Improved Multiplication-Free Biometric Recognition under Encryption*

Amina Bassit[1], Florian Hahn[1], Raymond Veldhuis[1,3], Andreas Peter[1,2]

***Abstract*—Modern biometric recognition systems extract distinctive feature vectors of biometric samples using deep neural networks to measure the amount of (dis-)similarity between two biometric samples. Studies have shown that personal information (e.g., health condition, ethnicity, etc.) can be inferred, and biometric samples can be reconstructed from those feature vectors, making their protection an urgent necessity. State-of-the-art biometrics protection solutions are based on homomorphic encryption (HE) to perform recognition over encrypted feature vectors, hiding the features and their processing while releasing the outcome only. However, this comes at the cost of those solutions' efficiency due to the inefficiency of HE-based solutions with a large number of multiplications; for (dis-)similarity measures, this number is proportional to the vector's dimension. In this paper, we tackle the HE performance bottleneck by freeing the two common (dis-)similarity measures, the cosine similarity and the squared Euclidean distance, from multiplications. Assuming normalized feature vectors, our approach pre-computes and organizes those (dis-)similarity measures into lookup tables. This transforms their computation into simple table lookups and summations only. We integrate the table lookup with HE and introduce pseudo-random permutations to enable cheap plaintext slot selection, which significantly saves the recognition runtime and brings a positive impact on the recognition performance. We then assess their runtime efficiency under encryption and record runtimes between 16.74ms and 49.84ms for both the cleartext and encrypted decision modes over the three security levels, demonstrating their enhanced speed for a compact encrypted reference template reduced to one ciphertext.**

***Index Terms*—Biometric verification, fully homomorphic encryption (FHE), single-instruction multiple-data (SIMD).**

I. Introduction

Modern biometric recognition technologies lean on deep neural networks (DNNs) to extract distinctive representations of biometric samples (i.e., facial images), called feature vectors. Thus, the biometric recognition task becomes the comparison of two feature vectors against each other by calculating a (dis-)similarity score, usually via the cosine similarity or the squared Euclidean distance (SED). Studies have shown that from those feature vectors, it is possible to infer personal information (e.g., gender, age, health condition, ethnicity, occupation, etc.) [2] and reconstruct raw biometric samples (e.g., facial images) of individuals, known as model inversion attacks [3]. The plaintext access to those feature vectors enables, on the one hand, an undesirable personal information inference that intensifies the severeness of social issues, such as gender inequality and discrimination, due to biased decision-making models. Such models violate the privacy of individuals by performing classification tasks other than recognition. On the other hand, it permits the reconstruction of the raw biometric sample from a given reference template or probe, which leads to security issues such as identity fraud and impersonation attacks. Therefore, feature vectors extracted from DNNs are extremely sensitive and require strong protection.

Biometric template protection schemes (BTPs) [4] try to protect biometric information (e.g., biometric feature vector) with a maintained recognition performance. BTPs come in different flavors, each approaching the biometrics privacy challenges with distinct techniques (e.g., helper data and Bloom filters). Among existing BTPs, homomorphic encryption (HE) based BTPs [5]–[9] seem promising in tackling these issues since they carry out both the biometric data and its processing to the encrypted domain. Generally, HE-based BTPs compare two encrypted biometric feature vectors against each other and distinguish between two decision modes, namely *cleartext decision mode* and *encrypted decision mode*, which indicate whether the decision occurs within or outside the encrypted domain. Hence, measuring a (dis-)similarity score under encryption is followed by a recognition decision delivered to the party of interest, who, in the cleartext decision mode, receives a final score and performs the comparison with the threshold in the clear to make a decision [5]–[8] or, in the encrypted decision mode, the party of interest receives the recognition decision encrypted for which the comparison with the threshold was performed under encryption [9]. Calculating such a (dis-)similarity score under encryption involves a number of homomorphic multiplications proportional to the feature vector's dimension. The failure of HE to handle computations with many multiplications hurts those solutions' efficiency. For instance, to calculate the SED under encryption using the CKKS encryption scheme [10], [8] takes 2.11s for 128-dimensional encrypted feature vectors while for 512-dimensional vectors [6] runs in 5s and [7] runs in 3.39s for 128bits security level; using another HE scheme, BFV [11], the runtime improves to 0.85s [6] and 0.61s [7] but still improvable.

To reduce the number of homomorphic multiplications, the work [5] adopts the *single-instruction multiple-data* (SIMD) and the plaintext packing properties of fully HE to decrease this number to one homomorphic multiplication for calculating the inner product (IP) under HE over encrypted normalized feature vectors, which corresponds to the cosine similarity. In [5], the author considers only the cleartext decision mode, which has the downside of exposing the final score. The knowledge of the final score can lead to the reconstruction of the original biometric template, as shown by [12]–[15]. In contrast to the encryption decision mode, the comparison with the threshold performed inside the encrypted domain reveals only one bit of information (*match* / *no match*). We compare our approach with [5] in Section VI. The first multiplication-free biometric recognition (MFBR) scheme is the homomorphically encrypted log likelihood-ratio classifier (HELR) introduced in [9]. It pre-computes the log likelihood-ratio (LLR) and organizes it into lookup tables,

{a.bassit, f.w.hahn, r.n.j.veldhuis}@utwente.nl, andreas.peter@uol.de
[1]University of Twente, Enschede, The Netherlands,
[2]University of Oldenburg, Oldenburg, Germany,
[3]Norwegian University of Science and Technology, Gjøvik, Norway

*This paper is an extension of [1] published at IEEE IJCB 2022.

reducing the biometric recognition into three operations: selection of the individual scores from the tables, their addition to calculate a final score, and the comparison of the final score with the biometric threshold. However, to determine a score, this classifier requires prior knowledge about the statistics of the features learned from training the LLR. In general, this prior knowledge is hard to acquire for large-scale applications. Hence, the HELR classifier requires training data, and homomorphic multiplications represent a burden for biometrics, motivating us to tackle these challenges.

In this paper, we extend the work conducted in [1] by improving the integration of the MFBR schemes with HE, as illustrated in Table I, where we denote by MFBRv1 [1] our initial version, and by MFBRv2 our improved one proposed in the present paper. Our solution, as in [1], is built upon the HELR framework [9] but applied to the IP and SED measures that do not require training. Assuming normalized feature vectors extracted from a well-trained DNN, we determine the probability density function (PDF) and cumulative distribution function (CDF) corresponding to the projection of a point on the unit d-ball upon which we generate the lookup tables (that we call MFIP and MFSED) in an equiprobable manner to reinforce their security. We assess the biometric performance of our tables on synthetic and facial feature vectors and achieve a performance identical to their baseline measures, preserving biometric accuracy. Furthermore, the MFBRv1 represents the encrypted reference template by a set of d ciphertexts to facilitate the selection of specific components under encryption by using homomorphic rotations only; however, this comes at the cost of the encrypted reference template storage because, in each of those ciphertexts, $c-N$ plaintext slots are left empty. Given that the reference template requires $d\times N$ plaintext slots to be packed in a single ciphertext and that a ciphertext has sufficient capacity c to pack all $d\times N$ plaintext values into one ciphertext because $c > d \times N$[1], in this work, we optimize the space complexity of the encrypted reference template by making use of the entire ciphertext packing capacity. Thus, we enhance the integration of our MFBRs with HE by reducing the encrypted reference template size from d ciphertexts to one ciphertext and representing the protected probe as a binary mask, facilitating the selection of encrypted individual scores for a more compact encrypted reference template, unlike the initial integration. We evaluate the runtime of the integration of our MFBRv2 with HE and compare it with the MFBRv1 for three security levels. For the cleartext decision mode, the two integration versions achieve comparable efficiency. Our MFBRv2 runs slightly faster, with a speed difference of 9 to 15ms, and the encrypted reference is more compact, 131.9kB to 263kB, instead of 67.3MB to 134.4MB for MFBRv1, over the three security levels. For the encrypted decision mode, using the comparison with the threshold method described in [9], our MFBRv2 outperforms the MFBRv1, being state-of-the-art, requiring tremendously less storage. It decreases the runtime by a factor of two to four and the encrypted reference template storage by two to three orders of magnitude over the three security levels. This is because, in the improved integration MFBRv2, the resulting final score ciphertext contains the final score replicated over the plaintext slots, which represents an important step before comparing the encrypted final score with the encrypted threshold vector, which is a permuted and packed ciphertext. Unlike the initial integration MFBRv1, where a replication step is needed before proceeding to this comparison under encryption, it induces extra computations that influence the runtime. As a result, our improved integration enhances the encrypted reference template storage and closes the runtime gap between the cleartext and encrypted decision modes.

In summary, we make the following contributions:

- We propose two MFBRs implementing IP and SED comparison measures that do not require training.
- We experimentally investigate the MFIP and MFSED tables' parameters and evaluate their biometric performance w.r.t. the influence of their integration with HE, achieving a performance better than the baseline.
- We improve the integration of MFBRs with HE presented in [1], rendering the encrypted reference template more compact, restricting it to one ciphertext instead of d ciphertexts.
- We evaluated the effect of our improved integration MFBRv2 on the speed and found that it eliminates the speed gap between cleartext and encrypted decision modes, making them equally fast and faster than MFBRv1.

TABLE I: Comparison of our improved MFBR solution vs. the related work.

Schemes	Decision mode	Computational Complexity			Space Complexity	
		HE Mult.	HE Rot.	HE Add	Reference	Probe
Baseline	Cleartext	2	d	d	1 ciphertext	1 ciphertext
	Encrypted	$2+\lceil \frac{l}{c} \rceil$	$d+\log_2(c)$	$d+\log_2(c)+\lceil \frac{l}{c} \rceil$		
[5]	Cleartext	1	$\log_2(c)$	$\log_2(c)$	1 ciphertext	1 ciphertext
MFBRv1 [1]	Cleartext	0	N	$d+1$	d ciphertexts	d-dim integer vector
	Encrypted	0	$N+\log_2(c)$	$d+\log_2(c)+2$		
MFBRv2 (this work)	Cleartext	0	$\log_2(c)$	$\log_2(c)$	1 ciphertext	d-dim integer vector
	Encrypted	0	$\log_2(c)$	$\log_2(c)+1$		

Feature vector's dimension (d), plaintext slots capacity of a ciphertext (c), the length of an MFBR table's row (N), and comparison vector's length (l), that is, the number of integers between the threshold and the max score.
The HELR classifier is excluded from this table because its integration with HE supporting SIMD results in a similar complexity, and, in [9], the presented integration is based on a partially homomorphic encryption scheme that does not support the SIMD property and encrypts each element separately.

II. Background: HELR Framework

The HELR classifier assumes that the features are independent and follow the Gaussian distribution. The features' independency allows treating each feature separately and thus calculating the LLR per feature. Therefore, in the following, we describe the HELR framework process for a given feature; the same applies to all features.

A. Generation of HELR lookup tables

For a single feature, the LLR is a two-input-one-output function. Pre-computing it yields a lookup table where the rows' (resp. columns') indexes represent the possible values of the first (resp. second) input, feature from the reference template (resp. probe), and the cells contain the output, individual scores. The rows' and columns' indexes result from a feature quantization that maps the continuous domain of real numbers to a finite set of integers, which is needed to limit the possible feature values. In HELR, the feature quantization on $N = 2^n$ *feature quantization levels* is performed by dividing the PDF of the zero-mean and unit variance Gaussian distribution $\mathcal{N}(0,1)$ in an equiprobable manner so that the lookup table's cells have an identical probability for an arbitrary feature observation. Assuming features with Gaussian distribution, they follow $\mathcal{N}(0,1)$ and the bins' borders are determined by following Algorithm 1

[1]Concretely, for $d=512$ and $N=8$, a reference template requires 4096 plaintext slots, and depending on the security parameters, $c\in\{4096,8192,16384\}$

where ICDF$(p,0,1)$ is the inverse of the CDF of a $\mathcal{N}(0,1)$ at the cumulative probability p and it returns the value associated with p. In Algorithm 2, $a_i \in \text{Bn}_{a_i}$ (resp. $b_i \in \text{Bn}_{b_i}$) denotes the measured value for feature from the first (resp. second) sample and is quantized to $\hat{a}_i$ (resp. $\hat{b}_i$) using the same Bn bins' borders array of the i-th feature.

Algorithm 1: Procedure to determine the bins' borders. Algorithm from [9].

Input: $N = 2^n$ feature quantization levels
Output: Bn array containing the bins' borders
Bn array of size $N-1$;
for $j \leftarrow 1$ **to** $N-1$ **do**
  $p = j/N$;
  Bn$[j]$ = ICDF$(p,0,1)$;
end for

B. Biometric Recognition based on HELR

Once the lookup tables are generated, the HELR has the following conventions for the reference template and the probe. The d-dimensional feature vector corresponding to the reference template is mapped to its integer representation according to the feature quantization procedure (Algorithm 1 and Algorithm 2). Thus, the HELR reference template becomes a vector of rows selected from the lookup table corresponding to a feature and its index is indicated by the quantized value of that feature. The same feature quantization is applied to the probe feature vector but a feature quantized value indicates the column's index of the row corresponding to that feature in the reference template. Hence, given the HELR reference template and an HELR probe, the biometric recognition reduces to the row-wise selection of the individual scores from the reference template based on the probe. Then, their addition produces a final score S, which is compared against a biometric threshold Θ. The case where S exceeds Θ is considered a *match*; otherwise, it is a *non-match*.

Algorithm 2: Feature quantization on $N = 2^n$ feature levels. Algorithm from [9].

Input: a_i raw feature value of the i-th feature
  and Bn array containing the bins' borders of the i-th feature
Output: $\hat{a}_i$ quantized value
for $j \leftarrow 1$ **to** $N-1$ **do**
  if $a_i < Bn[j]$ **then return** $j-1$;
end for
return $N-1$

III. Multiplication-Free Biometric Recognition

Our primary goal is to apply the HELR framework to common (dis-)similarity measures not requiring training to learn the features' statistics for determining a score, such as the cosine similarity and the squared Euclidean distance. To construct suitable lookup tables, we need to determine 1) the table's cell borders by equiprobably dividing the proper PDF that represents a random observation of the features, 2) the proper PDF and its CDF, and 3) the representative value of a cell. For this purpose, finding the proper PDF and its CDF that defines the feature vectors resulting from a DNN might be tricky and dependent on the DNN's training and architecture, which may lead to a different distribution per architecture. To avoid this, assuming that the feature vectors resulting from a DNN are points spread on the $\mathbb{R}^d$ space, we normalize them to bring them on the surface of the unit d-ball. Note that this normalization does not affect the (dis-)similarity measures on non-normalized feature vectors. A normalized vector of dimension d can be seen as a point on the unit d-ball. We assume that the points on the unit d-ball are uniformly distributed. This assumption is justified by the fact that a well-trained DNN (i.e., feature extractor) can achieve a (near-)optimal recognition performance only if the features are uniformly distributed over the d-ball. Thus, we derive the PDF of the projection of a point on the d-ball on an axis to determine the PDF corresponding to the coordinates of the normalized vector. Based on the inverse CDF of this PDF, we equiprobably quantize the observed projected point following Algorithm 1 and Algorithm 2 using the inverse CDF of the point projection on the unit d-ball instead of the ICDF$(p,0,1)$.

Remark III.1 ((Dis-)Similarity Measure Equivalent to the Inner Product)**.** *Note that the cosine of normalized vectors equals to their inner product. While the squared Euclidean distance of two normalized vectors is equivalent to their inner product via the monotonic function* $x \rightarrow 2(1-x)$.

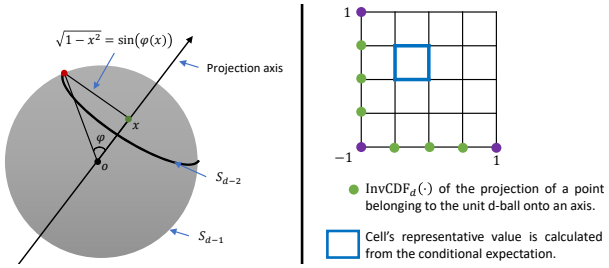


Fig. 1: Illustration of the projection of a point belonging to the surface area of the unit d-ball onto the x-axis (left) and an example of an MFBR table with $2^2 = 4$ feature quantization levels and feature vectors of dimension d (right).

A. Point Projection on the Unit d-ball

In this section, we describe how we derive the PDF and its CDF corresponding to the projection of a point on the unit d-ball. Note that this PDF is the PDF of every coordinate of a d-dimensional normalized feature vector.

1) PDF of the Point Projection on the Unit d-ball: Let X denote the random variable after projection on the x-axis. Let x denote a realization of that random variable. Let $\varphi(x)$ denote the angle of a point on the d-ball that is projected onto x. Let S_{d-1} denote the surface of a d-ball given by $\text{S}_{d-1}(r) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})} r^{d-1}$ where r denotes the radius and Γ denotes the Gamma function $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt$. The probability mass of the points on the d-ball is uniformly distributed over the d-ball. The probability mass that a point on the d-ball is projected onto the x-axis is the mass contained on the surface of the (d-1)-ball with center x and radius $\sqrt{1-x^2} = \sin(\varphi(x))$. In order to derive the PDF $f_X(x)$ we have $f_X(x) = \frac{d}{dx}\Pr\{X \leq x\}$. By looking at Figure 1 we can see that $\Pr\{X \leq x\} = \frac{1}{S_{d-1}(1)} \int_{\varphi(x)}^{\pi} S_{d-2}(\sin(t))\, dt$ where $\frac{1}{S_{d-1}(1)}$ is normalization factor. Hence, $f_X(x) = \frac{-1}{S_{d-1}(1)} S_{d-2}(\sqrt{1-x^2})\varphi'(x)$.

Note that $x=\cos(\varphi) \Longrightarrow \varphi'(x)=\frac{-1}{\sqrt{1-x^2}}$, thus we have

$$f_X(x)=\frac{1}{S_{d-1}(1)}S_{d-2}(\sqrt{1-x})\frac{1}{\sqrt{1-x^2}} = \frac{1}{B(\frac{1}{2},\frac{d-1}{2})}\left(\sqrt{1-x^2}\right)^{(d-3)}$$

where the Beta function: $B(x,y)=\int_0^1 t^{x-1}(1-t)^{y-1}dt$. Recall the relation between the Gamma function and the Beta function is given by $B(u,v)=\frac{\Gamma(u)\cdot\Gamma(v)}{\Gamma(u+v)}$. Therefore, the PDF of the projection of a point belonging to the unit d-ball onto an axis is given by

$$f_X(x)=C\cdot\left(\sqrt{1-x^2}\right)^{(d-3)} \tag{1}$$

where $x\in[-1,1]$ and the normalizing constant $C=\frac{1}{B(\frac{1}{2},\frac{d-1}{2})}$.

2) CDF of the Point Projection on the Unit d-ball: Let F_X denote the CDF corresponding to the PDF f_X. To calculate F_X, we need to solve the integral in Equation (2).

$$F_X(x)=\int_{-1}^{x} C\cdot\left(\sqrt{1-t^2}\right)^{(d-3)}dt \tag{2}$$

where $x\in[-1,1]$. Let us solve the following integral separately $\int\left(\sqrt{1-t^2}\right)^{(d-3)}dt$. Using the substitution $t=\sin(u)$, we have $dt=\cos(u)du$.

$$\int\left(\sqrt{1-t^2}\right)^{(d-3)}dt=\int(\cos(u))^{(d-2)}\,du \tag{3}$$

In Equation (3), the passage from the left-hand side to the right-hand side is justified by the fact that $t=\sin(u)$ implies $u=\arcsin(t)\in[\frac{-\pi}{2},\frac{\pi}{2}]$. Thus, $\cos(u)\in[0,1]$ when $u\in[\frac{-\pi}{2},\frac{\pi}{2}]$. Solving this integral depends on the parity of d. Since d in our case represents the size of a feature vector that is generally expressed as $2^{\#Bits}$, let us assume that d is even which implies that $d-2$ is also even. By using the trigonometric power formula for cosine raised to an even power [16], for $n=d-2$ we have:

$$\cos^n(u)=\frac{1}{2^n}\binom{n}{\frac{n}{2}}+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cos((n-2k)u) \tag{4}$$

Plugging in (4) in the integral (3), the integral becomes:

$$\int(\cos(u))^n\,du=\int\frac{1}{2^n}\binom{n}{\frac{n}{2}}+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cos((n-2k)\,u)\,du \tag{5}$$

$$=\frac{1}{2^n}\binom{n}{\frac{n}{2}}\,u+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\frac{\sin((n-2k)\,u)}{(n-2k)} \tag{6}$$

$$=\frac{1}{2^n}\binom{n}{\frac{n}{2}}\arcsin(t)+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\frac{\sin((n-2k)\arcsin(t))}{(n-2k)} \tag{7}$$

From (6) to (7), we go back to the original variable by replacing $u=\arcsin(t)$. The term $\sin((n-2k)\ \arcsin(t))$ in Equation (7) can be simplified using the multiple-angle formula in [16]. That is Equation (8)

$$\sin(rx)=\sum_{j\text{ odd}}^{r}(-1)^{\frac{j-1}{2}}\binom{r}{j}\cos^{r-j}(x)\sin^j(x) \tag{8}$$

Replacing $r=n-2k$, $x=\arcsin(t)$, and $\cos(\arcsin(t))=\sqrt{1-t^2}$, and the simplification of Equation (8) solve the integral in Equation (5). Therefore, the CDF of the projection of a point belonging to the unit d-ball onto an axis is given by

$$F_{X_i}(x)=C\cdot\left[\frac{1}{2^n}\binom{n}{\frac{n}{2}}\arcsin(t)+\frac{1}{2^{n-1}}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cdot\frac{1}{(n-2k)}\cdot\right. \tag{9}$$

$$\left.\left(\sum_{j\text{ odd}}^{n-2k}(-1)^{\frac{j-1}{2}}\binom{n-2k}{j}\left(\sqrt{1-t^2}\right)^{n-2k-j}\cdot t^j\right)\right]_{-1}^{x}$$

where $n=d-2$. For the inverse CDF, we use Brent's method implemented in SciPy [17] to compute the numerical inverse of the CDF at a specific point.

B. Construction of MFIP and MFSED Tables

Unlike the HELR framework, where there is a lookup table for each feature, the pre-computed inner product and the SED require the generation of a single table for all features because they follow the same PDF. Figure (7b) shows the theoretical prediction under the uniformity assumption (the solid line) covers the empirical data (the bar graph), which empirically proves that the points on the unit d-ball are uniformly distributed. Moreover, the classifier's performance can be improved as long as the training samples' feature vectors are not uniform. If the training set is representative of the test data, we may assume that the features are uniformly spread over the ball. We call MFIP the lookup table that pre-computes the inner product and MFSED the one that pre-computes the SED. To generate those tables, we first specify the borders of a cell by equiprobably cutting the x-axis and y-axis according to the PDF of the projection of a point belonging to the unit d-ball onto an axis. Then, we define a table of $N\times N$ cells, where $N=2^n$ denotes the feature quantization levels on n bits. Each cell's borders are determined according to the bins Bn for both axes x and y following Algorithm 1 and Algorithm 2; see Figure 1. Subsequently, we specify the cell's representative value by calculating the joint conditional expectation Equation (10) for independent X and Y. For the MFIP table, it is equal to Equation (10), and for the MFSED table, it is equal to Equation (11).

$$E_{X,Y}\left[x,y|\text{Bn}\right]=\frac{E_X\left[x|\text{Bn}_x\right]\cdot E_Y\left[y|\text{Bn}_y\right]}{P(\text{Bn})} \tag{10}$$

$$E_{X,Y}\left[x,y|\text{Bn}\right]=\frac{(E_X\left[x|\text{Bn}_x\right]-E_Y\left[y|\text{Bn}_y\right])^2}{P(\text{Bn})} \tag{11}$$

where Equation (12) defines the expectation over the cell's borders with respect to the x-axis Bn_x (similarly for the y-axis) and Equation (13) represents the probability of a cell. Note that the interval Bn_x has one of the three forms $\left[-1,\text{Bn}_x[1]\right)$, $\left[\text{Bn}_x[j],\text{Bn}_x[j+1]\right)$ or $\left[\text{Bn}_x[n-1],1\right]$.

$$E_X\left[x|\text{Bn}_x\right]=\int_{\text{Bn}_x}x f_X(x)\,dx \tag{12}$$

$$P(\text{Bn})=\frac{1}{N\times N} \tag{13}$$

Given that the cells' representative values are real-valued, we apply another quantization mapping, that we call *cell quantization*, to render them to integers, making them suitable for homomorphic

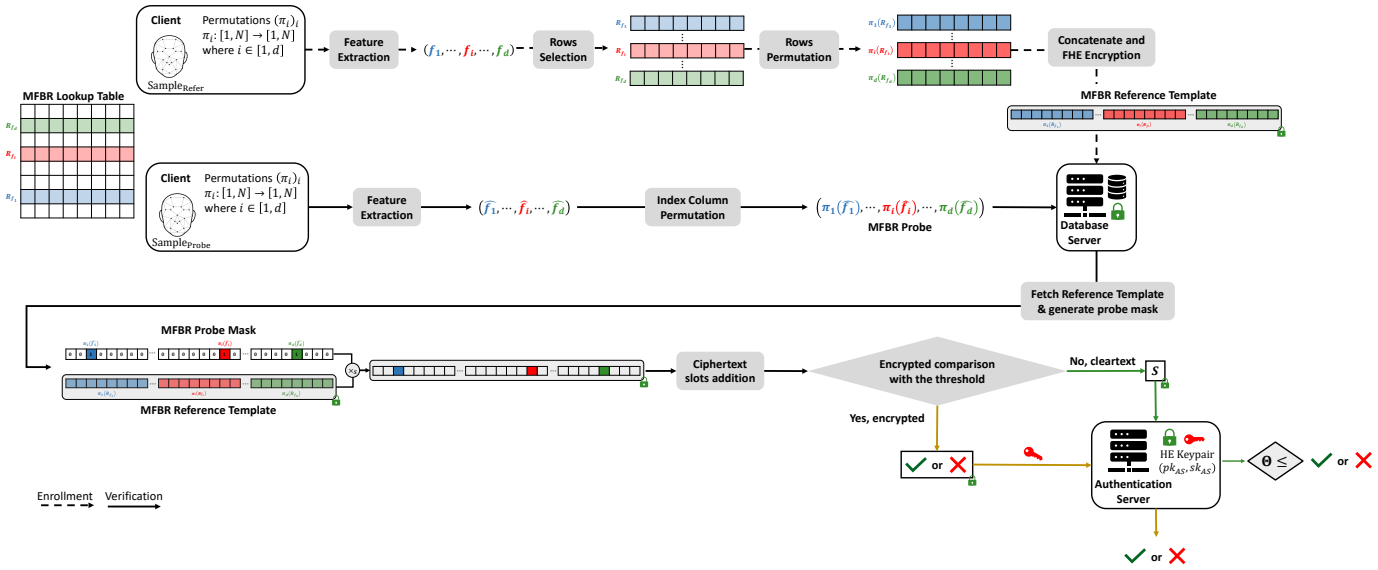


Fig. 2: Improved integration of MFBRv2 with HE supporting the SIMD and packing properties, where the encrypted reference template packs all the rows into one ciphertext, unlike MFBRv1 [1], where each row is packed in a separate ciphertext.

encryption (HE) schemes with an integer plaintext space. The cell quantization takes the cell's representative value, divides it by a quantization step Δ, and rounds it to the nearest integer.

IV. Improved Integration of MFBR with HE

Similarly to the HELR framework, described in Section II-B, MFBR-based biometric recognition follows the same reference template and probe convention. This convention facilitates the application of an HE layer over the recognition process when both the reference template and probe are protected. The HELR classifier, in [9], was implemented with an additively homomorphic encryption scheme (additive ElGamal encryption) where the encrypted reference template comprises the rows encrypted component-wise making the number of the ciphertexts, constituting the encrypted reference template, proportional to the sum of the rows' sizes, that is $\sum_{i\in[1,d]} N = d \cdot N$. The integration of MFBR with HE supporting the SIMD property presented in [1], that is MFBRv1, compresses this proportionality to the number of rows, such as each ciphertext encrypts one row, resulting in an encrypted reference template represented by a set of d ciphertexts. In [1], the encrypted reference template adopts this form to facilitate the selection of specific components under encryption by using homomorphic rotations; however, this comes at the cost of the encrypted reference template storage. In this paper, we further optimize the space complexity of the encrypted reference template by reducing it to one ciphertext that encrypts the packed concatenation of the rows, leveraging the SIMD property. As a result, the space complexity of the encrypted reference template becomes constant since the plaintext slot capacity c of a ciphertext is larger than $d \cdot N$, allowing for a full packing of the reference template. This storage optimization is also applicable to the HELR classifier when implemented with an FHE scheme supporting SIMD.

Figure 2 depicts our improved version of the integration of our MFBR lookup table with HE, that is MFBRv2, for both the cleartext and encrypted decision modes. For the encrypted decision mode, we use the same procedure described in Section V-A in [9], which is suitable for our MFBR lookup table regardless of its integration with FHE. The setting of Figure 2 is a semi-honest three-party protocol comprising a client, a database server, and an authentication server, assuming no collusion between both servers. The client is the biometric data owner possessing d permutations $\pi_i : [1,N] \to [1,N]$ that uses them to row-wise permute the reference template during the enrollment phase and locate the specific individual scores corresponding to a probe during the authentication phase. The database (DB) server is the holder of the protected reference templates encrypted under the authentication server's public key pk_{AS}. To select the individual scores under encryption, the DB server forms the mask corresponding to the permuted probe as a concatenation of vectors of the form $(\cdots,0,1,0,\cdots)$ where 1 is at the desired position to be selected. The encrypted reference template is multiplied, via one scalar multiplication, with this mask to yield a ciphertext of selected encrypted individual scores at their respective positions and zero elsewhere. To form the final score ciphertext, we sum over the plaintext slots of the selected individual scores ciphertext. We achieve this by adopting a similar technique used in [5], [18] where we clone the selected individual scores ciphertext as an intermediate ciphertext and rotate it by positions of 2^k where $k \in [0, \log_2(c) - 1]$ and add it to a temporary ciphertext to accumulate the plaintext slots along the process. This allows us to sum the selected individual scores and replicate the final score over all the plaintext slots of the final score ciphertext. This requires a constant number of additions and rotations, which is $\log_2(c)$ where c is the ring dimension. Unlike in [1], where the final score ciphertext requires the isolation of its first plaintext slot, in this work, the plaintext slot isolation is not needed as the resulting ciphertext comprises the final score replicated over all the plaintext slots after the summation. Once the final score ciphertext is formed, we distinguish between two decision modes: the cleartext decision and the encrypted decision. In the cleartext decision, the authentication server learns the recognition outcome by decrypting the encrypted final score sent by the DB server and comparing it with its biometric threshold to make a decision. In this case, the authentication server

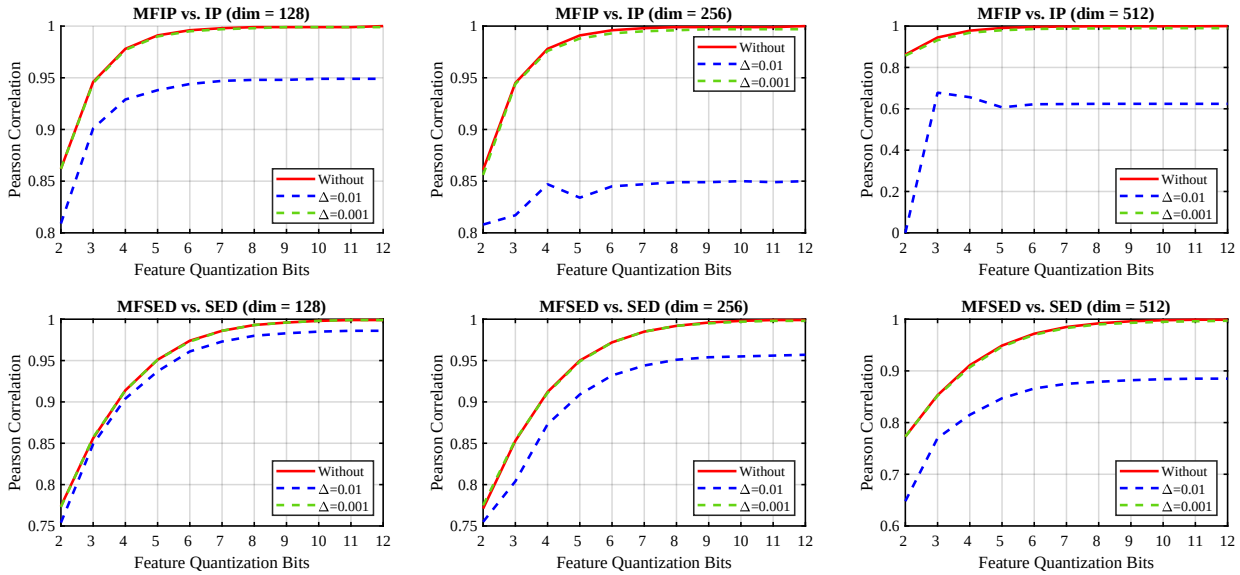


Fig. 3: Comparison of MFIP and MFSED against non-quantized and non-precomputed IP and SED expressed in terms of Pearson correlation coefficient for three different dimensions of feature vectors, IP vs. MFIP (first row) and SED vs. MFSED (second row). The x-axis indicates the number of bits (n) used in the feature quantization level. In the solid-line red curve, the lookup table is tested without rounding, while in the dashed-lines blue and green curves, the table is tested with two different rounding values $\Delta = 0.01$ and $\Delta = 0.001$, respectively.

has access to the cleartext scores, which is sometimes undesirable since it leaks more than one bit of information that enables the collection of scores for score-based attacks [15]. Hence, we limit this type of leakage to one bit of information by letting the comparison with the threshold be performed under encryption by the DB server following a procedure similar to [9] with an adaptation that leverages the SIMD property, as described in Remark IV.1. Thus, the DB server follows this adapted procedure to produce the permuted blinded comparison ciphertext $[\![\boldsymbol{aC}]\!]$ and sends it. Subsequently, the authentication server decrypts $[\![\boldsymbol{aC}]\!]$ and searches for zero in the vector $(\cdots,a_i\cdot(S-\theta_i),\cdots)$; if a zero is found, it decides for *a match*, otherwise *a non-match* since all the blinded values different from zero signifies that the final score is below the threshold.

Remark IV.1 (FHE Adaptation of the Comparison Procedure in [9])**.** *The comparison with the threshold under encryption procedure described in [9] needs an adaptation that leverages the SIMD property. Assuming that the final score ciphertext has the final score replicated over all its plaintext slots* $[\![\boldsymbol{S}]\!] = [\![(S,S,\cdots,S)]\!]$*, the DB server encrypts the permuted threshold vector as one ciphertext* [2] $[\![\boldsymbol{\theta}]\!] = [\![(\theta_1,\cdots,\theta_i,\cdots,\theta_l)]\!]$ *where* $\theta_i \in \pi([\Theta, S_{max}])$*,* $i \in [1,l]$ *and* $l = S_{\max} - \Theta + 1$*.*

Then, it subtracts $[\![\boldsymbol{\theta}]\!]$ *from the final score ciphertext* $[\![\boldsymbol{S}]\!]$ *to get* $[\![\boldsymbol{C}]\!] = [\![\boldsymbol{S}]\!] - [\![\boldsymbol{\Theta}]\!] = [\![(\cdots,S-\theta_i,\cdots)]\!]$*, and blinds the resulting ciphertext using a vector of random non-zero values* $\boldsymbol{a} = (a_1,\cdots,a_l)$*, where* $a_i \neq 0$*, to form a permuted blinded comparison ciphertext* $\boldsymbol{a} \times_s [\![\boldsymbol{C}]\!] = [\![\boldsymbol{a}\cdot\boldsymbol{C}]\!] = [\![(\cdots,a_i\cdot(S-\theta_i),\cdots)]\!]$*. Here,* a_i *is an element of the plaintext vector* $\boldsymbol{a}$ *while* $c_i = S - \theta_i$ *is an element of the encrypted vector* $\boldsymbol{C}$ *packed into the ciphertext* $[\![\boldsymbol{C}]\!]$*. To avoid expensive homomorphic multiplication, the blinding is performed through element-wise plaintext multiplication* [3] *of the random vector* $\boldsymbol{a}$ *by the permuted comparison ciphertext* $[\![\boldsymbol{C}]\!]$*.*

[2]Thanks to the lookup tables small score range, the threshold vector is packed into one ciphertext, opposite to IP baseline and [5] that require more ciphertexts for packing it.

[3]This is known as scalar multiplication.

V. Experiments and Evaluations

In this section, we experimentally study the MFIP and MFSED tables in terms of their parameter choices, their biometric performance on a facial dataset, the impact of their integration with HE on the recognition performance, and their runtime performance under encryption. We implement the experiments of Section V-A and Section V-B in Python 3.9 and the experiments of Section V-C in C++ using the OpenFHE [19] for the BFVrns homomorphic encryption scheme and OpenMP [20] for parallelization. We used a Linux Ubuntu 20.04.3 LTS machine run on a 64-bit computer Intel(R) Core i7-10750H CPU with 4 cores (8 logical processors) rated at 2.60 GHz and 16GB of memory. We make our source code publicly available [4].

A. Parameters Investigation

The MFIP (resp. MFSED) table is parametrized by a feature vector of dimension d, a feature quantization level 2^n, and a cell quantization step Δ, which we denote as MFIP(d,n,Δ) (resp. MFSED(d,n,Δ)). To understand the performance impact of these parameters, we study the MFIP and MFSED under different combinations of these parameters. We assess the quality of those tables in terms of the Pearson correlation coefficient by generating 200000 synthetic normalized feature vectors. We compare the inner product against the pre-computed inner product from the MFIP and the SED against the pre-computed SED from the MFSED. Figure 3 shows the MFIP and the MFSED tables' quality in terms of Pearson correlation coefficient. We notice that the larger the lookup table's size gets, the more accurate the table gets; the Pearson coefficient converges to 1. The score quantization step Δ has a faint impact on the table's

[4]Our implementation is available at https://github.com/aminabassit/improvedMFBR

accuracy, starting from values strictly smaller than 0.01, while it has a huge impact on the score range; the smaller Δ gets, the larger the score range becomes. The latter affects the runtime of the encrypted comparison with the threshold setting since it runs fast for smaller score ranges, as mentioned in [1]. The lookup table achieves an optimal accuracy starting from the size $2^3 \times 2^3$ without score quantization and with score quantization for $\Delta = 0.001$ for the three dimensions (128, 256 and 512). For $\Delta = 0.01$, the accuracy is maintained for feature vectors of low dimensions (128), while for high dimensions (256 and 512), the accuracy drops. This is justified by the information loss resulting from the majority of the cells rounded to zero.

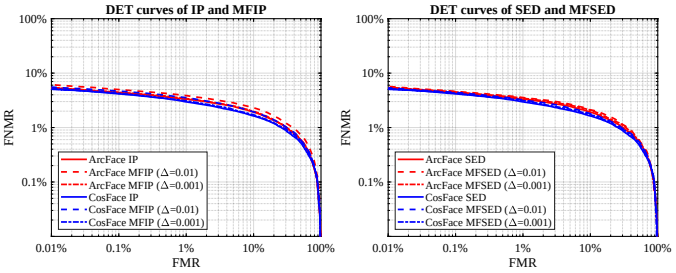


Fig. 4: Biometric performance of the baseline systems (IP and SED) and MFBR (MFIP and MFSED) over the VGGFace2 dataset with features extracted from ResNet-100-ArcFace and ResNet-100-CosFace.

B. Biometric Evaluation

As in [1], our MFBR approach supports any biometric modality that can be encoded as a fixed-length real-valued feature vector, assuming that the feature extractor is well-trained to yield features uniformly spread over the d-ball. To demonstrate this, we evaluate the biometric performance of the MFIP and MFSED tables on facial feature vectors. We used the VGGFace2 dataset [21] to extract facial feature vectors of dimension 512 using ResNet-100 [22] trained by two different losses: one trained with ArcFace [23] and another one trained with CosFace [24]. In the following experiments, we perform 52500 mated comparisons and 623750 non-mated comparisons.

The integration of MFBR tables with HE described in Section IV involves the use of pseudo-random permutations to generate the reference and probe templates. In the mated case[5], the permuted probe is equal to the non-permuted probe in the mated case because $\pi_i^{-1}(\pi_i(p_i)) = p_i$; consequently, the permutations preserve the mated score distribution. They differ in the non-mated case because the permutations are associated with different subjects, implying different permutations $\pi_i^{-1}(\pi_i'(p_i)) \neq p_i$, which benefits the non-mated score distribution by increasing the randomness of the features, resulting in incorrect selections. This is due to the fact that the permutation $\Pi = \{\pi_i\}_{i \in [1,d]}$ used to construct the reference template and the probe template should be the same so that both can link to the same cells with or without permutation. To analyze this permutation effect, in this section, we evaluate the biometric performance of the MFBR tables w.r.t. *fixed permutation*, where the same permutation is used for all templates, and *different permutations*, where each subject has a distinct permutation different from the others. In the case of different permutations, mated comparisons will be identical, whereas the non-mated comparisons will be affected by the permutations.

[5]In the mated case, the probe and reference are derived from the same subject, while in the non-mated case, they are derived from different subjects.

1) Fixed permutation: In this section, we fix the permutation used to generate the MFBR reference and probe templates to neutralize the permutation effect as if they are not involved in the generation of the MFBR-based reference and probe templates. Figure 4 compares the biometric performance of the baseline IP (resp. SED), as a non-pre-computed and non-quantized approach, against the MFIP (resp. MFSED), as a pre-computed quantized approach, for the following lookup table's parameters: table's size of $2^3 \times 2^3$, $\Delta = 0.01$ and $\Delta = 0.001$. The choice of the first optimal smallest lookup table size is justified by the aim for reference templates of small size since they are formed by a vector of rows belonging to the lookup table. The DETs on Figure 4 show similar performances, which are explained by the monotonic relationship between IP and SED (as discussed in Section III.1), implying an identical relationship between their corresponding pre-computed lookup tables MFIP and MFSED. Although the chosen lookup table's parameters ($2^3 \times 2^3$, $\Delta = 0.01$ and $\Delta = 0.001$) would be expected to yield less performant results compared to the other parameters evaluated in Figure 3, our evaluation results show that they preserve the biometric performance of the baseline non-pre-computed and non-quantized IP and SED measures. As for both score quantization steps, $\Delta = 0.01$ and $\Delta = 0.001$, the DETs are overlapping with the baseline's DETs with a slight performance loss noticed for $\Delta = 0.01$ at the gain of a smaller score range compared to $\Delta = 0.001$ that yields a larges score range. For features extracted by ResNet-100-ArcFace, the baseline IP and SED achieve an EER of 2.76% while their corresponding lookup tables for $\Delta = 0.01$ achieve an EER of 3.14% for MFIP and 3.01% for MFSED and for $\Delta = 0.001$ they achieve an EER of 2.82% for MFIP and 2.88% for MFSED. For features extracted by ResNet-100-CosFace, the baseline IP and SED achieve an EER of 2.47% while their corresponding lookup tables for $\Delta = 0.01$ achieve an EER of 2.85% for MFIP and 2.68% for MFSED and for $\Delta = 0.001$ they achieve an EER of 2.55% for MFIP and 2.53% for MFSED.

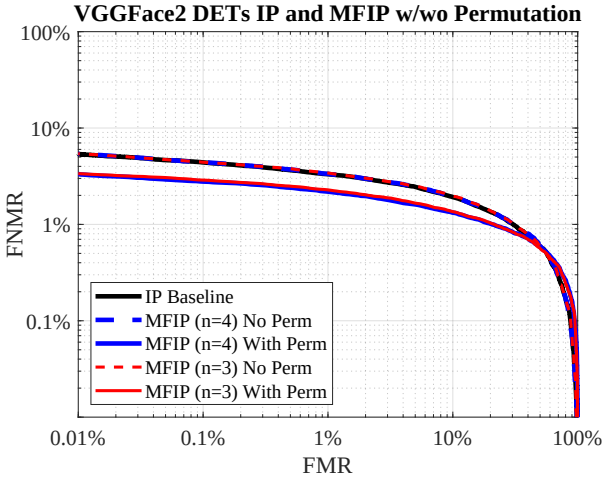


Fig. 5: Biometric performance of the IP baseline system and MFIP over the VGGFace2 dataset with fixed permutation and different permutations.

2) Different permutations: In this section, we vary the permutations used to generate the MFBR reference and probe template, analyzing their impact on recognition performance. We assume that each subject possesses a different permutation with which he generates his MFBR-based reference and probe templates. Since IP and SED are equivalent, we focus on the MFIP table

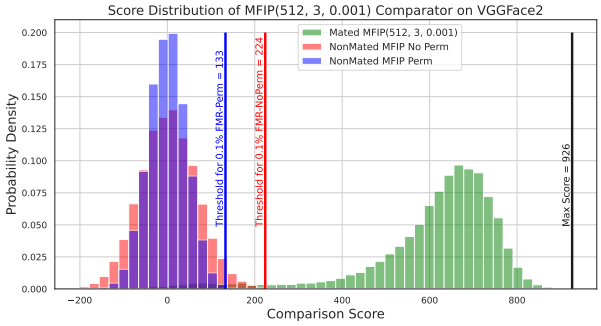


Fig. 6: Score distribution of MFIP($n=3$) with fixed permutation and different permutations over the VGGFace2 dataset.

and claim that the MFSED table will yield similar observations. Figure 5 depicts the DET curves corresponding to the MFIP lookup table with fixed and different permutations compared to the baseline IP that operates directly on the normalized vectors without pre-computation, quantization, and permutation. For both feature quantization levels ($n=3$ and $n=4$), we observe that the DETs corresponding to the MFIP with different permutations (the colored solid-line curves) remain consistently below the overlapping DETs that correspond to the MFIP with fixed permutation (the colored dashed-line curves) and the baseline IP DET (the black solid-line curve), showing the positive effect of the permutations on the recognition performance as it helps to outperform even the baseline IP. Hence, using different permutations per subject significantly lowers the false non-match rate compared to the baseline IP and the MFIP with fixed permutations for a similar false match rate. This positive effect of permutations is also shown in Table II where we measure the TMR@0.1%FMR and EER operating points for the VGGFace2 dataset. The improvement brought by the permutation effect over the baseline is countified by an increase of $\sim 1.5\%$ in TMR@0.1%FMR and a decrease of an $\sim 0.74\%$ increase in EER.

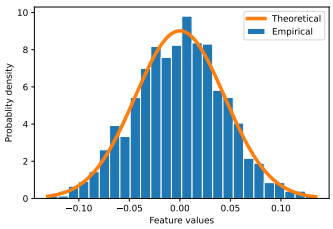
TABLE II: Performance of the MFIP comparator at the operating points.

Operating point	Baseline IP	MFIP($n=3$) Fixed Permut.	MFIP($n=3$) Diff. Permut.	MFIP($n=4$) Fixed Permut.	MFIP($n=4$) Diff. Permut.
TMR at 0.1% FMR	95.60%	95.58%	**97.12%**	95.58%	**97.22%**
EER	2.76%	2.82%	**2.02%**	2.79%	**1.97%**

To further investigate the permutation effect, in Figure 6, we depict the distribution of the mated and non-mated scores. The permutations do not affect the mated scores since they are calculated from probe-reference templates belonging to the same subjects. Thus, using permutations or not results in selecting identical scores since the same permutation is applied. Hence, the distribution of mated scores is depicted by the green histogram, a single histogram. Conversely, the non-mated scores experience a substantial impact from the permutations as they result from probe-reference templates derived from different subjects, each using a different permutation. Hence, the probe templates would select wrong positions non-corresponding to the order with which the reference template was permuted. In each graph, we plot the non-mated score distributions for the fixed permutation case (the red histogram) and the different permutation case (the blue histogram). We notice that the permutation impacts the non-mated score distribution by making it narrower than the non-mated score distribution for the fixed permutation case, causing a variation in the score threshold chosen at 0.1% FMR. This variation diminishes the amount of overlap between the mated and non-mated score distributions, which improves the overall performance observed in Figure 5.

x \ y	0	1	2	3	4	5	6	7
0	5	3	2	1	-1	-2	-3	-5
1	3	2	1	0	0	-1	-2	-3
2	2	1	0	0	0	0	-1	-2
3	1	0	0	0	0	0	0	-1
4	-1	0	0	0	0	0	0	1
5	-2	-1	0	0	0	0	1	2
6	-3	-2	-1	0	0	1	2	3
7	-5	-3	-2	-1	1	2	3	5

(a) MFIP$(512,3,0.001)$



(b) Theoretical vs. Empirical

Fig. 7: MFIP (Figure 7a) is the lookup table used in Section V-C. Figure 7b plots the histogram of a feature coming from a 512-dim normalized facial feature vector and its value w.r.t. its projection on the unit d-ball using our derived PDF with $d=512$.

C. Storage and Runtime Evaluation

In this section, we omit the runtime evaluation of the integration of MFSED with HE on the basis that its runtime is similar to the MFIP in [1] for both decision modes and focus on the runtime assessment of the MFIP. Hence, we consider the following HE-based BTPs: the IP baseline system[6], the integration of MFIP described in [1], and the improved integration of MFIP presented in Figure 2 in Section IV of this work. For both the MFIPv1 and MFIPv2, we use the best parameters obtained in Section V-B (see Figure 7a) and measure their runtimes using the BFVrns encryption scheme. Our HE-based BTP for the IP baseline system consists of a) homomorphically multiplying two packed normalized feature vectors that were quantized similarly to MFIP, b) rotating the resulted ciphertext to the i-th position where $i \in [1,512]$, c) adding the rotated ciphertexts, d) HE multiplying the resulted ciphertext by an encryption of $(1,0,\cdots,0)$ to isolate of the first plaintext, and then e) revealing the final score to compare it against the biometric threshold. Although our IP baseline system uses only two homomorphic multiplications, it has a runtime comparable to previous works [6], [7] when using the BFVrns scheme; it runs in 0.67s, 1.21s, and 1.24s for 128, 192, and 256 bits security levels, respectively.

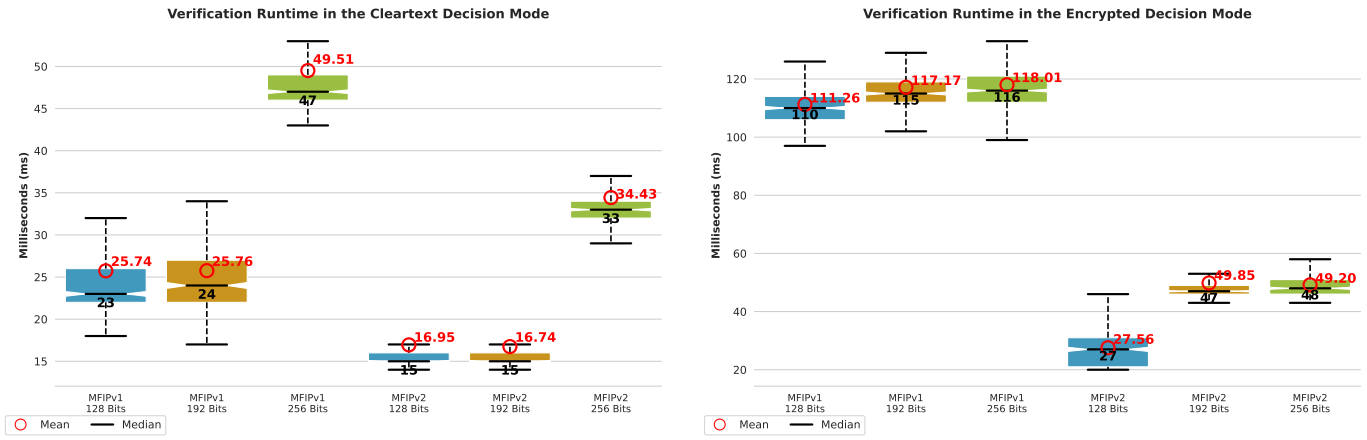
TABLE III: OpenFHE BFVrns parameters used in the HE-based BTPs: IP baseline, MFIPv1, and MFIPv2.

	MFIPv1 [1]			MFIPv2 (this work)			IP Baseline		
Security level (bits)	128	192	256	128	192	256	128	192	256
Plaintext modulus (p)	65537	65537	65537	65537	65537	65537	1146881	1146881	1146881
Error distribution (σ)	3.19	3.19	3.19	3.19	3.19	3.19	3.19	3.19	3.19
	Cleartext decision mode								
CRT moduli sizes (bits)	36	37	38	36	37	38	36	37	38
Ring dimension (c)	4096	4096	8192	4096	4096	8192	4096	8192	8192
	Encrypted decision mode								
CRT moduli sizes (bits)	39	39	39	36	37	38	37	37	38
Ring dimension (c)	8192	8192	8192	4096	8192	8192	8192	8192	16384

These parameters have a complex interdependency and cannot be freely chosen; otherwise, the security breaks. The CRT, which stands for the Chinese remainder theorem, optimizes the decryption and homomorphic multiplication in the Residue Number System (RNS) [25].

1) Probe-Reference template size: For both MFIP integrations with HE, the protected probe template consists of an integer vector of dimension 512, making it of size 2.1kB regardless of the security

[6]Our baseline system uses the same quantization as MFIP and comprises two HE multiplications: one for the element-wise multiplication of the coordinates and another one for the isolation of the first plaintext containing the IP.



(a) Cleartext decision mode

(b) Encrypted decision mode

Fig. 8: Verification runtime in the cleartext and encrypted decision modes of the MFIPv1 [1] and MFIPv2 (this work) measured over 500 comparisons using BFVrns implemented in OpenFHE v1.0.4 configured using the parameters in Table III.

level. The encrypted reference template size is mainly influenced by the implementation of the HE and its parameters, as in Table III, which vary w.r.t. the security level and decision mode. In general, we observe that the encrypted reference template size is identical for the 128 and 192 bits while it doubles for 256 bits. Table IV presents the storage improvement per mode and security level. For the IP baseline, in the cleartext (resp. encrypted) decision mode, the reference template and the probe template consist of one ciphertext each, making them of size 394.2kB (resp. 525.4kB), for the security levels 128 and 192 bits and of size 788.4kB (resp. 1MB) for 256 bits. For the cleartext (resp. encrypted) decision mode, the MFIPv1 reference template is represented by a set of 512 ciphertexts, making it of size 67.3MB (resp. 201.5MB) for the security levels 128 and 192 bits and 134.4MB (resp. 402.8MB) for 256 bits. For the improved MFIPv2, we reduce the encrypted reference template to one ciphertext with a size of 263kB for the security levels 128 and 192 bits and 525.1kB for the 256 bits in both decision modes. Therefore, our improved MFIPv2 significantly reduces the size of the encrypted reference template, making it approximately two orders of magnitude smaller than the MFIPv1 for all security levels.

2) Cleartext decision mode: Figure 8a compares the speed of the improved MFIPv2 against the MFIPv1 [1] in the cleartext decision mode by measuring their runtime using the BFVrns scheme configured as given in Table III over three different security levels (128, 192, and 256bits). In the cleartext decision mode, the runtime of the improved MFIPv2 is comparable to MFIPv1, with MFIPv2 being slightly faster than MFIPv1, a difference between 9ms to 15ms. The design difference between the improved MFIPv2 and MFIPv1 [1] resides in how the individual scores are selected and the final score is formed either by the summation of rotated ciphertexts and first plaintext isolation of the resulting ciphertext (as MFIPv1) or by the application of a binary mask encoding the to-be-selected scores through a scalar multiplication over the encrypted reference template and summation over its plaintext slots, which does not require first plaintext isolation (as MFIPv2). Hence, the design difference in the cleartext decision mode has a minor influence on the speed but reduces the encrypted reference template to one ciphertext, resulting in about 3 orders of magnitude less. Table IV demonstrates that MFIPv2 is about 50 times (resp. 94 and 44 times) faster than the baseline IP for a 128 bits security level (resp. 192 and 256 bits) for a more compact encrypted reference template, saving from 65.7kB to 262.3kB. The same Table shows that MFIPv2 is $\sim$2 times faster than [5] for a space gain as for the IP baseline.

3) Encrypted decision mode: Unlike the cleartext decision mode, where the final score is revealed, the encrypted decision mode reveals only one bit of information. To realize this, in Remark IV.1, we adapt the procedure described in Section V-A of [9] to FHEs supporting SIMD, which compares the encrypted final score against all possible scores between the threshold and the maximum score. We consider the score range depicted in Figure 6 by the threshold for the MFIP with different permutations (the blue vertical line) and the maximum score (the black vertical line). Hence, this range is small (794 integer scores) and contributes to the speedup of the verification in the encrypted decision mode. Figure 8b shows the runtime of the MFIPv1 and MFIPv2 in the encrypted decision mode. In this mode, the MFIPv2 outperforms the MFIPv1, making the runtime in both modes comparable. Unlike the MFIPv1, where the runtime in the encrypted decision mode is 2 to 4 times slower than the runtime in the cleartext decision mode. This improvement stems from how the final score ciphertext is formed after the selection of individual scores. In the MFIPv2, the final score ciphertext already contains the final score replicated over its plaintext slots. While in the MFIPv1, the replication of the final score over the plaintext slots requires the isolation of the first plaintext slot to be performed via the scalar multiplication of the binary mask $(1,0,\cdots,0)$ to ensure that the other plaintext slots are empty before replication [7]. As a result, Table IV demonstrates both the runtime and space improvements of MFIPv2 over the MFIPv1, the IP baseline, and [5], which we measured on the same machine using the same version of the OpenFHE library.

VI. Discussions

Comparison with other work. The IP-based HE-based BTP in [5] differs from our baseline by cleverly summing the plaintext slots under encryption, reducing both the number of multiplications

[7]The plaintext isolation trick in [1] does not work for the encrypted decision mode.

TABLE IV: Comparison between IP baseline system, MFIPv1 [1], MFIPv2 improved, and [5] for 512-dim feature vectors in the cleartext and encrypted decision modes.

HE-based BTP	Baseline			[5]			MFIPv1 [1]			MFIPv2 improved		
Quantization	$n=3$			Precision $=0.0025$			$n=3, \Delta=0.001$			$n=3, \Delta=0.001$		
Mated score range	$[5498, 9643]$			$[-33009, 158171]$			$[-201, 926]$			$[-201, 926]$		
#Ciphertexts in reference	1			1			512			1		
#Ciphertexts in probe	1			1			0			0		
Security level	128Bits	192Bits	256Bits	128Bits	192Bits	256Bits	128Bits	192Bits	256Bits	128Bits	192Bits	256Bits
Cleartext decision	822.11ms	1519.46ms	1583.87ms	31.22ms	52.89ms	52.62ms	25.74ms	25.76ms	49.51ms	**16.94**ms	**16.74**ms	**34.43**ms
Reference size	197.6kB	394.2kB	394.2kB	197.6kB	394.2kB	394.2kB	67.3MB	67.3MB	134.4MB	131.9kB	131.9kB	263kB
Probe size	197.6kB	394.2kB	394.2kB	197.6kB	394.2kB	394.2kB	2.1kB	2.1kB	2.1kB	2.1kB	2.1kB	2.1kB
Encrypted decision	2420.15ms	2373.95ms	5168.06ms	497.99ms	566.30ms	587.64ms	111.25ms	117.17ms	118.01ms	**27.55**ms	**49.84**ms	**49.2**ms
Reference size	525.4kB	525.4kB	1MB	197.6kB	394.2kB	394.2kB	201.5MB	201.5MB	201.5MB	197.6kB	394.2kB	394.2kB
Probe size	525.4kB	525.4kB	1MB	197.6kB	394.2kB	394.2kB	2.1kB	2.1kB	2.1kB	2.1kB	2.1kB	2.1kB

to one and the number of rotations to $\log_2(c)$, where c is the ring dimension. In [5], Algorithm 1 rotates the intermediate ciphertexts by positions of 2^k where $k \in [0, \log_2(c)-1]$ and accumulates them along the process. This allows the replication of the final score over all the plaintext slots of the final ciphertext, making the first plaintext slot isolation disposable and saving one multiplication over our baseline. For the cleartext decision mode, a multiplication-free approach and a single-multiplication approach are of comparable efficiency; with our improved MFIPv2 requiring less space compared to [5], between 131.9kB to 263kB instead of 197.6kB to 394kB. Table IV compares our solution with [5], which we re-implemented in the OpenFHE library based on its updated code[8]. The advantage of our approach over [5] is more visible in the encrypted decision mode. Using the procedure described in Remark IV.1, our solution is 10 to 18 times faster than [5]. This procedure compares the encrypted final score against all possible scores between the threshold and the maximum score. Thus, it requires a small score range to run efficiently, making it unsuitable for [5] whose feature quantization approach yields a large score range. If there exists an efficient comparison under encryption algorithm supporting large score ranges or independent from them, then [5] would be as fast as ours.

Security of our MFBR lookup table. Our lookup table is generated independently from a dataset and does not contain any personally identifiable information. Its cells are generated in an equiprobable manner so that they have an identical probability for an arbitrary feature observation, reinforcing the table's security. We assume that our MFBR lookup table is public knowledge. Because we encrypt the rows using a probabilistic encryption scheme, even encrypting the same row twice results in two completely different ciphertexts. As a result, the encrypted rows cannot be linked to the cleartext rows, whose index provides the quantized feature value. The client can make the server blindly select the specific columns without learning their real values by changing the probe's encoding to the encryption of a vector of ones in the to-be-selected coordinates and zeros elsewhere and multiplying the encrypted probe with the encrypted reference template. However, this introduces one homomorphic multiplication that we avoid by making the client apply a secret permutation of the rows before encrypting the reference template and sending the indexes' permutation for the selection. This enables a cheap blind selection on the server side since it transforms the protected permuted probe into a binary mask with which the server performs only a scalar multiplication.

[8]Updated code w.r.t the standardized security parameters is at [26]

VII. Conclusion

In this work, we demonstrated that the two common biometric comparison measures (cosine similarity and squared Euclidean distance) can be pre-computed and quantized without biometric accuracy loss. Upon our findings, we succeeded in freeing these comparison measures from homomorphic multiplications for a smooth application of an encryption layer. The results of our experiments show that our approach improves the biometric performance baseline when tested on facial features, the speed by a factor of 2 to 4 for space-efficient encrypted references, 2 to 3 orders of magnitude less. This makes our multiplication-free solution more compact, more accurate, and faster under encryption for both the cleartext and encrypted decision modes than its initial version and the baseline. Consequently, our improved integration enhances the storage of encrypted reference templates and effectively reduces the time difference between the two decision modes.

Acknowledgments

This work was supported by the PriMa project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860315.

References

[1] A. Bassit, F. Hahn, R. Veldhuis, and A. Peter, "Multiplication-free biometric recognition for faster processing under encryption," in *IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2022.

[2] A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome, and J. Fierrez, "Measuring the gender and ethnicity bias in deep models for face recognition," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2018.

[3] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[4] M. Sandhya and M. V. Prasad, "Biometric template protection: A systematic literature review of approaches and modalities," *Biometric Security and Privacy*, 2017.

[5] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–10.

[6] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf, and C. Busch, "On the application of homomorphic encryption to face identification," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2019.

[7] J. Kolberg, P. Drozdowski, M. Gomez-Barrero, C. Rathgeb, and C. Busch, "Efficiency analysis of post-quantum-secure face template protection schemes based on homomorphic encryption," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2020.
[8] T. Yang, Y. Zhang, J. Sun, and X. Wang, "Privacy enhanced cloud-based facial recognition," *Neural Processing Letters*, 2021.
[9] A. Bassit, F. Hahn, J. Peeters, T. Kevenaar, R. Veldhuis, and A. Peter, "Fast and accurate likelihood ratio-based biometric verification secure against malicious adversaries," *IEEE Transactions on Information Forensics and Security*, 2021.
[10] 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*. Springer, 2017.
[11] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Annual Cryptology Conference*. Springer, 2012.
[12] P. Mohanty, S. Sarkar, and R. Kasturi, "Privacy & security issues related to match scores," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE, 2006.
[13] P. Mohanty, S. Sarkar, and R. Kasturi, "From scores to face templates: A model-based approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
[14] P. Mohanty, S. Sarkar, and R. Kasturi, "Reconstruction of biometric image templates using match scores," 2012, US Patent 8,165,352.
[15] A. Bassit, F. Hahn, Z. Rezgui, U. Kelly, R. Veldhuis, and A. Peter, "Template Recovery Attack on Homomorphically Encrypted Biometric Recognition Systems with Unprotected Threshold Comparison," in *IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2023.
[16] I. Gradshteyn, I. Ryzhik, and R. H. Romer, "Tables of integrals, series, and products," 1988.
[17] "SciPy (release 1.9.0)," https://scipy.org/.
[18] C. Gentry, S. Halevi, and N. P. Smart, "Fully homomorphic encryption with polylog overhead," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2012, pp. 465–482.
[19] A. Al Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca, "OpenFHE: Open-Source Fully Homomorphic Encryption Library," in *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC'22. Association for Computing Machinery, 2022.
[20] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE computational science and engineering*, 1998.
[21] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019.
[24] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
[25] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," in *Cryptographers' Track at the RSA Conference*. Springer, 2019.
[26] V. N. Boddeti, "Source code of: Secure Face Matching Using Fully Homomorphic Encryption," https://github.com/human-analysis/secure-face-matching.

Amina Bassit received an MSc degree in cryptography from Mohammed V University, Morocco, and Limoges University, France, in 2015. From 2016 to 2019, she worked as a cryptographer at the DGSSI, Morocco. Since 2020, Amina has been pursuing her Ph.D. as part of the PriMa (Privacy Matters) project at the University of Twente, The Netherlands. Her research focuses on the integration of deep learning-based biometric recognition and homomorphic encryption to develop privacy-preserving biometric recognition solutions that enhance efficiency while maintaining recognition accuracy. Her research interests comprise privacy-preserving technologies applied to biometric recognition and the security of biometric-based protocols.

Florian F.W. Hahn studied computer science and obtained a Ph.D. degree from the Karlsruhe Institute of Technology (KIT) in Germany in 2018. From 2014 to 2018, he was a researcher at SAP Security Research in Karlsruhe, Germany. Since 2019, he is an Assistant Professor in the Semantics, Cyber Security and Services group at the University of Twente in the Netherlands. He is widely interested in security and privacy for data in its entire processing cycle. His research is currently particularly focused on cryptographic protection mechanisms for outsourcing scenarios and privacy-preserving machine learning.

Raymond N. J. Veldhuis (Senior Member, IEEE) graduated from the University of Twente, The Netherlands, in 1981. He received the Ph.D. degree from Nijmegen University in 1988 on a thesis entitled Adaptive Restoration of Lost Samples in Discrete-Time Signals and Digital Images. From 1982 to 1992, he was a Researcher with Philips Research Laboratories, Eindhoven, in various areas of digital signal processing. From 1992 to 2001, he was involved in the field of speech processing. He is currently a part-time Professor with the University of Twente, Enschede, The Netherlands, and NTNU, Gjøvik, Norway. His main research topic is machine learning for biometrics, with a focus on face recognition and biometric template protection. The research is both applied and fundamental.

Andreas Peter studied mathematics at the Universities of Oldenburg (DE) and Cambridge (UK) and received his doctorate in computer science from TU Darmstadt (DE) in 2013. He then worked at the University of Twente (NL), initially as a research assistant, then from 2014 to 2018 as an assistant professor and later as professor in applied cryptography. Since 2022, he is a professor in computer science at the University of Oldenburg (DE), where he leads the research group on Safety-Security-Interaction. He is also a visiting professor at the University of Twente (NL). His research includes both fundamental and applied security and privacy aspects in IT systems; he is currently particularly focusing on the application of cryptography in the field of privacy-preserving machine learning as well as the use of machine learning to improve security solutions.