ciphertext for each dimension of the result. This method is best suited for large n.

SIMD: When the query vectors are encoded using the row scheme, the projection matrix can also be in a repeated row manner to support scalable matrix-vector products for large n. The scheme, shown in Figure 4, was adopted by Engelmsa [8] for scaling search over an encrypted database. This method takes $\gamma\delta$ plaintext-ciphertext multiplications for a single matrix-vector multiplication but admits greater ciphertext packing potential, making it a computationally more efficient solution when $n >> \gamma\delta$. This method also avoids any expensive ciphertext rotations. The SIMD scheme, however, is more memory intensive due to the need for loading many plaintexts and ciphertexts in memory, as seen in Figure 5b.

4) Practical Solution For Non-Arithmetic Operations: Recent biometric representations (e.g., DeepPrint [45], ArcFace [46]) are typically projected to the surface of a unit ℓ_2-ball[2]. Formally, $\hat{u} = u/\|u\|_2$ where $\|u\|_2^2 = \sum_{i=1}^{d} u_i^2$ for $u \in \mathbb{R}^d$. This normalization allows for computing cosine distance simply through a dot-product between vectors. Such a normalization operation, however, cannot be performed directly on the ciphertexts since FHE schemes do not support non-arithmetic operations such as square root and division in the encrypted domain. Although it is possible to approximate each of these operations individually [47], [48], the computational efficiency can be significantly improved by directly approximating the inverse square root operation. Panda [49] showed it is possible to approximate inverse square root through the iterative Goldschmidt's Algorithm [50], [51]. A similar but more general approach is to adopt a composite polynomial of the form $f(x) = (P_k \circ P_{k-1} \circ \cdots \circ P_1)(x)$, where each $g_i(x)$ is a low-degree polynomial, to approximate the inverse square root function in a desired interval of x i.e., $\frac{1}{\sqrt{x}} \approx f(x) \forall x \in [a, b]$[3].

[2]Other norms like ℓ_1 or ℓ_∞ can also be supported by HEFT if desired.
[3]See supplementary material for discussion on choosing the interval.



(a) (b)

Fig. 5: Comparison of theoretical runtimes (a) and memory requirements (b) for Hybrid and SIMD encoding schemes with $\delta = 1024$ and $\gamma = 32$.

The number of composite functions k and the degree of each g_i determine the homomorphic multiplicative depth of the operation. Higher-degree polynomials offer a better approximation of this function but also increase the multiplicative depth of the circuit. Hence, there is a trade-off between the accuracy of the approximation and computational efficiency. While this is operationally feasible, a composite polynomial that attains an adequate level of accuracy for approximating normalization demands a notable multiplicative depth due to the asymptotic nature of the inverse square root function, see Figure 6. This increases the computational complexity, which translates to more latency during fusion. In the following section, we present two alternative approaches. The first encompasses a rough estimate of the inverse square root function, while the second entirely circumvents this step. In both scenarios, we employ an approximate metric distance resembling cosine distance and adapt the loss function to reflect this modification to the projection matrix.

5) Approximation-Aware Projection: For a thorough analysis of the projection matrix $\boldsymbol{P}$ and its role, we consider a geometric view of the action of the projection operation. Let the Singular Value Decomposition (SVD) of $\boldsymbol{P}$ be $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the left and right singular basis and $\boldsymbol{\Sigma}$ is a diagonal matrix of singular values. When $\boldsymbol{P}$ is optimally learned using exact normalization during training, it transforms a unit hyper-sphere into a hyper-ellipsoid. Geometrically, the column vectors of $\boldsymbol{V}$ corresponding to the non-zero singular values, which act first on the concatenated vector, projects the vector onto a lower-dimensional space. Then $\boldsymbol{\Sigma}$ scales the i-th axis depending on the singular values σ_i. Finally, $\boldsymbol{U}$ performs a rotation of the resulting vector. For the purpose of our analysis, the action of $\boldsymbol{U}$ can be ignored since it preserves the cosine distance and does not affect the match score computation. Therefore, learning the optimal $\boldsymbol{P}$ reduces to i) aligning an orthonormal basis within the feature space (how data is read within the concatenated feature space by $\boldsymbol{V}$) and ii) finding an optimal anisotropic scaling (prioritizing the selected directions in the concatenated feature space) within the projected space to minimize the loss function and thus implicitly maximize matching accuracy. When $\boldsymbol{P}$ is learned with exact normalization, all directions in the transformed space are equally likely to maintain their norm after projection. Assuming the learning was robust, then the basis $\boldsymbol{V}$ has filtered out less important information (for the sake of verification performance) from the feature space. Without exact normalization, this filtering leads

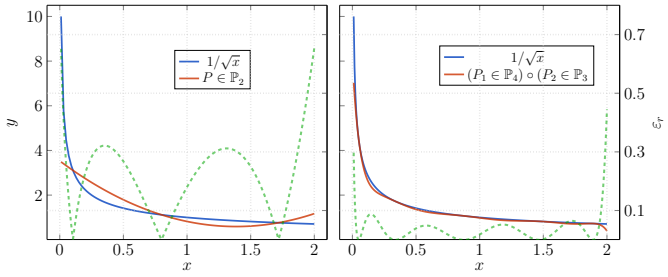


Fig. 6: Polynomial approximations of inverse square root over the interval [0.01, 2.0] for $P \in \mathbb{P}_2$ and $(P_1 \in \mathbb{P}_2) \circ (P_2 \in \mathbb{P}_3)$ polynomials. Relative error $\varepsilon_r = \left(\left|f(x) - \frac{1}{\sqrt{x}}\right| / \frac{1}{\sqrt{x}}\right)$ of the approximations are shown on right y-axis of each plot.

to non-uniform norms in different directions in the projected space. Additionally, anisotropic scaling by $\boldsymbol{\Sigma}$, learned through exact normalization, amplifies this skewness, which leads to performance loss. Therefore, in the absence of normalization, this skewness should be adjusted to favor the cosine distance, and the optimal $\boldsymbol{V}$ should also consider the skewness along with importance of information. Errors originating from the skewness, are less severe when using a polynomial approximation for normalization compared to a normalization free approach, and a more accurate polynomial approximation continues to reduce these errors. To effectively mitigate such errors, one strategy is to integrate awareness of these errors into the learning process, allowing the projection matrix $\boldsymbol{P}$ to adapt to approximate normalization (or no normalization) by optimizing skewness. This adaptation can be achieved by adjusting the columns of $\boldsymbol{V}$ corresponding to the non-zero singular values and the anisotropic scaling matrix $\boldsymbol{\Sigma}$. Both $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ influence the skewness of the projection. Therefore, training with an approximation-aware scheme, should change the direction of the right singular vectors and reassign their corresponding singular values. For example, the first right singular vector (corresponding to the largest singular values) of the projection matrix learned with approximation may lie in the span of the first few right singular vectors—such as the first three or five that correspond to largest singular values—from the projection learned with exact normalization. If the normalization approximation is less accurate, this span is expected to include more singular vectors.

6) Computational Complexity: Table I shows an analytical comparison of the time and space complexity of the end-to-end pipeline for both the column and SIMD encoding schemes. We show the required number of atomic operations for each pipeline stage, including concatenation, projection, normalization, preprocessing (necessary for matching), and matching.

E. FHE Aware Learning of Projection Matrix

We now focus on learning the optimal projection matrix $\boldsymbol{P}$ for template fusion. We posit that $\boldsymbol{P}$ can be learned in the unencrypted domain using biometric templates that are already available and thus do not suffer from privacy concerns. Once learned, it can be employed for fusing the encrypted templates and hence, user's data remains private.

The projection matrix should map vectors of the same class close together for a given distance metric, while those of different classes should be far apart. To realize this goal, we adapt the concept of the maximum-margin loss function introduced by Siena et al. [34] for learning $\boldsymbol{P}$. The loss function minimizes the distance between samples of the same class and uses a hinge loss on triplets of samples involving a similar and dissimilar pair. We build upon this concept and adapt it in several ways to satisfy the unique combination of constraints imposed by the multimodal fusion of features from deep neural networks and those of normalization approximations induced by FHE computations at inference.

Firstly, we adopt the loss function for multimodal feature-level fusion. Specifically, unlike Siena et al. [34] who seek to learn a pair of projection matrices with Euclidean distance-based metric, we learn a single projection matrix with cosine similarity[4] based metric. Given a concatenated dataset $\boldsymbol{X}$, the loss function is defined as:

$$\begin{aligned}
\mathcal{L}_{pull} &= 1 - \frac{1}{|Q|} \sum_{i,j \in Q} \langle \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j \rangle \\
\mathcal{L}_{push} &= \frac{1}{|S|} \sum_{i,j,k \in S} \max\left(\nu - \langle \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j \rangle + \langle \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_k \rangle, 0\right) \\
\mathcal{L} &= \lambda \mathcal{L}_{pull} + \mathcal{L}_{push} \qquad (1)
\end{aligned}$$

where $\langle \boldsymbol{.}, \boldsymbol{.} \rangle$ is the *inner product*, $\hat{\boldsymbol{\xi}}$ is a unit vector in the *projected space*, $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}/\|\boldsymbol{\xi}\|_2$ and $\boldsymbol{\xi}$ is the projection of $\boldsymbol{x}$, Q is the set of all pairs belonging to the same class, S denotes the set of all triplets $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ belong to the same class and $(\boldsymbol{x}_i, \boldsymbol{x}_k)$ belong to different classes, λ is a hyperparameter that weighs the "push" and "pull" terms' influences on the loss, ν is the margin hyperparameter that determines the desired margin of separation between samples belonging to the same class and those belonging to different classes. The margin hyperparameter used in the triplet hinge loss can appropriately take on any value in the range $\left[0, \frac{c}{c-1}\right]$ for c classes [52].

Secondly, we note that the loss function in (1) is defined with exact normalization, while at the inference stage HEFT can only perform approximate normalization as described in §Section III-D4. For instance, Figure 6 compares the exact inverse square root function and polynomial approximations of degrees 2 and 6. Following the discussion in §Section III-D5 we propose the following two training schemes:

Poly Similarity :

$$\langle \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j \rangle \approx \langle \boldsymbol{\xi}_i, \boldsymbol{\xi}_j \rangle \times f(\langle \boldsymbol{\xi}_i, \boldsymbol{\xi}_i \rangle) \times f(\langle \boldsymbol{\xi}_j, \boldsymbol{\xi}_j \rangle) \qquad (2)$$

Tanh Similarity :

$$\langle \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j \rangle \approx \tanh(\langle \boldsymbol{\xi}_i, \boldsymbol{\xi}_j \rangle) \qquad (3)$$

In equation (2), f represents the composite polynomial function approximating the inverse square root function. Our choice of $\tanh(\mathrm{x})$ in equation (3), maps the range of Euclidean inner product to cosine similarity and encourages adaptation in $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ to drive the inner product of similar classes to a larger value compared to the inner product of dissimilar classes. Therefore, there is no notion of normalization in this approach.

[4]Note that HEFT can also optimize for Euclidean distance if desired.

Encoding	Operation	Time Complexity					Space Complexity
		Additions	Plain-Cipher Mult.	Cipher-Cipher Mult.	Mult. Depth	Rotations	
Column	Concatenation	$\lceil \frac{n}{l} \rceil$	0	0	0	$\lceil \frac{n}{l} \rceil$	$O(p\lceil \frac{n}{l} \rceil)$
	Projection	$(\gamma + log(\delta) - log(\gamma) - 2)\lceil \frac{n}{l} \rceil$	$\gamma\lceil \frac{n}{l} \rceil$	0	1	$(\gamma + log(\delta) - log(\gamma) - 1)\lceil \frac{n}{l} \rceil$	$O(\gamma p + p\lceil \frac{n}{l} \rceil)$
	Normalization	$log(\gamma)\lceil \frac{n}{l} \rceil$	$d\lceil \frac{n}{l} \rceil$	$2\lceil \frac{n}{l} \rceil$	$2 + d$	$log(\gamma)\lceil \frac{n}{l} \rceil$	$O(p\lceil \frac{n}{l} \rceil)$
	Preprocessing	$\lceil \frac{n}{l} \rceil - \lceil \frac{n\gamma}{m} \rceil$	$\lceil \frac{n}{l} \rceil$	0	1	$\lceil \frac{n}{l} \rceil - \lceil \frac{n\gamma}{m} \rceil$	$O(p\lceil \frac{n}{l} \rceil + p\lceil \frac{n\gamma}{m} \rceil)$
	Matching	$log(\gamma)\lceil \frac{n\delta}{m} \rceil$	0	$\lceil \frac{n\gamma}{m} \rceil$	1	$log(\gamma)\lceil \frac{n\delta}{m} \rceil$	$O(p\lceil \frac{n\gamma}{m} \rceil)$
Row	Concatenation	-	-	-	-	-	-
	Projection	$\gamma(\delta - 1)\lceil \frac{n}{m} \rceil$	$\delta\gamma\lceil \frac{n}{m} \rceil$	0	1	0	$O(\delta\gamma p + \gamma p\lceil \frac{n}{m} \rceil)$
	Normalization	$(\gamma - 1)\lceil \frac{n}{m} \rceil$	$d\lceil \frac{n}{m} \rceil$	$2\gamma\lceil \frac{n}{m} \rceil$	$2 + d$	0	$O(\gamma p\lceil \frac{n}{m} \rceil)$
	Preprocessing	-	-	-	-	-	-
	Matching	$(log(\gamma) - 1)\lceil \frac{n}{m} \rceil$	0	$\gamma\lceil \frac{n}{m} \rceil$	1	0	$O(\gamma p\lceil \frac{n}{m} \rceil)$

TABLE I: Time and memory complexity comparison of the column and row encoding schemes for template fusion and matching. A preprocessing step is used in the column encoding scheme to reduce the number of ciphertexts in the gallery to enable faster matching. γ is the output dimensionality of the resultant vector. δ is the dimensionality of the query vector. For m slots available in a single ciphertext, we define $l = \lfloor \frac{m}{2\delta} \rfloor$. Depending on the encoding scheme, to process n samples, we must perform each operation $\lceil \frac{n}{l} \rceil$ or $\lceil \frac{n}{m} \rceil$ times ($\lceil \frac{n\gamma}{m} \rceil$ times to perform matching in the column scheme). p denotes the amount of space a single ciphertext occupies in memory.

For *Poly Similarity* case, compared to [53], our approach involves a two-phase training process. Initially, we pre-train a model with exact normalization, and next with a fine-tuning phase, we readjust the projection matrix. Also, prior to the fine-tuning phase of training, we scale $\boldsymbol{P}$ values to align the Euclidean norm's range with the domain of polynomials used for approximating the inverse square root function.

In the case of *Tanh Similarity*, the loss surface is more complex. The Euclidean inner product inherently depends on both cosine similarity and Euclidean distance, meaning both metrics can be adjusted during optimization to minimize the loss. Additionally, the distribution of the Euclidean norm after projection is crucial; ideally, it should avoid the saturated regions of the $\tanh(x)$ function. To address this optimization issue, we suggest to employ a similar approach as in *Poly Similarity*. We first pre-train the model using exact cosine similarity, enabling the projection matrix to find a fair initial basis $\boldsymbol{V}$. Next, we scale $\boldsymbol{P}$ so the average size of the projected concatenated features will have a unit length. Next, during the fine-tuning phase, we modify the loss function with (3) to adjust the projection matrix $\boldsymbol{P}$. It is worth mentioning, with *Tanh Similarity*, we expect more significant changes to the projection matrix compared to when a weaker normalization methods (such as low order polynomial) is being used. As a result, the pre-training phase for *Tanh Similarity* can be shorter, while having an extended fine-tuning phase compared to the *Poly Similarity* approach.

IV. Experiments

We evaluate the effectiveness of HEFT and analyze the effect of our design choices in terms of matching accuracy and computational complexity.

Implementation Details: To learn the projection matrix, we use the Adam [54] optimizer with an exponential learning rate decay and with a dropout on the concatenated vector of unimodal features.

For face⊕voice ($\oplus$ denotes direct sum of two subspaces) in the exact training, the learning rate is 10^{-3} with decay of 0.98, weight decay (Regularization) is 10^{-1}, dropout percentage is 0.35 and number of epochs are 60. For the pre-training phase of *Poly Similarity* approach, learning rate is 5×10^{-4} and number of epochs is 30 while all other hyperparameters are the same as the exact training. For the fine-tuning, learning rate is 10^{-4} and number of epochs is 30. In the *Tanh Similarity*, for the pre-training phase, we use a learning rate of 10^{-4} with decay of 0.99 and 5 epochs. For the fine-tuning part, learning rate is set to 10^{-3} with decay of 0.98, dropout is 0.2 and number of epochs are 100.

For face⊕finger, in the exact training, the learning rate is 3×10^{-4} with decay of 0.98, weight decay (Regularization) is 10^{-1}, dropout percentage is 0.1 and number of epochs are 40. For the pre-training phase of *Poly Similarity* approach, learning rate is 3×10^{-4} and number of epochs is 30 while all other hyperparameters are the same as the exact training. For the fine-tuning, learning rate is 10^{-4} and number of epochs is 25. In the *Tanh Similarity*, for the pre-training phase, we use a learning rate of 2×10^{-4} with decay of 0.99 and 2 epochs. For the fine-tuning part, learning rate is set to 7×10^{-4} with decay of 0.98, dropout is 0.1 and number of epochs are 70.

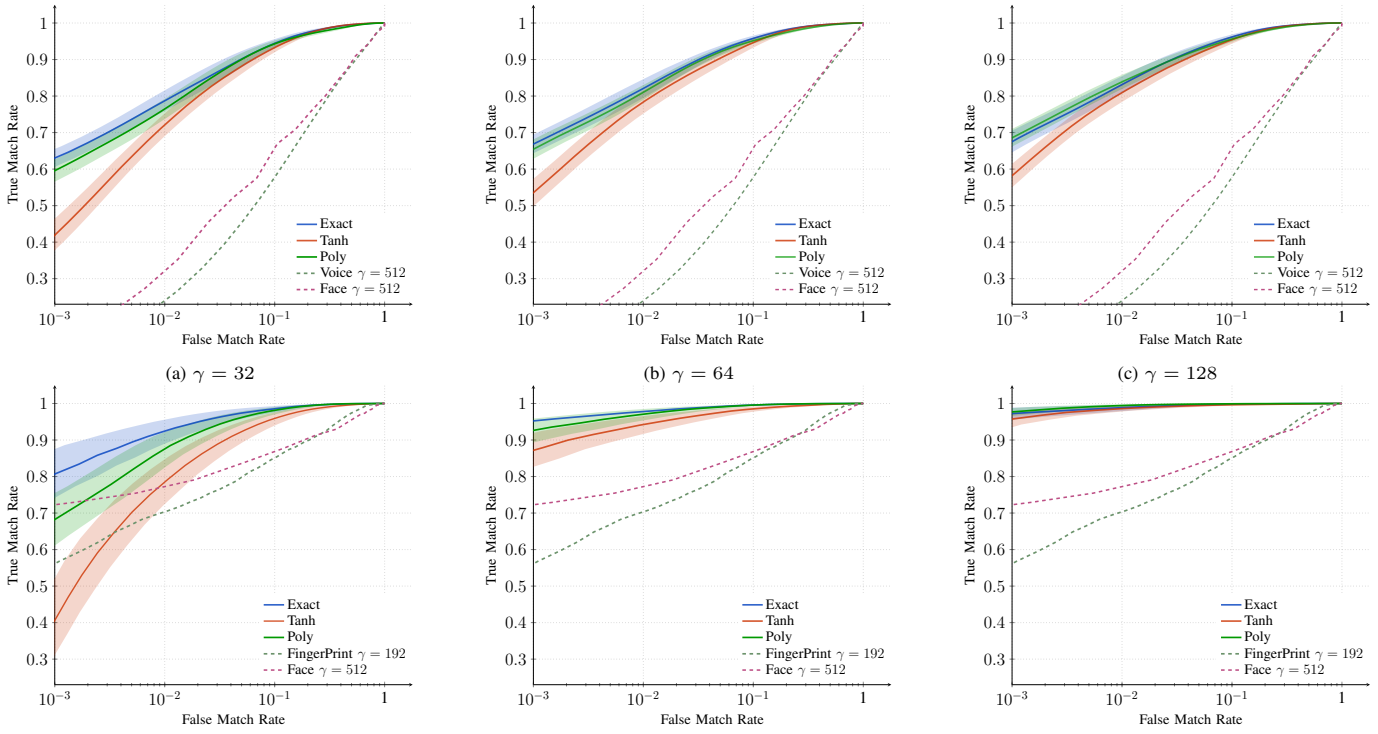
The encrypted projection is based on the CKKS scheme implemented in Microsoft's SEAL [55] library. Depending on the multiplicative depth of our approximate normalization method, we either use a polynomial modulus degree (N) of 16,384 or 32,768 along with a chain of very large prime numbers totaling 420, 580 or 860 bits as the coefficient modulus (q). This configuration satisfy 128-bit security [56].

A. Evaluation Datasets

We consider two datasets for evaluation HEFT: 1) a dataset with a synthetic combination of voice and face biometrics from different datasets, and 2) a real multi-biometric dataset of face and fingerprint.

1) Synthetic Voice and Face Multimodal Dataset: **Google Speech Commands:** This dataset comprises spoken single-word commands from many speakers. We use 5380 samples over 188 classes. We extract 512-dimensional feature vectors with the Deep Speaker [57] model, which is trained on the train-clean-360 portion of the LibriSpeech [58] dataset using a publicly available implementation.

CPLFW [59]: This benchmark face dataset is a harder version of LFW that incorporates cross-posed faces. We extract 512-dimensional feature vectors from a pre-trained VGG16 model trained on VGGFace [60], [61]. We pair two samples of 188 identities from CPLFW with those in the Google Speech Commands Dataset to create a multimodal dataset. This results in 10,760 samples over 188 classes as our dataset. Of these,



(a) $\gamma = 32$ (b) $\gamma = 64$ (c) $\gamma = 128$

(d) $\gamma = 32$ (e) $\gamma = 64$ (f) $\gamma = 128$

Fig. 7: ROC Curves for the Face⊕Voice (a-c) and Face⊕Finger Print (d-f) dataset. For statistical reliability, we repeat the experiment 8 times for Face⊕Voice and 10 times for Face⊕Finger with different train/test splits of the dataset and show the uncertainty. Fusion improves significantly matching performance over the weakest unibiometric modalities (dashed lines). For more clarity, we considered 0.3σ for the confidence intervals.

20% of the classes are used for testing, 20% for validation, and 60% for training. This yields a test set of 1028 samples.

2) Real Face and Fingerprint Multimodal Dataset: We evaluate on a real multimodal dataset [62] combining face, iris, fingerprint, palmprint, and voice samples over the same subjects. Among these, we consider face and fingerprints for fusion. For both face and fingerprints, we extract fixed-length representations from ArcFace [46] (512 dimensions) and global features from a fingerprint transformer [63] (192 dimensions), respectively. The dataset has 195 face images and 890 total fingerprints (after selecting fingerprints with id 2 and removing *Cross Match* sensor). Pairing the face and fingerprint samples and considering at least 5 samples per class, yields a dataset with 2472 multimodal samples across 61 subjects. Furthermore, to demonstrate the utility of biometric template fusion, we consider a more challenging scenario where the features are noisy. Accordingly, we add uniform noise from $\mathcal{U}(0, 0.1)$ to 85 percent of training and the test sets in face features, with the rest being the noiseless features. This will bring face modalities at around the same level of performance as the fingerprint modalities.

B. Comparison and Selection of Encoding Scheme

As discussed in §Section III-D1, there are two encoding schemes, each with different computational properties. To select the appropriate one for our purposes, we first numerically compare them. The time and space complexity for the end-to-end pipeline, i.e., concatenation, projection, approximate normalization, and match score computation, of each encoding scheme, are shown in Figure 5a and Figure 5b respectively. To compute the numerical values from the theoretical expressions in Table I, we compute the runtime of each atomic operation in SEAL by averaging over 1,000 operations with the appropriate encryption parameters. Similarly, space is calculated by examining the size of a single ciphertext. As expected, we observe a cross-over point between the two, with SIMD being more efficient in terms of latency for $n > 1000$ and in terms of memory for $n > 10000$. Furthermore, for our dataset of size 1028, while the latency between the two is comparable, the dense encoding scheme has lower memory requirements. Therefore, we use the dense encoding scheme for all further experiments.

C. Evaluation Metrics and Results

In HEFT to compute the score of feature vectors, first, we apply the appropriate transformation as discussed in §Section III-E. Next, we use the value of True Match Rate at 1% False Match Rate as the metric to evaluate the template fusion performance. The metric is computed in the unencrypted domain after decrypting the match scores. To evaluate the performance of HEFT, we compare it against the following baselines: i) the unibiometric templates, ii) a simple concatenation of the unibiometric features, i.e., $\boldsymbol{X}_{\oplus}$, iii) training using exact normalization, and iv) the feature averaging fusion technique introduced in [30]. Figure 7 and Table II compare the performance of HEFT with the baselines. Accordingly, we make the following observations : i) the concatenation and fusion of features (trained with exact normalization) across both datasets (rows 4-6 for Face⊕Voice and rows 14-15 for Face⊕Finger) improves the matching performance

Index	Data	Domain	Normalization		Dimensionality	True Match Rate @1% False Match Rate
			Inference	Learning		
1	CPLFW	Unencrypted	Exact	-	512	0.3142
2	GSC	Unencrypted	Exact	-	512	0.2356
3	Average [30]	Encrypted	Exact	-	512	0.7225
4	Face $\oplus$ Voice	Unencrypted	Exact	-	1024	0.7148
5		Unencrypted	Exact	Exact	32	0.7917
6		Unencrypted	Exact	Exact	64	0.8177
7		Encrypted	Poly $\in \mathbb{P}_2$	Poly $\in \mathbb{P}_2$	32	0.7643
8		Encrypted	Poly $\in \mathbb{P}_2$	Poly $\in \mathbb{P}_2$	64	0.8113
9		Encrypted	-	Tanh	32	0.7199
10		Encrypted	-	Tanh	64	0.7831
11	Face $\oplus$ Voice [53]	Encrypted	Poly $\in \mathbb{P}_2$	Poly $\in \mathbb{P}_2$	32	0.6012
12	Finger	Unencrypted	Exact	-	192	0.7031
13	$\widetilde{\text{Face}}$	Unencrypted	Exact	-	512	0.7670
14		Unencrypted	Exact	-	704	0.9424
15	$\widetilde{\text{Face}} \oplus$ Finger	Unencrypted	Exact	Exact	32	0.9241
16		Unencrypted	Exact	Exact	64	0.9780
17		Encrypted	Poly $\in \mathbb{P}_2$	Poly $\in \mathbb{P}_2$	32	0.8743
18		Encrypted	Poly $\in \mathbb{P}_2$	Poly $\in \mathbb{P}_2$	64	0.9702
19		Encrypted	-	Tanh	32	0.7834
20		Encrypted	-	Tanh	64	0.9421

TABLE II: Comparison of HEFT versus baselines for True Match Rate at 1% False Match Rate - $\widetilde{\square}$ indicates noisy version of original features

Protocol	Enc. Norm. Method	Concatenation	Projection	Normalization	Preprocessing	Fusion Total	Score Comp.
Enrollment	Poly (Deg=2)	5.68	244.89	31.40	3.41	285.38	-
	Poly (Deg=6)	11.17	470.86	83.32	3.62	568.97	-
	Goldschmidt's	23.22	954.03	380.28	2.31	1,359.84	-
Authentication	Poly (Deg=2)	22.72	979.54	125.59	-	1,127.85	4.87
	Poly (Deg=6)	89.05	3,752.24	663.95	-	4,505.24	5.21
	Goldschmidt's	185.00	7,602.64	3,030.47	-	10,818.11	2.75

TABLE III: Time (milliseconds) breakdown for each step in enrollment and authentication for a single sample for projection onto 32-dim. For comparison, the same operations in plaintext i.e., unencrypted features, take 0.62, 1.02, 11.75, and 4.51 μs, respectively, for concatenation, projection, normalization, and score computation per sample/match.

compared to uni-biometric features. ii) Feature fusion yields more robust and efficient matching performance compared to averaging features. iii) approximate-learning schemes also show superior performance over uni-biometric features in both datasets, as well as over feature averaging in the Face$\oplus$ Voice dataset. iv) the gap between approximation schemes and exact normalization narrows as the fusion dimension increases. This trend is also observed when comparing *Tanh Similarity* with *Poly Similarity*. v) compared to [53], approximate learning through fine tuning results in more robust training row(7 vs 11). Additionally, Figure 8 illustrates the impact of approximate learning compared to a naive approach where different normalization methods are used during training and score matching. The ROC curves show that approximation learning, whether using *Tanh Similarity* or *Poly Similarity*, improves performance. This indicates that a projection learned by approximation, even without using normalization in score matching (*Tanh Similarity*), outperforms a naive approach with approximate normalization (a second order polynomial without learning). Similarly, in the case of *Poly Similarity*, a weak approximation using a second-degree polynomial is nearly equivalent to the Goldschmidt method, which is more accurate for approximating normalization but comes with a significantly high computational costs.

Computational Complexity: The efficiency of homomorphic operations critically depends on the chosen encryption parameters. We select these parameters based on the multiplicative depth needed for end-to-end fusion and matching. Table III shows the latency of each component of HEFT. First, we observe a trade-off between performance and time complexity, with the 2^{nd}-degree polynomial being $2\times$ faster than the 6^{th}-degree polynomial for enrollment. Although Goldschmidt's algorithm performs the best, it is $4.8\times$ and $9.6\times$ slower than HEFT with degree two approximation for enrollment and authentication, respectively. For the normalization-free case, i.e $\tanh(x)$, the time complexity now corresponds to the time in the Fusion Total column minus the time in the Normalization column which leads to improvement of the latency during the score matching phase.

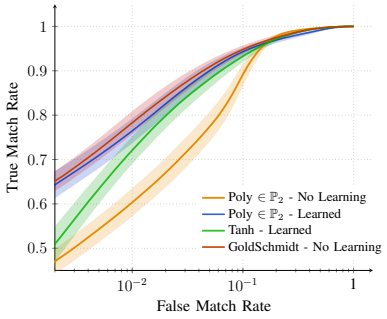
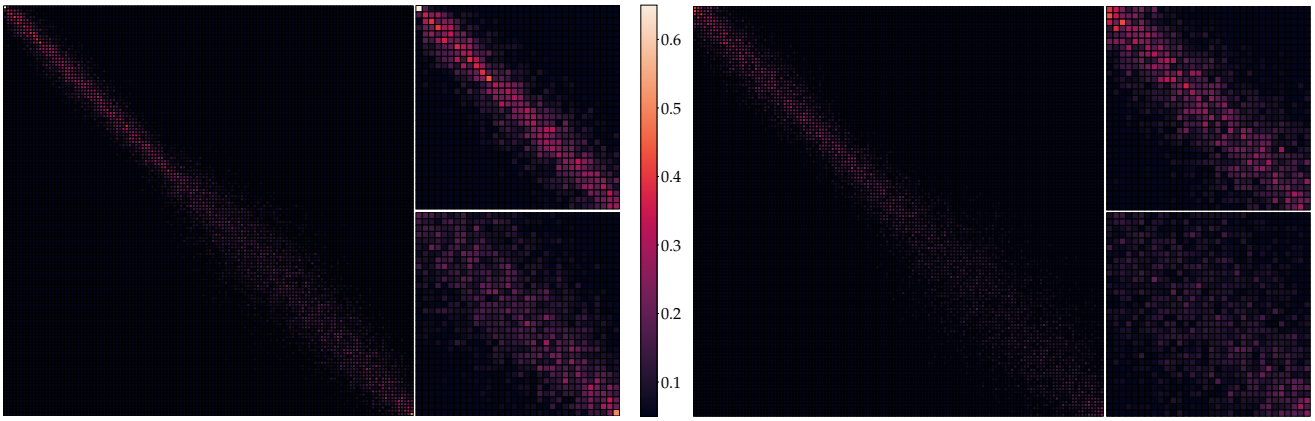


Fig. 8: Effectiveness of approximation-aware schemes compared to naive approach where different normalization is used for training and matching. Dataset is Face $\oplus$ Voice and fusion dimension is $\gamma = 32$. The confidence interval shown corresponds to 0.3σ for clarity.

V. Ablation Studies

In this section, we analyze the effect of our design choices on the accuracy of biometric matching.

Effects of Approximation-Learning: As discussed in §Section III-D5, it is necessary to re-adjust the right singular vectors $\boldsymbol{V}$ and their corresponding singular values to correct the skewness in the projection and prevent this skewness from affecting the score matching stage. We previously mentioned



(a) $\boldsymbol{V}_{Exact}^{T}\boldsymbol{V}_{Poly}$, $\gamma = 128$ (b) $\boldsymbol{V}_{Exact}^{T}\boldsymbol{V}_{Tanh}$, $\gamma = 128$

Fig. 9: Similarity matrix of the right singular vectors ($\boldsymbol{V}$ - ordered by singular values in descending manner) for projections learned through exact normalization and approximation-aware methods (*Poly/Tanh Similarity*). Each figure shows the full similarity matrix, the first 32×32 block (top right subplot) and the last 32×32 block (bottom right subplot). The dataset used is Face $\oplus$ Voice.

that incorporating approximate normalization reduces these errors compared to a normalization-free approach. First, this is confirmed by Figure 7 where *Poly Similarity* outperforms *Tanh Similarity*, particularly in lower-dimensional fusion spaces. It is known that the first few right singular vectors contain the most information and are critical for score matching. When projecting to much lower dimensions, a normalization-free approach must sacrifice some of this important information to mitigate skewness. As shown in Figure 7, when fusion dimension increases, *Tanh Similarity* demonstrates a more significant performance gains across both datasets compared to exact normalization and *Poly Similarity*. Second, to assess the extent of modification in selected directions within the concatenated space, we can analyze the similarity matrix between the right singular vectors of the projection matrix obtained through exact normalization and approximation-aware methods. Figure 9 presents these similarity matrices for both *Poly Similarity* and *Tanh Similarity*, where the right singular vectors are ordered by singular values (from larger to smaller ones). A banded similarity matrix indicates fewer modifications, as it suggests that a singular vector of a given order (based on its singular value) in the modified projection matrix lies within the span of right singular vectors in the original projection matrix that have the same order (singular values). From the figure, we observe that both matrices exhibit a banded structure, with the bandwidth being larger in *Tanh Similarity* compared to *Poly Similarity*. This suggests that *Tanh Similarity* involves greater modification of the projection matrix. Additionally, the bandwidth increases as we move toward lower-order singular vectors (associated with smaller singular values), since these directions carry less information for matching scores and can be more freely adjusted to optimize skewness. Lastly, to compare the skewness of the projection matrix across different methods, we analyze the distribution of the Euclidean norms of the projected feature vectors in the training set. Figure 10 illustrates this distribution, where the projected modalities are normalized so that the average norm equals 1. We observe that in *Tanh Similarity*, the projected vectors tend to have more

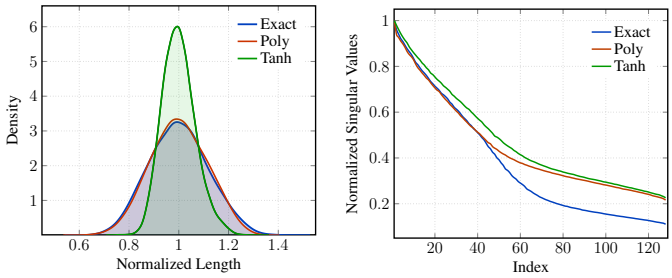
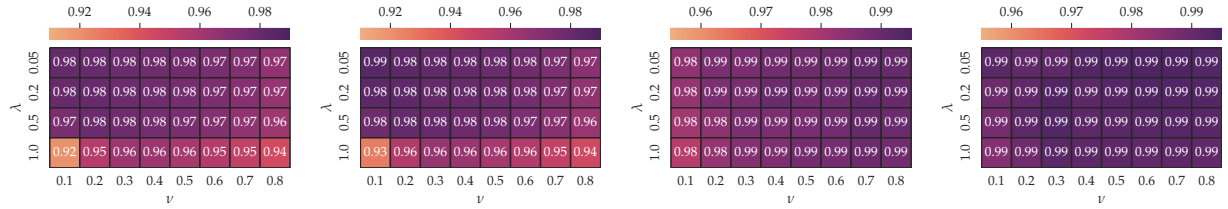


Fig. 10: Distribution of projected (concatenated) features (Left) and Singular values (Right) for Projection matrices obtained by exact normalization and approximation-aware schemes (*Poly/Tanh Similarity*). The projected features are normalized such that the average projection length is one, while singular values are normalized with respect to the largest singular value. The dataset used is Face $\oplus$ Voice and fusion dimension is $\gamma = 128$.

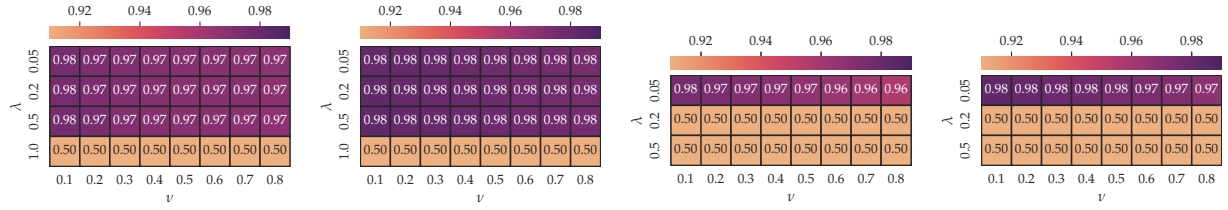
uniform lengths, as no normalization is applied in this method, and large skewness would cause significant errors during the matching phase. Figure 10 shows the distribution of singular values, where we see that in approximation-aware methods, singular values decay more gradually. This helps optimize the skewness of the projection and as mentioned above, this is particularly important for lower singular values, as their corresponding directions contain less information relevant to matching performance. All of these observation,are confirmed in the discussion in §Section III-D5.

Effect of Pre-training: As mentioned in §Section III-E, the implicit presence of the two metrics in the inner product will result in a more complex loss surface which can make the training more difficult. To alleviate this issue, we first pre-train the network with exact normalization to find a fair state for $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$. This stage can be thought as traversing the initial stage of the problem with a surrogate loss (exact normalization) to hopefully start the fine-tune stage at a better position in the original loss surface Equation (3). Figure 12 shows the sensitivity of the hyperparameters in each scenario based on AU-ROC value. It reveals that the model becomes more sensitive to the choice of hyperparameters and only shows progress for a small subset of hyperparameters without pre-training.



(a) face ⊕ voice - $\gamma = 32$ (b) face ⊕ voice - $\gamma = 64$ (c) fingerprint ⊕ face - $\gamma = 32$ (d) fingerprint ⊕ face - $\gamma = 64$

Fig. 11: Heatmap of AU-ROC w.r.t. hyperparameters for the exact normalization fusion.



(a) Pre-Trained - $\gamma = 32$ (b) Pre-Trained - $\gamma = 64$ (c) No Pre-Training - $\gamma = 32$ (d) No Pre-Training - $\gamma = 64$

Fig. 12: Heatmap of AU-ROC w.r.t. hyperparameters for *Tanh Similarity* in face ⊕ voice dataset.

Effects of hyper-parameters and Dimensionality: Figure 11 shows the influence of hyperparameters in the exact learning for face ⊕ voice (a-b) and fingerprint ⊕ face fusion (c-d). We observe that as the fusion dimensionality increases, the sensitivity of the model to hyper-parameter decreases. Regarding the dimensionality of the fusion space, it is desirable to pick a smaller fusion space since it reduces the computational cost of the homomorphic operations. With regard to Figure 7, in face ⊕ voice dataset, we observe that increasing the projection dimension beyond $\gamma = 64$ does not yield significant performance gains. However, in the case of fingerprint/face fusion, where features are noisy, increasing γ to 128 still notably improves the performance. Considering that *Tanh Similarity* approach avoids the normalization step and improves the latency (around 11% based on Table III), choosing a proper fusion dimension and algorithm is a matter of design choice and optimization.

VI. Conclusions

In this paper, we proposed HEFT, the first non-interactive end-to-end homomorphically encrypted multimodal feature-level fusion and matching system. To enhance the practical efficiency of fusion, we conducted a detailed analysis of different data encoding schemes for matrix-matrix and matrix-vector multiplications. To optimize fusion performance and the limitation of FHE, namely the inability to perform non-arithmetic operations, we introduced our FHE-Aware learning algorithm. In this algorithm, we readjust the projection matrix to account either for a low-order polynomial approximation of normalization or for a normalization free approach with the aid of a non-linear transformation ($\tanh(x)$) to map the inner product to the cosine similarity. Our experimental results show that HEFT can overcome the performance losses due to approximations induced by FHE constraints and improve performance over the unibiometric features. On the face⊕voice dataset HEFT improves biometric verification performance by 143.25% and 224.40% for projection on to 32-dim space and 158.21% and 244.35% for projection on to 64-dim space over the unibiometric features, while being practically feasible, taking 1150 ms for fusing a pair of 512-dimensional vectors and matching against a gallery of 1028 templates. On the fingerprint⊕face dataset, HEFT improves verification performance by 24.35% and 13.99% for projection onto 32-dim and 37.99% and 26.49% for projection onto 64-dim space.

In this work, we employed a linear projection for fusion. For future directions, one can use nonlinear projection, as their performance is superior to linear methods. This nonlinear projection must be described in terms of polynomials to make it a possible choice for FHE. Moreover, it should be optimized for efficiency to make it practical. Another direction could be related to the efficiency of score computation. In our method, after fusion, a probe should be compared against the whole gallery of identities. One may employ a clustering approach followed by an approximate comparison operation in FHE to reduce the size of comparisons and make it more efficient.

Acknowledgments: This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation under Grant No. 1841517.

References

[1] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018. 1

[2] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

[3] C. Gentry, S. Halevi, and N. P. Smart, "Fully homomorphic encryption with polylog overhead," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2012. 1, 2

[4] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Annual Cryptology Conference*, 2012. 1

[5] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *Cryptology ePrint Archive*, 2012. 1

[6] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2017. 1, 4

[7] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *IEEE International Conference on Biometrics Theory, Applications, and Systems (BTAS)*, 2018. 1, 2, 5

[8] J. J. Engelsma, A. K. Jain, and V. N. Boddeti, "HERS: Homomorphically encrypted representation search," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 349–360, 2022. 1, 2, 5, 6

[9] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption using image processing," in *Optical Security and Counterfeit Deterrence Techniques II*, vol. 3314, pp. 178–188, 1998. 2

[10] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption," in *ICSA guide to Cryptography*, vol. 22, p. 649, McGraw-Hill New York, 1999. 2

[11] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006. 2

[12] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae, and J. Kim, "A new method for generating an invariant iris private key based on the fuzzy vault system," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1302–1313, 2008. 2

[13] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in *International Conference on Audio-and Video-Based Biometric Person Authentication*, 2005. 2

[14] J. Bringer, H. Chabanne, M. Izabachene, D. Pointcheval, Q. Tang, and S. Zimmer, "An application of the goldwasser-micali cryptosystem to biometric authentication," in *Australasian Conference on Information Security and Privacy*, 2007. 2

[15] Manisha and N. Kumar, "Cancelable biometrics: a comprehensive survey," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3403–3446, 2020. 2

[16] T. A. T. Nguyen, T. K. Dang, and D. T. Nguyen, "A new biometric template protection using random orthonormal projection and fuzzy commitment," in *Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019 13*, pp. 723–733, Springer, 2019. 2

[17] V. K. Hahn and S. Marcel, "Towards protecting face embeddings in mobile face verification scenarios," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 117–134, 2022. 2

[18] H. O. Shahreza, V. K. Hahn, and S. Marcel, "Mlp-hash: Protecting face templates via hashing of randomized multi-layer perceptron," in *2023 31st European Signal Processing Conference (EUSIPCO)*, pp. 605–609, IEEE, 2023. 2

[19] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2017. 2

[20] B. Yalavarthi, A. R. Kaushik, A. Ross, V. Boddeti, and N. Ratha, "Enhancing privacy in face analytics using fully homomorphic encryption," in *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–9, IEEE, 2024. 2

[21] T. Zhou, D. Chen, W. Liu, and X. Yang, "Attacks and improvement of unlinkability of biometric template protection scheme based on bloom filters," *IEEE Transactions on Cloud Computing*, vol. 11, no. 3, pp. 3251–3261, 2023. 2

[22] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge University Press, 2009. 2

[23] M. Barni, G. Droandi, and R. Lazzeretti, "Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 66–76, 2015. 2

[24] J. R. Troncoso-Pastoriza, D. Gonzalez-Jimenez, and F. Perez-Gonzalez, "Fully private noninteractive face verification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1101–1114, 2013. 2

[25] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Efficient biometric verification in encrypted domain," in *International Conference on Biometrics (ICB)*, 2009. 2

[26] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Blind authentication: a secure crypto-biometric verification protocol.," *IEEE Transactions Information Forensics and Security*, vol. 5, no. 2, pp. 255–268, 2010. 2

[27] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *European Symposium on Research in Computer Security*, 2011. 2

[28] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, *et al.*, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, 2010. 2

[29] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on homomorphic encryption," *Pattern Recognition*, vol. 67, pp. 149–163, 2017. 2

[30] P. Drozdowski, F. Stockhardt, C. Rathgeb, D. Osorio-Roig, and C. Busch, "Feature fusion methods for indexing and retrieval of biometric data: Application to face recognition with privacy protection," *IEEE Access*, vol. 9, pp. 139361–139378, 2021. 2, 9, 10

[31] A. A. Ross and R. Govindarajan, "Feature level fusion of hand and face biometrics," in *Biometric Technology for Human Identification II*, SPIE, 2005. 3

[32] P. P. Sarangi, D. R. Nayak, M. Panda, and B. Majhi, "A feature-level fusion based improved multimodal biometric recognition system using ear and profile face," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 4, pp. 1867–1898, 2022. 3

[33] Y. Xin, L. Kong, Z. Liu, C. Wang, H. Zhu, M. Gao, C. Zhao, and X. Xu, "Multimodal feature-level fusion for biometrics identification system on iomt platform," *IEEE Access*, vol. 6, pp. 21418–21426, 2018. 3

[34] S. Siena, V. N. Boddeti, and B. V. Kumar, "Maximum-margin coupled mappings for cross-domain matching," in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013. 3, 7

[35] S. Siena, V. N. Boddeti, and B. Vijaya Kumar, "Coupled marginal fisher analysis for low-resolution face recognition," in *European Conference on Computer Vision Workshops (ECCVW)*, 2012. 3

[36] P. H. Silva, E. Luz, L. A. Zanlorensi, D. Menotti, and G. Moreira, "Multimodal feature level fusion based on particle swarm optimization with deep transfer learning," in *IEEE Congress on Evolutionary Computation (CEC)*, 2018. 3

[37] L. C. O. Tiong, S. T. Kim, and Y. M. Ro, "Multimodal facial biometrics recognition: Dual-stream convolutional neural networks with multi-feature fusion layers," *Image and Vision Computing*, vol. 102, p. 103977, 2020. 3

[38] E. Bartuzi, K. Roszczewska, M. Trokielewicz, and R. Białobrzeski, "Mobibits: Multimodal mobile biometric database," in *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2018. 3

[39] Q. Zhang, H. Li, Z. Sun, and T. Tan, "Deep feature fusion for iris and periocular biometrics on mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2897–2912, 2018. 3

[40] N. Alay and H. H. Al-Baity, "Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits," *Sensors*, vol. 20, no. 19, p. 5523, 2020. 3

[41] M. Leghari, S. Memon, L. D. Dhomeja, A. H. Jalbani, and A. A. Chandio, "Deep feature fusion of fingerprint and online signature for multimodal biometrics," *Computers*, vol. 10, no. 2, p. 21, 2021. 3

[42] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Multibiometric secure system based on deep learning," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017. 3

[43] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, *et al.*, "Homomorphic encryption standard," in *Protecting Privacy through Homomorphic Encryption*, 2021. 4

[44] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *USENIX Security Symposium*, 2018. 5

[45] J. J. Engelsma, K. Cao, and A. K. Jain, "Learning a fixed-length fingerprint representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1981–1997, 2019. 6

[46] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6, 9

[47] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2019. 6

[48] M. Babenko and E. Golimblevskaia, "Euclidean division method for the homomorphic scheme ckks," in *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 2021. 6

[49] S. Panda, "Principal component analysis using ckks homomorphic encryption scheme," *Cryptology ePrint Archive*, 2021. 6

[50] G. S. Cetin, Y. Doroz, B. Sunar, and W. J. Martin, "Arithmetic using word-wise homomorphic encryption," *Cryptology ePrint Archive*, 2015. 6

[51] P. Markstein, "Software division and square root using goldschmidt's algorithms," in *Conference on Real Numbers and Computers (RNC)*, 2004. 6
[52] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7
[53] L. Sperling, N. Ratha, A. Ross, and V. N. Boddeti, "Heft: Homomorphically encrypted fusion of biometric templates," in *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–10, IEEE, 2022. 8, 10
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 8
[55] "Microsoft SEAL (release 4.0)." https://github.com/Microsoft/SEAL, 2022. 8
[56] J. H. Cheon, M. Hhan, S. Hong, and Y. Son, "A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret lwe," *IEEE Access*, vol. 7, pp. 89497–89506, 2019. 8
[57] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017. 8
[58] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015. 8
[59] T. Zheng and W. Deng, "Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments," tech. rep., Beijing University of Posts and Telecommunications, 2018. 8
[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 8
[61] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015. 8
[62] S. Crihalmeanu, A. Ross, S. Schuckers, and L. Hornak, "A protocol for multibiometric data acquisition, storage and dissemination," tech. rep., Technical Report, WVU, Lane Department of Computer Science and Electrical ..., 2007. 9
[63] S. Tandon and A. Namboodiri, "Transformer based fingerprint feature extraction," in *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 870–876, IEEE, 2022. 9

Ramin Akbari .earned a B.S. degree in Civil Engineering from the University of Tehran. He got his Master of Science in structural mechanics from Sharif University of Technology. He is pursuing his PhD at Michigan State University in the Department of Computer Science and Engineering. His research interest includes representation learning.

Luke Sperling earned a B.S. degree in Computer Science and Engineering from Michigan State University in 2019. He went on to graduate with an M.S. degree from the same department. He is currently a PhD candidate at Michigan State University. His research interests include secure multiparty computation and distributed systems. He won the best student paper award at the 2022 IEEE International Joint Conference on Biometrics (IJCB).

Nalini K. Ratha (Fellow, IEEE) received the M.Tech. degree in computer science and engineering from IIT Kanpur, Kanpur, India, in 1984, and the Ph.D. degree in computer science from Michigan State University, East Lansing, MI, USA, in 1996. He is an Empire Innovation Professor of Computer Science and Engineering with the University at Buffalo (State University of New York, Buffalo), NY, USA. He has authored more than 100 research papers in the area of biometrics. He has co-authored a popular book on biometrics titled Guide to Biometrics and coedited two books titled Automatic Fingerprint Recognition Systems and Advances in Biometrics: Sensors, Algorithms, and Systems. He was awarded the IEEE Biometrics Council Leadership Award in 2019. He has been the co-chair of several leading biometrics conferences and served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Systems, Man, and Cybernetics-B, the IEEE Transactions on Image Processing, and Pattern Recognition. He has offered tutorials on biometrics technology at leading IEEE conferences and teaches biometrics and security courses. From 2011 to 2012, he was the President of the IEEE Biometrics Council. He is a Fellow of IAPR and an ACM Distinguished Scientist.

Arun Ross (Senior Member, IEEE) received the B.E. degree (Hons.) in computer science from BITS Pilani, India, and the M.S. and Ph.D. degree in computer science and engineering from Michigan State University, where he is the John and Eva Cillag Endowed Chair with the College of Engineering and a Professor with the Department of Computer Science and Engineering. He was with the Faculty of West Virginia University from 2003 to 2012. His expertise is in biometrics, computer vision, and machine learning. He received the Benedum Distinguished Scholar Award for excellence in creative research and the WVU Foundation Outstanding Teaching Award from West Virginia University. He is a recipient of the NSF CAREER Award and was designated a Kavli Fellow by the U.S. National Academy of Sciences by virtue of his presentation at the 2006 Kavli Frontiers of Science Symposia. In recognition of his contributions to the field of pattern recognition and biometrics, he received the JK Aggarwal Prize in 2014 and the Young Biometrics Investigator Award in 2013 from the International Association of Pattern Recognition. He has advocated for the responsible use of biometrics in multiple forums, including the NATO Advanced Research Workshop on Identity and Security in Switzerland in 2018. He testified as an Expert Panelist in an event organized by the United Nations Counter-Terrorism Committee at the UN Headquarters in 2013.

Vishnu Naresh Boddeti (Member, IEEE) is an Associate Professor in the computer science department at Michigan State University. He received a Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University in 2013. His research interests are Computer Vision, Pattern Recognition, and Machine Learning. He has offered tutorials on multiobjective optimization and Homomorphically encrypted computer vision at leading computer vision conferences. Papers co-authored by him have received the Best Paper Awards at BTAS 2013, GECCO 2019, and IEEE-CCF 2024 and the Best Student Paper Awards at ACCV 2018, SMAIS 2022, IJCB 2022, and TBIOM 2023.