CryptoFace: End-to-End Encrypted Face Recognition

Wei Ao   Vishnu Naresh Boddeti
Michigan State University
{aowei, vishnu}@msu.edu

Abstract

Face recognition is central to many authentication, security, and personalized applications. Yet, it suffers from significant privacy risks, particularly arising from unauthorized access to sensitive biometric data. This paper introduces CryptoFace, the first end-to-end encrypted face recognition system with fully homomorphic encryption (FHE). It enables secure processing of facial data across all stages of a face-recognition process—feature extraction, storage, and matching—without exposing raw images or features. We introduce a mixture of shallow patch convolutional networks to support higher-dimensional tensors via patch-based processing while reducing the multiplicative depth and thus inference latency. Parallel FHE evaluation of these networks ensures near-resolution-independent latency. On standard face recognition benchmarks, CryptoFace significantly accelerates inference and increases verification accuracy compared to the state-of-the-art FHE neural networks adapted for face recognition. CryptoFace will facilitate secure face recognition systems requiring robust and provable security. The code is available at https://github.com/human-analysis/CryptoFace.

1. Introduction

Face Recognition (FR) [13, 26, 44] has become integral to identity management in many practical applications, from unlocking personal devices to facilitating law enforcement and accessing financial services. These systems process sensitive biometric data that, if compromised, can lead to privacy invasions, identity theft, and unauthorized surveillance. Unlike passwords, biometric data is immutable—once compromised, it cannot be changed, which elevates the need for robust security mechanisms to protect it. Such protections are also mandated by legal regulations on the acquisition, storage, and usage of biometric data [34], *e.g*. the European Union's General Data Protection Regulation (GDPR) [43]. FR systems in the wild consist of three entities: a probe face image, a feature extractor (*i.e*. a FR neural network), and a reference database of face features. The feature extractor processes a face image to generate its corresponding compact feature representation.

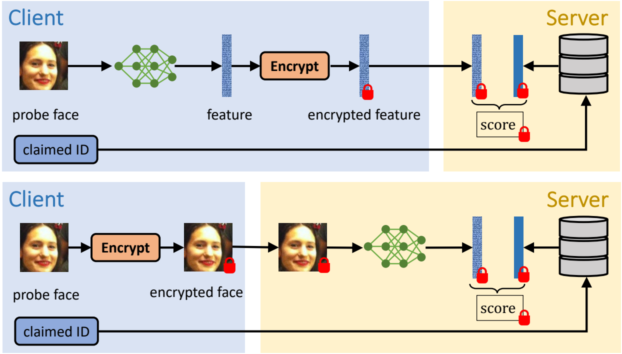


Figure 1. Secure FR systems. (1) Top: existing secure FR system which encrypts only the features, rather than raw face images, limits its practical utility and security; (2) bottom: proposed end-to-end encrypted face recognition system which ensures stronger protection of the user's face image and feature while safeguarding the server's reference database.

Secure FR Systems. FR systems meet increasing security challenges. An adversarial client could attempt to infer biometric data from the server's reference database or leverage the feature extractor to generate adversarial probes to deceive the system. Similarly, an adversarial server might exploit client-provided data to extract biometric information or infer sensitive attributes such as gender, age, or race. Existing secure FR systems apply homomorphic encryption (HE) in a client-server two-party scenario [3, 15] to ensure security of *face features*. HE allows computation over encrypted data and provides provable post-quantum security [7, 9, 16, 17, 32]. As shown in Figure 1 (top), the client's probe feature and the reference database at a server are encrypted. The client performs feature extraction locally and cannot delegate this task to the server. Verification is performed by the server directly within the encrypted domain. Limiting security measures to *feature protection only* reduces the practical utility of such systems. Moreover, such a secure FR system is vulnerable to a template recovery attack [2] where an adversarial client could attempt to infer features in the server's reference database. Unlike HE, which provides robust and provable security guarantees, other lines of research in privacy-preserving face recognition focus on different security or privacy issues, such as adversarial attacks [12], viewing attacks [35], and information leakage [23].

CryptoFace. To address security vulnerabilities in existing secure FR systems, we introduce CryptoFace, the *first end-to-end encrypted* FR system using fully homomorphic encryption (FHE). All operations–including feature extraction, feature matching, and score comparison–are performed entirely within the encrypted domain without decryption at any point. As shown in Figure 1 (bottom), the client encrypts the probe face image and sends it to the server. The server uses a neural network (NN) to extract an encrypted feature. This encrypted feature is matched against the encrypted reference feature stored in the server's database, producing an encrypted similarity score. The score is compared against a threshold, and the encrypted match/no-match result is returned to the client, who alone can decrypt it. By securing the user's sensitive biometric data throughout the verification process and protecting the server's reference database, CryptoFace offers robust end-to-end security for FR systems.

Realizing this goal presents three major technical challenges: (1) Homomorphic evaluation of state-of-the-art (SoTA) convolutional neural networks (CNNs) is computationally demanding due to the high multiplicative depth of CNNs [1, 7, 9, 28]. (2) Although several approaches [1, 18, 28] have demonstrated homomorphic evaluation of CNNs on low-resolution images, they cannot directly process higher-resolution face images. (3) FR requires the cosine similarity measure [13], which cannot be directly computed on FHE. Existing secure FR [3, 15] circumvented the evaluation of the cosine function under FHE by normalizing the features before encryption.

CryptoFaceNet is designed to address these technical challenges. A face image is divided into a grid of non-overlapping patches [14], and each is processed independently by a shallow patch CNN (PCNN). The mixture of PCNNs is jointly trained to learn the inter-patch relationships. Due to the lower resolution of individual patch, we reduce the multiplicative depth required for each PCNN. We also optimize convolutional blocks to further minimize the multiplicative depth and adopt other recent advancements for efficient FHE convolution [28] and low-degree polynomial [39] activation functions. Under FHE, the mixture of PCNNs is evaluated in parallel and features are additively aggregated, significantly accelerating the feature extraction process. CryptoFaceNet scales effectively to high-resolution face images and maintains near-resolution-independent latency due to parallelism. Additionally, we design a distribution-aware low-degree polynomial approximation of the cosine similarity function to efficiently compute the similarity score under FHE.

We evaluate CryptoFace on standard FR benchmarks, comparing its performance to SoTA FHE CNNs [1, 28] adapted for FR. Our results show that CryptoFace not only improves the one-to-one verification accuracy by up to $+8.8\%$ but also speeds up the encrypted FR by $7\times$. CryptoFace supports arbitrary-resolution images, maintaining near-constant latency across different resolutions. We summarize the **contributions** of this paper below:

1. **End-to-End Encrypted Face Recognition:** CryptoFace is the first secure FR system to perform feature extraction, feature matching, and score thresholding entirely within the encrypted domain, eliminating the need for decryption at any stage. CryptoFace enhances security and expands the applicability of secure FR.
2. **Efficient and Scalable Architecture:** CryptoFaceNet is a novel FHE-compatible architecture that reduces computational overhead by minimizing multiplicative depth and is scalable to high-resolution face images.
3. **Feasibility and Efficacy Demonstration:** We present the first practical implementation of end-to-end encrypted face recognition, demonstrating its feasibility on standard FR benchmarks under FHE.

2. Background and Related Work

2.1. Homomorphic Encryption (HE)

Homomorphic encryption (HE) is a class of encryption schemes that are considered quantum-secure and enable computations on encrypted data without requiring decryption. HE schemes are based on the Learning with Errors (LWE) problem [16, 17] or Ring Learning with Errors (RLWE) [32]. Among different HE schemes, the Cheon-Kim-Kim-Song (CKKS) encryption scheme [7–9] is particularly well-suited for encrypted inference in neural networks since it supports fixed-point approximate arithmetic over complex and real numbers.

Encryption and Decryption. A *cleartext* message vector $\mu \in \mathbb{C}^{\frac{N}{2}}$ is first encoded into a *plaintext* message m, which is subsequently encrypted into a *ciphertext* $\boldsymbol{c}$ using a public key $pk = (-\langle a, sk \rangle + e, a)$. Here, $\langle \cdot, \cdot \rangle$ is dot product operator, a is a random ring, e is an encryption noise, and sk is the secret key. The encryption and decryption processes are defined as follows:

$$\begin{aligned} &\text{Encrypt}(m, pk) = (m, 0) + pk = (c_0, c_1) = \boldsymbol{c} \\ &\text{Decrypt}(\boldsymbol{c}, sk) = c_0 + \langle c_1, sk \rangle = m + e \end{aligned} \tag{1}$$

The CKKS scheme uses a residue cyclotomic polynomial ring $\mathcal{R}_{Q_\ell} = \mathbb{Z}_{Q_\ell}[X]/(X^N + 1)$ to encode cleartext vectors. The modulus is defined as $Q_\ell = \prod_{i=0}^{\ell} q_\ell, 0 \leq \ell \leq L$. The polynomial degree N determines the message capacity, allowing $\frac{N}{2}$ complex numbers to be packed into $\frac{N}{2}$ slots.

Supported Operations. The CKKS scheme supports two homomorphic operations—addition and multiplication—and one automorphic operation, rotation. These operations are defined as follows [7, 28]:

$$\begin{gathered}
\underbrace{[m_1] \oplus [m_2]}_{\text{ciphertext-ciphertext}} \approx \underbrace{[m_1] \oplus m_2}_{\text{ciphertext-plaintext}} \approx [\underbrace{m_1 + m_2}_{\text{element-wise add}}] \\
\underbrace{[m_1] \otimes [m_2]}_{\text{ciphertext-ciphertext}} \approx \underbrace{[m_1] \otimes m_2}_{\text{ciphertext-plaintext}} \approx [\underbrace{m_1 \times m_2}_{\text{element-wise mul}}] \qquad (2) \\
\mathrm{Rot}([m], r) = [\mathrm{Rot}(m, r)]
\end{gathered}$$

Here, $[\cdot]$ represents an encrypted message or vector and $\mathrm{Rot}(\cdot)$ denotes a left cyclical rotation of the vector by r positions. Homomorphic addition ($\oplus$) and multiplication ($\otimes$) can be applied between two ciphertexts or between a ciphertext and plaintext, enabling element-wise computations.

Multiplicative Level and Depth. Each ciphertext is associated with a level ℓ, an integer indicating the number of homomorphic multiplications that can be performed before decryption fails. A function with a multiplicative depth k–defined as the number of sequential homomorphic multiplications it involves—consumes k levels. After each multiplication, the polynomial modulus Q_ℓ must be rescaled, transitioning from Q_ℓ to $Q_{\ell-1}$ to maintain the scale [7].

LHE and FHE. CKKS without bootstrapping is a leveled homomorphic encryption (LHE) scheme, allowing a limited number of multiplications determined by the initial level L of a freshly encrypted ciphertext. To evaluate functions with arbitrary depth, fully homomorphic encryption (FHE) incorporates a bootstrapping operation [4, 8, 30] to *refresh* ciphertexts, effectively resetting their level to enable further computation. For deeper neural networks, bootstrapping must be periodically applied to prevent decryption failures. However, this process is computationally expensive, with high latency due to the large number of rotation operations involved. Additionally, bootstrapping has a significant memory footprint because of the large size of the bootstrapping operators. While a freshly encrypted ciphertext starts with L levels, bootstrapping reduces the available levels to $L - K$, as K levels are consumed during the evaluation of polynomials required for bootstrapping.

Computational Complexity. Bootstrapping is slower than rotation or ciphertext-ciphertext multiplication by two orders of magnitude [25]. Ciphertext-ciphertext multiplication is slower than ciphertext-plaintext multiplication or addition by two orders of magnitude [25].

2.2. Homomorphic CNNs (HCNNs)

Homomorphic CNNs (HCNNs) are CNNs that are compatible with the operations that HE supports in Equation 2. We categorize HCNNs into FHENets and LHENets depending on whether bootstrapping is used or not, respectively. LHENets include CryptoNets [18], LoLa [5], Faster CryptoNets [11], while recent FHENets include MPCNN [28] and AutoFHE [1]. FHENets achieve SoTA prediction accuracy on image classification datasets but with much higher latency than LHENets. Existing HCNNs to speed up the inference on encrypted images have focused on two aspects, packing for convolutions and polynomial activation.

Packing for Convolutions refers to efficiently packing three-dimensional tensors to reduce the complexity of HE multiplication and rotation for convolutional layers [24, 28]. MPCNN designs multiplexed convolution by integrating (1) repeated packing and (2) multiplexed packing [28]. (1) A ciphertext with the cyclotomic polynomial degree N can pack $\frac{N}{2}$ numbers. Given a vector $x \in \mathbb{R}^d$ with $d < \frac{N}{2}$, a repeated vector is $x^{(M)} = [x, x, \cdots, x]$ with $M = \lfloor \frac{N}{2d} \rfloor$ copies of x. $x^{(M)}$ is encrypted to fill out all slots. The repeated packing can accelerate convolution and bootstrapping operations. A larger M leads to faster inference. (2) When the convolutional stride exceeds 1, gaps between valid values are introduced, causing some ciphertext slots to remain unused. MPCNN addresses this issue by packing numbers from different channels into alternate slots, effectively filling these gaps and fully utilizing all ciphertext slots, thereby preventing sparsity. Additionally, channels are computed in parallel to speed up convolutional layers.

Polynomial Activation. HCNNs cannot employ ReLU since it is not a homomorphism. So, they adopt polynomial activations (e.g., monomial, Chebyshev or Hermite polynomials) to replace ReLU. Monomial polynomials are widely used since they allow HCNNs to be formulated as traditional polynomial networks. Examples include CryptoNets [18], LoLa [5], and Faster CryptoNets [11]. However, training with monomial polynomials often becomes unstable due to exploding gradients. Minimax approximations using Chebyshev polynomials can achieve high-precision approximations of ReLU functions [27–29], but their high polynomial degree leads to prohibitively large multiplicative depths. AESPA [39] addresses this issue by introducing low-degree Hermite polynomials to reduce the multiplicative depth and proposes basis-wise normalization to stabilize training. Furthermore, search-based AutoFHE [1] explores layer-wise mixed-degree polynomials to further decrease multiplicative depth.

CryptoFaceNet Modules. We build upon the above mentioned prior advances for accelerating HCNN inference, specifically adopting multiplexed convolution [28] and low-degree, basis-wise normalized Hermite activation [39]. Our primary focus, however, lies in addressing other outstanding challenges of end-to-end encrypted FR, including processing high-resolution encrypted images and minimizing the multiplicative depth to reduce computationally expensive bootstrapping operations. We overcome these challenges through CryptoFaceNet, a novel architecture design.

3. End-to-End Encrypted FR System

3.1. CryptoFace

FR Models [13, 26, 44] take advantage of the outstanding representation learning ability of CNNs [20] to extract discriminative features. Given a face image $x \in \mathbb{R}^{C \times H \times W}$, a neural network $f_\omega(\cdot)$ with trainable parameters ω, we have feature $y = f_\omega(x) \in \mathbb{R}^d$. To verify if two face images x_1, x_2 belong to the same individual, we compare their features $y_1 = f_\omega(x_1)$ and $y_2 = f_\omega(x_2)$ to obtain their similarity, *i.e.* $\mathrm{Score}(y_1, y_2) = \|\frac{y_1}{\|y_1\|} - \frac{y_1}{\|y_1\|}\|^2 = 2 - 2\frac{y_1 y_2}{\|y_1\|\|y_2\|}$. If $\mathrm{Score}(y_1, y_2)$ is smaller than a predefined threshold T, the two face images are classified as corresponding to the same person. We formulate this process as a Match function:

$$\mathrm{Match}(y_1, y_2) = \mathrm{Score}(y_1, y_2) - \mathrm{T} \tag{3}$$

A neural network $f_\omega(\cdot)$ for FR is trained on a dataset $\mathcal{X}$ with M identities, each consisting of multiple face images. To train, we need to learn the feature center $W \in \mathbb{R}^{M \times d}$ with M d-dimensional feature centers, where d is the predefined feature dimension. We use ArcFace's [13] additive angular margin loss which is a modified cross-entropy loss:

$$\mathcal{L}_{\text{ArcFace}}(\omega) = -\log \frac{e^{s\cos(\theta_i + m)}}{e^{s\cos(\theta_i + m)} + \sum_{j=1, j \neq i}^{M} e^{s\cos\theta_j}} \tag{4}$$

Given a training sample x with identity i, the feature $y = f_\omega(x)$. The cosine similarity values between y and M d-dimensional features of the feature center are obtained. In Equation 4, θ_i is the angle between y and $M[i]$, while θ_j is the angle between y and $M[j], j \neq i$. The additive margin m is added to the angle θ_i. The scalar s increases the capacity of the unit ball.

Face verification involves two phases, *enrollment* and *verification*. In the enrollment stage, the client sends the server a reference face image and corresponding identity. The server employs the trained network to extract the feature from the reference face image and stores the feature and identity. In the verification stage, the client sends a probe face image and a claimed identity to the server. The server employs the same network to extract the feature from the probe face image and compares it with the reference feature indexed by the claimed identity.

Encrypted FR comprises two similar phases in our paper, *offline* and *online* as shown in Figure 2. The offline stage is similar to enrollment, and the online stage is analogous to verification. In the offline stage, the client generates a public key to encrypt the reference face image and sends the encrypted reference face image and the corresponding identity to the server. The server extracts the encrypted feature $[y_1] = f_\omega([x_1])$. The client does not need to wait for the offline stage to complete. However, during the online stage, the client must wait for the inference result; thus, the latency of the online stage is critical for real-world applications. In the online stage, the client encrypts a probe image $[x_2]$ and sends the encrypted probe face image and a claimed identity to the server. The server extracts the encrypted feature $[y_2] = f_\omega([x_2])$. Then, the server computes the match function (Equation 3) over the encrypted features, i.e., $\mathrm{Match}([y_1], [y_2])$, and finally returns the resulting encrypted match result to the client. The client uses the secret key to decrypt the result and check if it is negative or positive to determine a match or no match, respectively.

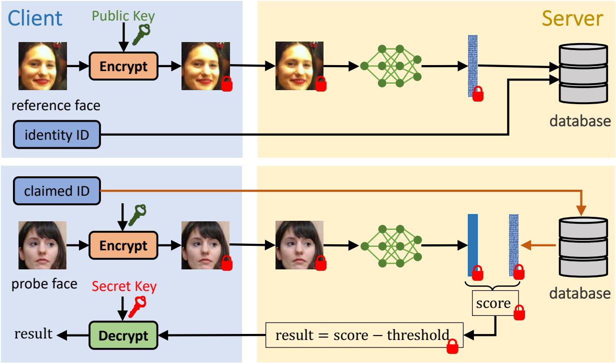


Figure 2. CryptoFace. Top: offline; bottom: online.

Threat Model. Following the threat model used by existing encrypted FR systems [3, 15] and recent FHENets [1, 28], we assume two semi-honest parties: a client and a server. Under this model, at most one of these parties may be corrupted by an adversary [36]. Although both parties follow the agreed-upon protocol honestly, they may attempt to extract additional information by analyzing the data received from each other [36].

- If *the client is adversarial*, it may attempt to infer encrypted features stored on the server. However, since encrypted feature extraction is performed entirely on the server side without releasing intermediate features, the client receives only the matching result—a positive or negative scalar. Consequently, an adversarial client cannot infer the encrypted features stored on the server.
- If *the server is adversarial*, it may attempt to collect biometric data from the client. However, since the server only holds encrypted face images and features without access to the client's secret key, it cannot decrypt or infer any face images or features provided by the client.

3.2. CryptoFaceNet

Inspired by patch-based neural networks [14], CryptoFaceNet applies a mixture of PCNNs to extract local features and fuse these local features to obtain a global feature. Such a design significantly reduces FHE latency due to shallow PCNNs with lower multiplicative depth and parallelized evaluation under FHE.

Training process on cleartext data is shown in Figure 3

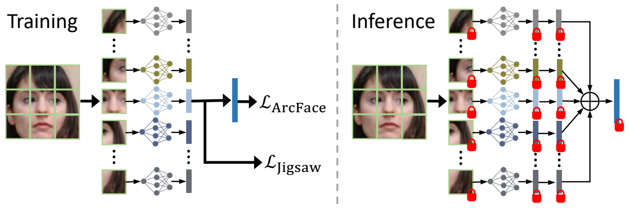


Figure 3. Left: cleartext training; right: encrypted inference.

(left). A two-dimensional face image $x \in \mathbb{R}^{C\times H\times W}$ is divided into a sequence of patches $x_i \in \mathbb{R}^{C\times P\times P} i = 1^L$, where $L = HW/P^2$. The image splitting approach follows that of Vision Transformer (ViT) [14]. A mixture of PCNNs, denoted as $\{f_{\omega_i}\}_{i=1}^L$, independently processes each patch x_i to generate local features $y_i \in \mathbb{R}^d$ for $1 \le i \le L$. These L local features are subsequently fused to form the global feature utilized for FR as follows,

$$y = y'A^T + b, \text{ where } y' = [y_1, y_2, \cdots, y_L] \tag{5}$$

where $A \in \mathbb{R}^{d\times dL}$ and $b \in \mathbb{R}^d$. The patch size is significantly smaller than the original image dimensions, i.e., $P \ll H$ or $P \ll W$, resulting in a reduced receptive field. This allows us to employ shallower PCNNs with lower multiplicative depth for each patch. Typically, each patch covers a small, distinctive facial structure, and the mixture of PCNNs learns separate filters for each patch instead of using weight-sharing [38]. The feature fusion process (Equation 5) encourages the PCNN mixture to capture inter-patch relationships effectively.

To train the mixture of PCNNs, we apply the ArcFace loss defined in Equation 4. Additionally, we introduce a jigsaw puzzle auxiliary task [6, 38, 40] to supplement positional information. This auxiliary task, previously shown to effectively capture positional details [6, 38, 40], naturally complements our mixture of PCNNs. Specifically, we use the local features $y_1, y_2, \cdots, y_L$ to predict their original positions (1, 2, $\cdots$, L) via a fully connected layer. Thus, the learned local features inherently encode positional information. The training objective of the mixture of PCNNs is:

$$\mathcal{L}(\omega, W, A, b) = \mathcal{L}_{\text{ArcFace}}(\omega, W, A, b) + \alpha\mathcal{L}_{\text{Jigsaw}}(\omega) \tag{6}$$

where $\omega = \{\omega_i\}_{i=1}^L$ are the parameters of all PCNNs, W is the feature center, A and b are feature fusion parameters, and α is the strength of jigsaw loss.

Inference. As illustrated in Figure 3 (right), the acceleration achieved by evaluating an encrypted face using the proposed mixture of PCNNs is two-fold. First, the multiplicative depth is significantly reduced, requiring only one bootstrapping operation per PCNN. Second, the evaluation of these L PCNNs can be performed in parallel, simplifying engineering implementation. However, the fusion step involves a large vector-matrix product $y'A^T$ as in Equation 5, which requires numerous computationally expensive homomorphic rotations. The mapping matrix $A \in \mathbb{R}^{d\times dL}$ is rectangular, making it incompatible with efficient HE vector-matrix multiplication approaches designed specifically for square matrices [19]. To address this issue, we rewrite A as $A = [A_1, A_2, \cdots, A_L]$, where each $A_i \in \mathbb{R}^{d\times d}$ for $1 \le i \le L$. Consequently, the vector-matrix multiplication can be decomposed as follows:

$$y = \sum_{i=1}^{L}\left(y_i A_i^T + \frac{b}{L}\right), \text{ where } y_i \in \mathbb{R}^d, A_i \in \mathbb{R}^{d\times d} \tag{7}$$

where the vector-matrix product $y_i A_i^T + b/L$ can be evaluated in parallel. Under FHE, the fusion function reduces to a simple *addition*, the computationally least expensive operation in FHE.

Scalability. Existing HCNNs described in Section 2.2 do not scale effectively to high-resolution face images. Increasing the size of tensors requires enlarging the degree (N) of the residue cyclotomic polynomial ring (see Section 2.1), leading to a substantial accumulation of latency. This occurs because ciphertext multiplication and rotation complexity under FHE scales as $\mathcal{O}(\ell^2 N \log N)$ [31]. In contrast, the proposed mixture of PCNNs is highly scalable to higher resolutions. By increasing the number of PCNNs, our approach achieves near-resolution-independent inference speed, due to the novel and efficient parallel evaluation strategy we introduce.

3.3. Homomorphic Architecture

Convolutional Block. As discussed in Section 2.2, we adopt FHE convolution from MPCNN [28] and the Hermite polynomial activation HerPN introduced by AESPA [39]. Figure 4 shows AESPA block and its FHE implementation provided by [1]. The AESPA block depth is 8 since one Conv consumes 2 levels, and one HerPN consumes 2 levels. We propose a depth-optimal *shifted* AESPA block. Figure 4 shows CryptoFaceNet blocks for $\text{stride} = 1$ and $\text{stride} = 2$. HerPN can be formulated as a degree-2 polynomial $ax^2 + bx + c$ with depth 2. We fuse the coefficient a to Conv weight and change the polynomial to $x^2 + \frac{b}{a}x + \frac{c}{a}$ with depth 1. Therefore, the proposed CryptoFaceNet block can save two levels.

Polynomial ℓ_2 Normalization. When computing the similarity score between two features y_1 and y_2, normalization of these features as $\frac{y_1}{\|y_1\|}$ and $\frac{y_2}{\|y_2\|}$ is necessary (Equation 3). ℓ_2 normalization can be expressed as $\frac{y}{\|y\|_2} = y \cdot \frac{1}{\sqrt{\sum_{i=1}^d y_i^2}}$. Under FHE, $[y_i]^2$ can be computed via ciphertext-ciphertext multiplication, and the summation $\sum_{i=1}^d [y_i]^2$ can be obtained using rotations. However, the primary challenge lies in approximating the non-linear function $q(t) = \frac{1}{\sqrt{t}}$. This is difficult as the domain of $q(t)$ is typically very wide (since

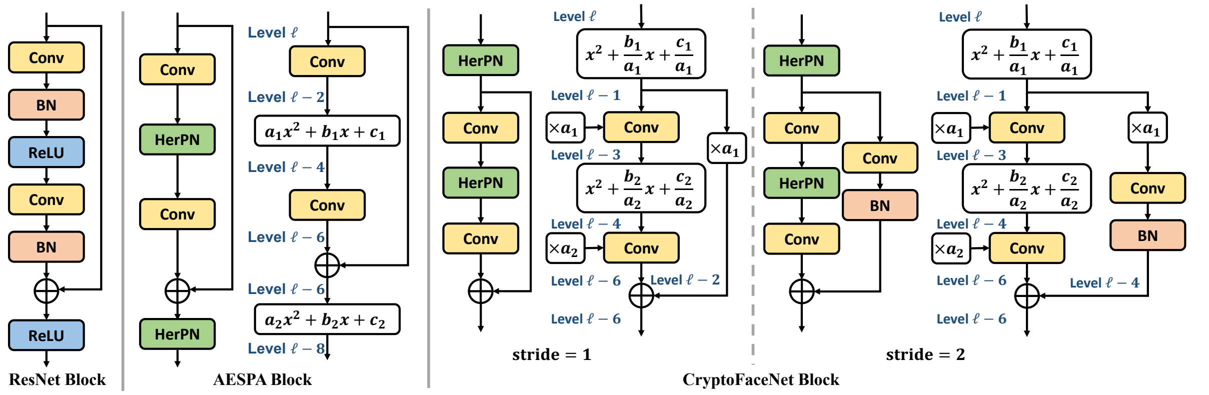


Figure 4. Convolutional blocks and their FHE implementations. Left: ResNet [20]; middle: AESPA [39]; right: CryptoFaceNet.

$t = |y|_2^2$). A Taylor expansion around $t = t_0$ would result in a polynomial with excessively high degree, and minimax approximation [27] is not directly applicable since $q(t)$ cannot be effectively scaled into the $[-1, 1]$ interval it requires. Observing that $q(t) = \frac{1}{\sqrt{t}}$ is a strictly decreasing function and we propose using a simpler polynomial approximation: $p(t) = \beta_2 t^2 + \beta_1 t + \beta_0$ to approximate $q(t)$. Unlike the minimax approximation that explicitly minimizes $|p(t) - q(t)|$, we instead ensure $p(t)$ has a similar shape to $q(t)$ within its relevant domain. We achieve this by selecting three control points t_1, t_2, and t_3, along with their corresponding reference values $\frac{1}{\sqrt{t_1}}$, $\frac{1}{\sqrt{t_2}}$, and $\frac{1}{\sqrt{t_3}}$, respectively. Solving these equations yields coefficients β_2, β_1, and β_0 that fit our polynomial to the distribution of data. Specifically, we set the control points based on the distribution of t as follows: $t_1 = \text{Mean}(t) - \text{Std}(t)$, $t_2 = \text{Mean}(t)$, $t_3 = \text{Mean}(t) + \text{Std}(t)$. Thus, the resulting polynomial $p(t)$ is distribution-aware, and its coefficients (β_2, β_1, and β_0) are determined by the underlying data distribution—similar to the score thresholding procedure. The polynomial approximation has a multiplicative depth of only 2, ensuring computational efficiency. Consequently, we re-estimate the matching threshold (i.e., T in Equation 3) to accommodate the polynomial-based ℓ_2 normalization.

Adaptive Average Pooling. MPCNN [28] originally employs adaptive average pooling with an output size of $(1, 1)$. To better preserve structural information crucial for FR, we customized it to have an output size of $(2, 2)$, resulting in 256-dimensional features. Our $(2, 2)$ pooling is implemented as four separate $(1, 1)$ pooling operations, which permute elements of the feature vector. So, we also rearranged the fusion matrix A accordingly (Equation 5).

CryptoFaceNet Architecture is shown in Figure 5. To reduce depth consumption, CryptoFaceNet fuses the Linear and BatchNorm1D layers to a single Linear. The aggregation of features is a simple ciphertext addition. CryptoFaceNet only uses one bootstrapping operation.

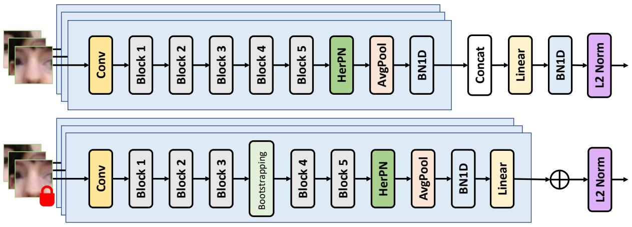


Figure 5. CryptoFaceNet. Top: training; bottom: inference.

4. Experiments

Datasets. (1) Training dataset: we use WebFace4M, a subset of WebFace260M [48] to train CryptoFace and baselines on cleartext data in Pytorch. WebFace4M has 205,990 identities and 4,235,242 face images in total. (2) Test datasets: we follow AdaFace [26] to benchmark CryptoFace and baselines on five standard test datasets [22]: LFW [21], AgeDB [37], CALFW [47], CPLFW [46], CFP-FP [42]. The image resolution is 112 × 112. In our experiments, we resize face images to small (64 × 64), medium (96 × 96), and high-resolution (128 × 128), which satisfies use cases corresponding to edge FR, embedded FR, and cloud FR in the wild. CFP-FP has 7,000 pairs, resulting in 14,000 face images, while others have 6,000 pairs, resulting in 12,000 face images. We report the one-to-one verification accuracy on the encrypted test datasets.

FHE Library and Hardware. We follow MPCNN [28] and AutoFHE [1] to adopt SEAL [41] library and report latency and RAM footprint on Amazon AWS r5.24xlarge. The modified SEAL [28] incorporates bootstrapping.

Baselines. As there is no prior end-to-end encrypted FR, we adopt two FHENets MPCNN and AutoFHE (see Section 2.2) as our baselines because they report the SoTA performance on CIFAR image classification. Our C++ implementation of CryptoFace is built on top of MPCNN and AutoFHE. To take 64 × 64 face images as input, we change the stride of the very first convolutional layer from 1 to 2 to ensure that a single ciphertext can pack any intermediate tensors. The new output layers used for FR are

Method	Backbone				Dataset						**Latency**(s)	**RAM**
	Network	Params	#Boot	Res	LFW	AgeDB	CALFW	CPLFW	CFP-FP	Avg		
MPCNN [28]	ResNet32	0.53M	31	64	97.02	83.02	87.00	78.90	82.07	85.60	7,367	286G
	ResNet44	0.72M	43	64	98.27	87.45	90.85	83.72	87.90	89.64	9,845	286G
AutoFHE [1]	ResNet32	0.53M	8	64	93.53	80.88	85.40	75.67	77.96	82.69	4,001	286G
CryptoFace	CryptoFaceNet4	0.94M	1	64	98.87	89.45	91.60	81.98	85.21	89.42	1,364	269G
	CryptoFaceNet9	2.12M		96	99.18	91.38	93.32	84.23	86.81	90.99	1,395	276G
	CryptoFaceNet16	3.78M		128	98.78	92.90	93.73	83.95	87.94	91.46	1,446	277G

Table 1. Experiments on end-to-end FR on FHE.

Method	Backbone	Conv	BN	Residual	AvgPool	Linear	Activation	L2 Norm	Match	Bootstrapping	Other
MPCNN [28]	ResNet32	896s (12.17%)	28s (0.38%)	0.5s (0.01%)	46s (0.62%)	807s (10.96%)	583s (7.92%)	5s (0.07%)	2s (0.02%)	4991s (67.76%)	7s (0.09%)
	ResNet44	1214s (12.33%)	39s (0.39%)	0.7s (0.01%)	46s (0.46%)	807s (8.19%)	807s (8.20%)	5s (0.05%)	2s (0.02%)	6917s (70.27%)	7s (0.08%)
AutoFHE [1]	ResNet32	1966s (49.14%)	28s (0.69%)	1.7s (0.04%)	38s (0.95%)	658s (16.43%)	17s (0.43%)	4s (0.10%)	1s (0.03%)	1274s (31.84%)	14s (0.34%)
CryptoFace	CryptoFaceNet4	858s (62.93%)	2s (0.16%)	0.1s (0.01%)	26s (1.94%)	277s (20.30%)	13s (0.94%)	2s (0.11%)	0.3s (0.02%)	141s (10.34%)	44s (3.26%)

Table 2. Latency of operations on FHE for 64×64 encrypted face images.

AdaptiveAvgPool2d((2, 2)) $\mapsto$ Linear(256, 256) $\mapsto$ BatchNorm1d(256). AutoFHE is a search-based approach and reports search results on the CIFAR dataset. Extending the search algorithm from a small CIFAR dataset to the much larger WebFace dataset is challenging, so we use the search result on CIFAR and transfer it to the WebFace dataset.

Parameters. (1) Training: we adopt the parameters from AdaFace [26] without tuning. We set the learning rate to 0.05, epochs to 26, batch size to 256, momentum to 0.9, and weight decay to 0.0005. We use the SGD optimizer with multi-step scheduler. The learning rate is scaled by $0.1\times$ at epochs 12, 20, and 24. When we train CryptoFace and AutoFHE, we clip the gradient to 1 as suggested by AutoFHE. The patch size is set to 32×32. We set the strength (α) of the jigsaw loss to 0.005. (2) FHE: To meet 128-bit security [10], we use the same CKKS parameters as MPCNN and AutoFHE. The degree of the cyclotomic ring is 2^{16} and Hamming weight is 192. We set the default modulus to 46 bits and the special modulus to 51 bits [1, 28].

4.1. End-to-End Encrypted FR on FHE

We follow the standard 10-fold cross-validation [13, 26] to benchmark CryptoFace and baselines on the encrypted face datasets. For each split, nine groups (cleartext) are used to estimate the threshold and polynomial approximation of ℓ_2 normalization, while the standalone group (ciphertext) is used to test performance. We report the average verification accuracy of 10-fold cross-validation for each dataset as shown in Table 1. For a pair of face images, we consider the first as the reference and the second as the probe. Each experiment includes an offline and an online stage (see Section 3.1). In Table 1, we report the online latency. We use numbers of patches (*i.e.* 4, 9, and 16) to denote CryptoFaceNet for different resolutions 64×64, 96×96, and 128×128, respectively.

For encrypted face images at resolution 64×64, MPCNN with ResNet44 shows the highest accuracy (89.64%) but a prohibitively large latency (9,845 seconds). CryptoFace greatly accelerates encrypted face recognition by $\mathbf{7.2}\times$ which translates to savings of 8,481 seconds. We only observe a negligible accuracy drop of 0.22%. MPCNN with ResNet32 is a faster version but only achieves 85.60% accuracy. CryptoFace speeds up inference by $\mathbf{5.4}\times$ and increases FR performance by $+\mathbf{3.82}\%$. AutoFHE is much faster than MPCNN. However, the transferred AutoFHE achieves 82.64% on encrypted FR with a latency of 4,001 seconds. Compared to AutoFHE, CryptoFace accelerates inference by $\mathbf{2.9}\times$ and improves the encrypted FR performance by $+\mathbf{6.73}\%$. CryptoFace also reduces RAM footprint by **17G** since we only need one bootstrapping operation, while the baselines require three operations for different repeated packing copies M (see Section 2.2). The experimental results demonstrate the effectiveness of CryptoFace, the proposed end-to-end encrypted FR. The proposed CryptoFaceNet is an efficient FHENet thanks to the mixture of PCNNs with simple, yet effective, parallelization.

We also analyze how the resolution of encrypted face images impacts verification accuracy, latency, and RAM footprint, as shown in Table 1. When the resolution increases from 64×64 to 96×96 and 128×128, CryptoFace can take advantage of high-resolution face images to effectively improve FR performance. Compared to CryptoFaceNet4, CryptoFaceNet9 and CryptoFaceNet16 increase the accuracy by $+\mathbf{1.57}\%$ and $+\mathbf{2.04}\%$, respectively. CryptoFace can maintain a nearly constant latency (1,364 to 1,446 seconds) even as the image resolution increases from 64×64 to 128×128. CryptoFace slightly increases the RAM footprint from 269G to 277G. The results demonstrate that CryptoFace is scalable to high-resolution images and can satisfy different requirements in real-world secure FR applications.

4.2. Operation Latency

Table 2 lists detailed latency of different operations. Bootstrapping operations dominate the latency of MPCNN, around 70%. AutoFHE successfully removes most bootstrapping operations by using mixed-degree polynomial activations. CryptoFace fundamentally decreases the depth of networks and significantly reduces the number of bootstrap-

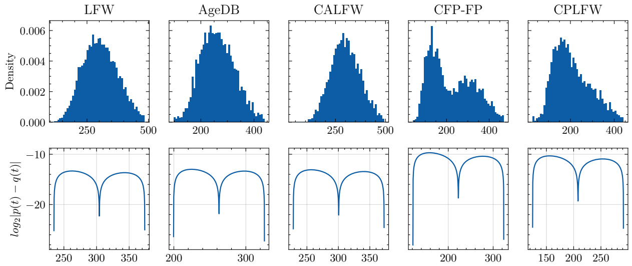


Figure 6. ℓ_2 polynomial approximation. Top: the distribution (domain) of $q(t) = 1/\sqrt{t}$, bottom: the approximation error $\log_2 |p(t) - q(t)|$.

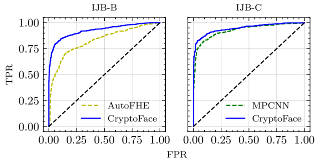
ping operations to *one* with bootstrapping only contributing 10% to the total latency. The parallelism of CryptoFaceNet on FHE introduces an acceptable overhead, around $< 3.26\%$ (Other). CryptoFace and MPCNN (ResNet32) spend roughly equal time evaluating convolutional layers. However, MPCNN (ResNet32) has 31 convolutional layers, and one patch CNN only has 13 (including two residual convolutional layers) since CryptoFaceNet convolutional layers take high-level ciphertext as input, leading to higher latency. Both AutoFHE and CryptoFace benefit from low-degree polynomial activations because expensive ciphertext-ciphertext multiplications are decreased. We apply the average pooling with the output size $(2, 2)$ to get 256-dimensional features (see Section 3.3). The $(2, 2)$ average pooling is slower than $(1, 1)$ averaging pooling used by the original versions of MPCNN and AutoFHE because it introduces more rotations and multiplications. However, FR necessitates high-dimensional features to preserve more structural information. The proposed polynomial approximation of ℓ_2 normalization and the feature matching are very efficient under FHE. CryptoFaceNet is a depth-optimized homomorphic neural architecture (see Section 3.3) demonstrating lower latency over different FHE operations.

4.3. Polynomial L2 Approximation

Figure 6 (top) shows the distribution of $\|y\|_2^2$, namely the domain of $q(t) = 1/\sqrt{t}$. We propose a distribution-aware polynomial approximation $p(t) \mapsto q(t)$ (see Section 3.3). Figure 6 (bottom) shows that the approximation error is $\log_2 |p(t) - q(t)| \leq 2^{-10}$. Table 2 shows the latency of the polynomial ℓ_2 approximation is 0.3 to 2 seconds on FHE only consuming 0.02% of inference time. Thus, the polynomial ℓ_2 approximation is accurate and efficient.

4.4. Mixed-Quality FR Benchmarks

The five standard FR datasets used in our experiments are regarded as high-quality FR benchmarks. They can satisfy the most real-world secure FR applications. IJB-B [45] and IJB-C [33] FR datasets include mixed-quality face images and are used to test FR models for challenging FR tasks.



IJB-B	AutoFHE	**CryptoFace**
Time	4,001s	1,364s (↓ **2,637**s)
AUC	0.825	0.926 (↑ **0.1**)
IJB-C	MPCNN	**CryptoFace**
Time	7,367s	1,364s (↓ **6,003**s)
AUC	0.927	0.946 (↑ **0.02**)

Figure 7. Experiments on IJB-B and IJB-C.

Figure 7 shows experimental results on encrypted IJB-B and IJB-C on FHE. Due to the prohibitively high computational cost, we randomly sample 1,200 pairs (with 50% positive pairs and 50% negative pairs) from each dataset. We use ROC (Receiver Operating Characteristic) curve and AUC (Area Under the Curve) to qualify the FR performance of CryptoFace and baselines. From Figure 7, CryptoFace consistently shows better performance on IJB-B or IJB-C on FHE compared to AutoFHE and MPCNN.

4.5. Identification

In this paper, we primarily focus on the face verification task, as discussed in Section 3.1. However, CryptoFace can be readily extended to face identification scenarios, such as one-to-many face matching. In Table 3, we report experimental results for a $1 : 128$ closed-set rank retrieval task under FHE, using a randomly selected subset of LFW consisting of 128 pairs. The latency difference solely arises from the feature extraction stage, as previously reported. We evaluate performance using Rank-1 and Rank-5 accuracy as metrics. The experimental results confirm that CryptoFace is also effective for face identification tasks.

Method	Rank-1 Acc.	Rank-5 Acc.
MPCNN	88.28	97.66
CryptoFace	92.19 (↑ **3.91%**)	98.44 (↑ **0.78%**)

Table 3. 1:128 closed-set retrieval.

5. Conclusion

This paper introduced CryptoFace, the first end-to-end encrypted face recognition system using fully homomorphic encryption (FHE). Once face images are encrypted, all subsequent operations like feature extraction, matching, and score thresholding are performed in the encrypted domain without decryption. The key idea behind CryptoFace is CryptoFaceNet, a novel architecture which is a mixture of shallow patch convolutional neural networks optimized for FHE compatibility and mitigating the steep computational burden of encrypted inference. Experimental results on standard face recognition benchmarks show that CryptoFace is 7× faster than SoTA FHENets while achieving better verification performance. CryptoFace can effectively process high-resolution encrypted face images to improve verification accuracy by $+2\%$ while maintaining near-resolution-independent latency. CryptoFace will facilitate the deployment of secure face recognition systems in applications requiring strict privacy and security guarantees.

References

[1] Wei Ao and Vishnu Naresh Boddeti. AutoFHE: Automated adaption of CNNs for efficient evaluation over FHE. In *USENIX Security Symposium*, pages 2173–2190. USENIX Association, 2024. 2, 3, 4, 5, 6, 7

[2] Amina Bassit, Florian Hahn, Zohra Rezgui, Una Kelly, Raymond Veldhuis, and Andreas Peter. Template recovery attack on homomorphically encrypted biometric recognition systems with unprotected threshold comparison. In *IEEE International Joint Conference on Biometrics*. IEEE, 2023. 1

[3] Vishnu Naresh Boddeti. Secure face matching using fully homomorphic encryption. In *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–10. IEEE, 2018. 1, 2, 4

[4] Jean-Philippe Bossuat, Christian Mouchet, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 587–617, 2021. 3

[5] Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha. Low latency privacy preserving inference. In *International Conference on Machine Learning*, pages 812–821, 2019. 3

[6] Yingyi Chen, Xi Shen, Yahui Liu, Qinghua Tao, and Johan AK Suykens. Jigsaw-ViT: Learning jigsaw puzzles in vision transformer. *Pattern Recognition Letters*, 166:53–60, 2023. 5

[7] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437, 2017. 1, 2, 3

[8] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 360–384, 2018. 3

[9] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A full RNS variant of approximate homomorphic encryption. In *International Conference on Selected Areas in Cryptography*, pages 347–368, 2018. 1, 2

[10] Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. *IEEE Access*, 7:89497–89506, 2019. 7

[11] Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. Faster CryptoNets: Leveraging sparsity for real-world encrypted inference. *arXiv preprint arXiv:1811.09953*, 2018. 3

[12] Debayan Deb, Jianbang Zhang, and Anil K Jain. AdvFaces: Adversarial face synthesis. In *IEEE International Joint Conference on Biometrics*, pages 1–10. IEEE, 2020. 2

[13] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, 2022. 1, 2, 4, 7

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2, 4, 5

[15] Joshua J Engelsma, Anil K Jain, and Vishnu Naresh Boddeti. HERS: homomorphically encrypted representation search. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):349–360, 2022. 1, 2, 4

[16] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford University, 2009. 1, 2

[17] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing*, pages 169–178, 2009. 1, 2

[18] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016. 2, 3

[19] Shai Halevi and Victor Shoup. Algorithms in HElib. In *Advances in Cryptology*, pages 554–571. Springer, 2014. 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4, 6

[21] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008. 6

[22] Insightface. https://github.com/deepinsight/insightface, 2023. 6

[23] Jiazhen Ji, Huan Wang, Yuge Huang, Jiaxiang Wu, Xingkun Xu, Shouhong Ding, ShengChuan Zhang, Liujuan Cao, and Rongrong Ji. Privacy-preserving face recognition with learnable privacy budgets in frequency domain. In *European Conference on Computer Vision*, pages 475–491. Springer, 2022. 2

[24] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *USENIX Security Symposium*, pages 1651–1669, 2018. 3

[25] Donghwan Kim, Jaiyoung Park, Jongmin Kim, Sangpyo Kim, and Jung Ho Ahn. HyPHEN: A hybrid packing method and its optimizations for homomorphic encryption-based neural networks. *IEEE Access*, 2023. 3

[26] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 4, 6, 7

[27] Eunsang Lee, Joon-Woo Lee, Young-Sik Kim, and Jong-Seon No. Minimax approximation of sign function by composite polynomial for homomorphic comparison. *IEEE Transactions on Dependable and Secure Computing*, 2021. 3, 6

[28] Eunsang Lee, Joon-Woo Lee, Junghyun Lee, Young-Sik Kim, Yongjune Kim, Jong-Seon No, and Woosuk Choi. Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In *International Conference on Machine Learning*, pages 12403–12422, 2022. 2, 3, 4, 5, 6, 7

[29] Junghyun Lee, Eunsang Lee, Joon-Woo Lee, Yongjune Kim, Young-Sik Kim, and Jong-Seon No. Precise approximation of convolutional neural networks for homomorphically encrypted data. *arXiv preprint arXiv:2105.10879*, 2021. 3

[30] Joon-Woo Lee, Eunsang Lee, Yongwoo Lee, Young-Sik Kim, and Jong-Seon No. High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 618–647, 2021. 3

[31] Qian Lou and Lei Jiang. HEMET: A homomorphic-encryption-friendly privacy-preserving mobile neural network architecture. In *International Conference on Machine Learning*, pages 7102–7110, 2021. 5

[32] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010. 1, 2

[33] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. IARPA janus benchmark-C: Face dataset and protocol. In *International Conference on Biometrics*, pages 158–165. IEEE, 2018. 8

[34] Blaž Meden, Peter Rot, Philipp Terhörst, Naser Damer, Arjan Kuijper, Walter J Scheirer, Arun Ross, Peter Peer, and Vitomir Štruc. Privacy–enhancing face biometrics: A comprehensive survey. *IEEE Transactions on Information Forensics and Security*, 16:4147–4183, 2021. 1

[35] Yuxi Mi, Zhizhou Zhong, Yuge Huang, Jiazhen Ji, Jianqing Xu, Jun Wang, Shaoming Wang, Shouhong Ding, and Shuigeng Zhou. Privacy-preserving face recognition using trainable feature subtraction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 297–307, 2024. 2

[36] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: A cryptographic inference service for neural networks. In *USENIX Security Symposium*, pages 2505–2522, 2020. 4

[37] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. AgeDB: the first manually collected, in-the-wild age database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017. 6

[38] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 5

[39] Jaiyoung Park, Michael Jaemin Kim, Wonkyung Jung, and Jung Ho Ahn. AESPA: Accuracy preserving low-degree polynomial activation for fast private inference. *arXiv preprint arXiv:2201.06699*, 2022. 2, 3, 5, 6

[40] Bin Ren, Yahui Liu, Yue Song, Wei Bi, Rita Cucchiara, Nicu Sebe, and Wei Wang. Masked jigsaw puzzle: A versatile position embedding for vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20382–20391, 2023. 5

[41] SEAL. Microsoft SEAL (3.6). *Microsoft Research*, 2020. 6

[42] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9. IEEE, 2016. 6

[43] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation GDPR. *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017. 1

[44] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 1, 4

[45] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. IARPA janus benchmark-b face dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017. 8

[46] Tianyue Zheng and Weihong Deng. Cross-Pose LFW: a database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5(7):5, 2018. 6

[47] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-Age LFW: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017. 6

[48] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. WebFace260M: A benchmark unveiling the power of million-scale deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021. 6