Check for updates

IET Biometrics

The Institution of Engineering and Technology | WILEY

RESEARCH ARTICLE OPEN ACCESS

Towards Secure Biometric Solutions: A Multifactor Authentication Framework for Facial Recognition While Protecting User Data

José Silva[1] | Ajnas Muhammed[1] | Aniana Cruz[1] | Bruno Sousa[2] | Nuno Gonçalves[1]

[1]Institute of Systems and Robotics, University of Coimbra (ISR-UC), Coimbra, Portugal | [2]Centre for Informatics and Systems of the University of Coimbra (CISUC), University of Coimbra, Coimbra, Portugal

Correspondence: Ajnas Muhammed (ajnas.muhammed@isr.uc.pt)

Received: 2 February 2026 | **Revised:** 10 August 2026 | **Accepted:** 22 September 2026

Academic Editor: Imad Rida

Keywords: biometric templates | cryptographic algorithms | data privacy | facial recognition | multifactor authentication | template protection

ABSTRACT

Storing raw facial images or face embeddings in an authentication database creates a persistent privacy liability. A single breach can permanently expose a person's biometric identity. This paper presents a multifactor authentication (MFA) framework for facial recognition systems that removes this liability by never storing biometric data, in any form, on the server. Unlike conventional biometric template protection (BTP) techniques, which typically trade a fixed transformation of the image or the feature vector (e.g., PolyProtect, cancelable biometrics, and the fuzzy commitment scheme) for a reduction in recognition accuracy, the proposed method couples a decomposition-based feature-obfuscation step with time-based one-time passwords, AES-256 encryption, and SHA-512 hashing to generate a single-use, revocable authentication proof. This proof is designed to be irreversible, unlinkable, and revocable while leaving recognition accuracy effectively unchanged. Across six public benchmarks, namely, Labeled Faces in the Wild dataset (LFW), Age Database 30 (AgeDB-30), Cross-Age LFW (CALFW), Cross-Pose LFW (CPLFW), Racial Faces in the Wild (RFW), and Cross-Quality LFW (XQLFW), the proposed method attains 99.91% accuracy on LFW (area under the curve [AUC] 0.997) and is within 0.08–5.38 percentage above the published ArcFace and MagFace baselines on the benchmarks. A dedicated ablation with and without the protection layer, repeated under cosine, Euclidean, and Manhattan similarity, shows no measurable accuracy degradation attributable to the protection step itself. The principal novelty of the approach is architectural rather than purely cryptographic. The framework embeds feature-level BTP directly inside a MFA protocol, so that biometric-free storage, revocability, and MFA are obtained jointly instead of as separate, loosely coupled add-ons. Performance is comparatively weaker on cross-quality (XQLFW, 83.7%) and cross-age (CALFW) dataset, a limitation we analyze in this work.

1 | Introduction

Digitization has opened opportunities across many application domains, but it has also amplified concerns about the protection of personal data and individual privacy [1]. As security threats grow more sophisticated, organizations are adopting stronger authentication mechanisms, and the combination of multifactor authentication (MFA) with face recognition (FR) is becoming increasingly common. This trend raises pointed questions about how biometric data are protected, how security and privacy controls are implemented, and what trade-offs those implementations impose. High-profile vulnerabilities in widely used software, including Fortinet's FortiOS (CVE-2024-21762) [2], Jenkins (CVE-2024-23897) [3], and XZ Utils (CVE-2024-3094)

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Copyright © 2026 José Silva et al. *IET Biometrics* published by John Wiley & Sons Ltd.

[4], were disclosed in the first quarter of 2024 alone, each rated 9.8 or higher on the Common Vulnerability Scoring System (CVSS) [5]. These are general-purpose systems, not biometric ones, yet they illustrate the same underlying risk that motivates this paper: Once an attacker gains unauthorized access to a database, whatever it contains, including biometric data, becomes exposed.

FR systems (FRSs) are not immune to this kind of risk and in some respects are more exposed to it. Design and implementation flaws in FRS pipelines can be exploited in the same way as in other software [6], but FRS carries the additional burden of presentation attacks and spoofing, which target the biometric sensor itself rather than the surrounding software stack [7]. Malicious modification of the databases that back an FRS or of systems that depend on them can further compromise decision integrity [8]. This means that vulnerabilities in an FRS arise not only from its algorithms but also from its data management practices. Consequently, the choice of which biometric features to store, at what accuracy threshold, and under which procedural safeguards is not a peripheral engineering decision but a central one for reducing both recognition errors and privacy risk [9]. This paper addresses that choice directly, proposing an approach that treats accuracy and privacy as joint design constraints rather than as a trade-off to be resolved after the fact.

Biometric characteristics are physiological or behavioral traits that can be used to uniquely identify individuals, including facial features, fingerprint minutiae, retinal patterns, signatures, and typing patterns [10]. A biometric template (BT) is the digital representation produced by extracting and encoding such a characteristic [11]. Algorithms that protect a BT, collectively called BT protection (BTP) algorithms, are usually classified as image-level or feature-level. Image-level BTP transforms the original biometric image, for example, through distortion, noise addition, or encryption, before feature extraction [12]; feature-level BTP instead secures the already-extracted biometric features, for example, through hashing, homomorphic encryption (HE), or cancelable biometrics [13].

Image-level BTP techniques reported in the literature include distortion functions [14], morphing cancelation (morphed cancelable face images) [15], block scrambling [16], XOR operations, secret sharing [17], permutations, and filters, several of which can incorporate user-defined passwords to add revocability to an MFA system [18]. Co-occurrence matrices have also been used to extract distinctive characteristics in a way that prevents correlating the transformed image with the original [19, 20]. The majority of BTP methods, however, operate at the feature level and rely on cryptosystems or cancelable functions. Several use HE to encrypt the features and enable comparison directly in the encrypted domain [21–27], but, as with any encryption scheme, the features can be recovered if the key is known [13]. Other approaches derive a stable cryptographic object from the features, such as the fuzzy commitment scheme [28], which is built on hash functions and is discussed in more detail in Section 2.

Existing feature-level BTP schemes are typically evaluated as standalone template-security mechanisms. They protect a BT so that it is irreversible, revocable, and unlinkable, but they are not designed around, or evaluated within, a broader authentication protocol. Conversely, most MFA deployments that incorporate a biometric factor treat that factor as an unprotected input to a separate authentication flow, so any privacy guarantee for the biometric data has to be bolted on afterwards. The result is a gap between two bodies of work that address closely related problems in isolation: Template protection research optimizes irreversibility/unlinkability/revocability without an MFA context, while MFA research optimizes protocol security without addressing what happens to the BT itself. This gap becomes concrete when a template-protection method that performs well in a standalone benchmark is embedded into a real authentication pipeline. The key management, one-time-password rotation, and server-side trust assumptions all interact with the protection mechanism in ways a standalone template-security evaluation cannot capture. Recent deep-learning–based FRS backbones (AdaFace [29], ElasticFace [30], and TransFace [31]) continue to push recognition accuracy upward, but, to the best of our knowledge, none of them is coupled to a template-protection or MFA layer; accuracy and privacy are still being optimized as separate problems. Our motivation is therefore not simply "how do we protect a biometric template" but "how do we protect a biometric template *as part of* an end-to-end MFA protocol without paying for it in recognition accuracy."

The primary purpose of this work is to introduce an FRS-based MFA approach for everyday authentication that maximizes recognition accuracy while enhancing security and privacy. The design's defining property is that it never requires storing biometric data in raw or plaintext form, which removes the possibility of recognizing an individual outside the boundaries of the application that enrolled them. The methodology presented here is built to preserve, and where possible improve, recognition and authentication accuracy while incorporating user privacy and security as first-class requirements rather than afterthoughts. High authentication accuracy remains a precondition for security, since a system that is private but inaccurate is not usable, and the approach must also be scalable enough to accommodate the encryption-related data-protection regulations that organizations are increasingly required to follow.

Hence, in this paper, we propose a method that satisfies the three properties expected of a BTP scheme, namely, irreversibility, revocability, and unlinkability [32]. We adopt a feature-level approach so that we can exploit facial image discriminators built with modern neural networks, and we rely on standard, well-vetted cryptographic primitives, namely, SHA-512 hashing, AES encryption [33], a cryptographically secure pseudorandom number generator (PRNG), and TOTP, to protect the resulting features. The recognition decision itself is carried out with a convolutional neural network (CNN), combined with distance and similarity functions, and the overall protocol is an MFA scheme built from TOTP and CNN-based FR together. Hashing and encryption are then layered on top to produce secure objects that are safe to store in an application database. As the recognition decision is made in the domain of the CNN-extracted features rather than in a lossy derivative of them, accuracy is preserved.

Irreversibility, revocability, and unlinkability are achieved by generating cryptographic objects that are computationally hard to reverse, that can be regenerated at will (revocable), and that do

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

not, on their own, identify a specific individual across different registrations of the same face (unlinkable). We implemented this proposal in Python and evaluated it against established public benchmarks: the Labeled Faces in the Wild dataset (LFW) [34], Age Database 30 (AgeDB-30) [35], Cross-Age LFW (CALFW) [36], Cross-Pose LFW (CPLFW) [37], Racial Faces in the Wild (RFW) [38], and Cross-Quality LFW (XQLFW) [39].

The primary contributions of this paper are as follows:

- MFA-FRS integration. A protocol that combines MFA with facial recognition natively, rather than treating the biometric factor as an unprotected add-on to an otherwise conventional MFA flow.
- Secure, privacy-preserving registration. A registration procedure is built on cryptographic objects that jointly provide irreversibility, revocability, and unlinkability.
- Biometric-free databases. No raw biometric data, image, or reversible feature vector is ever written to the server storage.
- Accuracy preserved, not traded away. The technique preserves the accuracy without compromising privacy and security.

The remainder of the paper is organized as follows. Section 2 reviews related work and situates our contribution relative to it, including a comparative summary table. Section 3 describes the architecture, notation, registration and validation protocols, and threat model. Section 4 reports experimental results, an ablation of the protection layer, and a critical evaluation covering complexity, fairness, and ethical/regulatory considerations. Section 5 concludes and outlines future work.

2 | Related Works

FRS must satisfy several quality attributes simultaneously: accuracy, computational efficiency, security, privacy, and usability [40, 41]. This section focuses on methods that pursue BTP at the feature level using cryptographic systems or cancelable functions, since this is the class of method most directly comparable to the proposed technique, and feature-level protection has the strongest track record for preserving recognition accuracy. In general, BTP aims for accuracy, irreversibility, renewability, and unlinkability [40]; Rui and Yan [42] add mission success rate, the system's ability to withstand attacks while preserving confidentiality, as a further criterion, complementary to the other three.

2.1 | Data-Hiding and Steganographic Approaches

Li and Kot [43] conceal a user's identity inside their own fingerprint template using data hiding and embedding, storing the encrypted template for later authentication. A similar steganographic strategy in [44] fuses two fingerprints so that recovering either one individually is computationally hard and associates users with random, dynamically revocable identifiers to raise entropy. These schemes are effective at hiding the auxiliary identity data, but they still require storing an image-like object (the fused or steganographically modified template) in the database, which is precisely the exposure our design avoids.

2.2 | Deterministic Key-Generation Approaches

Dang and Choi [20] and Smith and Xu [45] describe deterministic key-generation from facial features: preprocessing, distortion, randomization into a stable template, and key extraction, in a pipeline that mirrors BTP feature extraction but tolerates intra-class variation. These methods guarantee zero key-generation uncertainty given the auxiliary data they store, but that auxiliary data is itself a dependency an attacker can target, and the schemes are not designed with an accompanying MFA protocol.

2.3 | Genetic/Evolutionary Cancelable Biometrics

El-Rahiem et al. [46] use crossover and mutation operators over permutation matrices to produce a cancelable biometric image, reporting a strong average area under the curve (AUC) of 0.9998. This is a capable image-level scheme, but it stores the cancelable *image* in the database, which conflicts with our requirement to avoid persisting any image-like biometric artifact, and it is incompatible with liveness checks that need direct access to F_1 (e.g., ICAO-style passport-photo validation, ISO 19794-5)[1].

2.4 | Feature-Level Cryptographic Schemes

The fuzzy commitment scheme [28] (Algorithm 1) links a codeword to the user at registration, stores the hash of the codeword together with the protected template, and at validation recomputes a candidate codeword whose hash must match. Subsequent work showed that the scheme can be inverted without knowledge of the codeword [13, 47, 48], undermining the irreversibility property it was designed to provide, though it remains an instructive baseline for the challenge-response structure we reuse in our own validation protocol. HE-based schemes [21–27] compare features directly in the encrypted domain and offer strong, provable confidentiality guarantees, but they inherit the general weakness of any encryption scheme: Possession of the key allows full recovery of the plaintext features, and HE's computational overhead is nontrivial at inference time. Kim et al.'s IronMask [49] is a more recent modular protection scheme for deep face templates that reports strong irreversibility and unlinkability

ALGORITHM 1 | Fuzzy commitment scheme.

Data: 1999

Result: Recognition system decision

protected BT = registration BT − codeword

$\overline{codeword}$ = validation BT − protected BT

if hash (*codeword*) = = hash ($\overline{codeword}$) **then**

The recognition is successful

else

The recognition is not successful

end if

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

with minimal accuracy loss; Shahreza et al.'s MLP-Hash [50] instead learns a machine-learning–based hashing function for the same purpose, and the same authors have separately shown that unprotected deep face templates are vulnerable to 3D reconstruction–based inversion [51], which is the concrete attack our irreversibility analysis in Section 3.4 is written to resist. None of IronMask, MLP-Hash, IronMask-like schemes, or the HE-based methods above is embedded in an MFA protocol; they protect the template and stop there.

2.5 | Recognition Backbones

A separate line of work improves the CNN backbone itself rather than the protection layer around it: Inception ResNet [52], ResNet-50/ArcFace [53], QualFace [54], the Idiap model [40], MagFace [55], AdaFace [29], ElasticFace [30], and, more recently, transformer-based recognizers such as TransFace [31]. We selected MagFace as the backbone for its consistently strong published accuracy at the time this work was carried out; we return to the question of benchmarking against these newer backbones, including transformer-based ones, as an explicit limitation in Section 4.2, since rerunning the full benchmark suite against every recent backbone is beyond the scope of this work.

Table 1 summarizes the comparison above against the three properties a BTP scheme is expected to provide, plus whether the scheme is coupled to an MFA protocol and whether it avoids storing any image-like or directly reversible object. The last two columns are where our proposal differs from all of the methods surveyed: to our knowledge, no prior scheme in this table combines native MFA coupling with a fully biometric-free database.

3 | System Design

This section describes the proposed FRS design that uses MFA to guarantee irreversibility, revocability, and unlinkability for each enrolled subject, with enhanced security and privacy. The design keeps the database free of raw biometric information, which makes CNN-driven FRS authentication both more secure and, as no biometric matching has to be rederived from stored images, computationally lean at query time. Security and privacy are improved both while data is in transit and while any of its derivatives sit inside the database by adding explicit validation steps around an otherwise conventional FRS pipeline.

Figure 1 illustrates the block diagram of the proposed method. Section 3.1 gives an overview of the two functional phases, the registration phase and the validation phase, and of the three entities involved, the client, the registration server, and the authentication server. Table 2 defines every symbol used from this point onward.

3.1 | Overview

The client, registration server, and authentication server are the three components of the proposed FRS (Figure 1), and the system moves through two phases, registration and validation.

Registration starts when a client sends a registration request, together with a face image F_1, to the registration server. The server checks F_1 against a security checklist derived from ICAO guidance; only if F_1 is judged a valid, coherent human face does registration proceed. Registration then generates a set of cryptographic objects, *username*, *otp_key*, *biometric_key*, *user_hash*, *user_key*, and the *proof* (defined in Table 2 and derived formally in Section 3.2).

The registration server hands the relevant objects to the authentication server, which will store them, and separately shares a subset with the client; *user_hash*, *user_key*, and *proof* remain server-side only. Once this exchange completes, the registration server deletes F_1 and every cryptographic object it generated, so it retains no persistent memory of the registration.

Cryptographic key generation (*opt_key*, *biometric_key*, and *user_key*) is the sole responsibility of the registration server, not of the client or the authentication server, which keeps key-generation policy (and the choice of secure primitives) centralized and auditable. Every one of these objects, including *user_key* and *proof*, is unique to a single registration and exists only server-side. During validation, the *proof* is the object that actually gets checked, via hashing, which is why the server never needs to persist biometric data: Exposure of

TABLE 1 | Comparative summary of representative biometric template protection approaches.

Method	Irreversible	Revocable	Unlinkable	MFA-coupled	Biometric-free DB
Fingerprint steganography [43, 44]	~	✓	✓	×	×
Deterministic key generation [20, 45]	✓	×	~	×	×
Genetic cancelable images [46]	~	✓	~	×	× (stores image)
Fuzzy commitment [28]	×[a]	✓	✓	×	✓
HE-based feature protection [21–23]	~ (key-dependent)	✓	✓	×	✓
IronMask [49]	✓	✓	✓	×	✓
MLP-Hash [50]	✓	✓	✓	×	✓
Proposed	✓	✓	✓	✓	✓

Note: ✓ = property satisfied by design, × = not satisfied or not addressed, and ~ = partially satisfied/claimed but not formally shown.
[a]Originally designed to be irreversible, subsequent cryptanalysis showed the template can be recovered without the codeword [47, 48]. All other verdicts reflect each scheme's own stated design goal, as cited in the surrounding text.

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

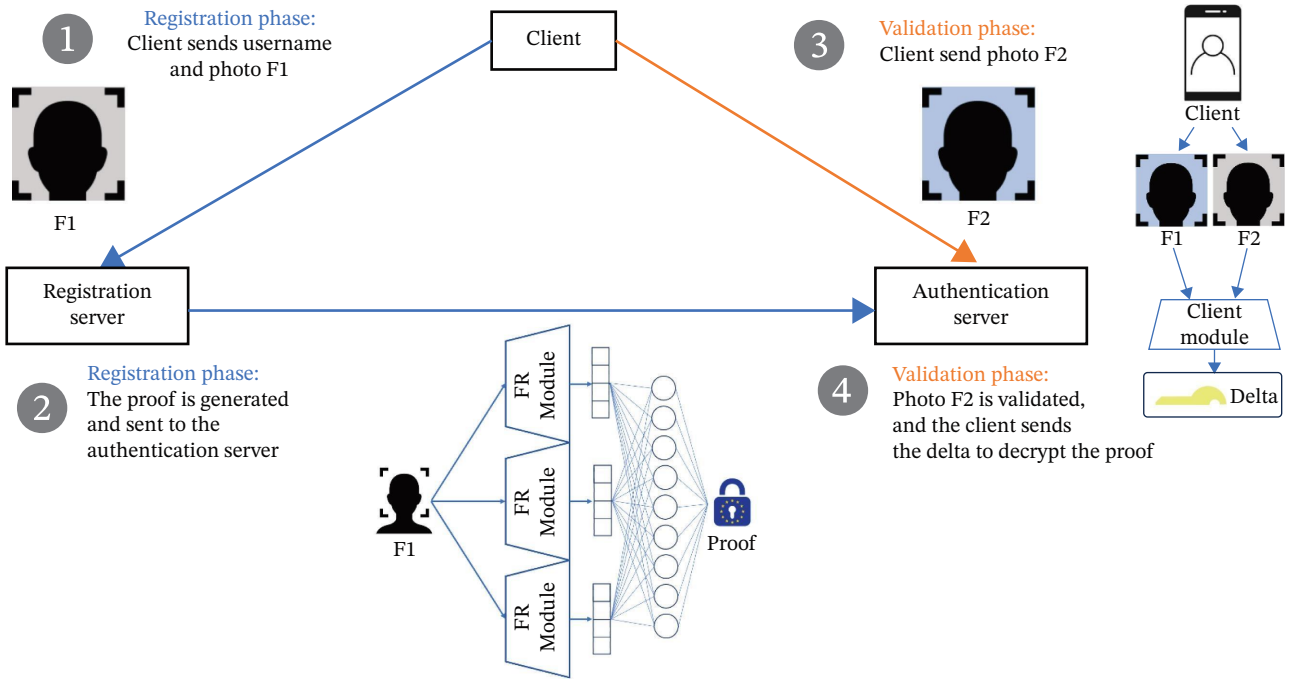


FIGURE 1 | Architecture of the proposed approach: an illustration of the registration (blue) and the validation phase (orange). The client, registration server, and authentication server exchange the cryptographic objects listed in Table 2; solid arrows denote messages sent over a secured channel (TLS/VPN), and no biometric image or reversible template ever crosses the registration server-authentication server boundary.

TABLE 2 | Notation used throughout Section 3.

Symbol	Phase	Meaning
F_1, F_2	Registration/validation	Registration and validation face images (frames) supplied by the client
BT_1, BT_2	Registration/validation	CNN-extracted feature embeddings (1024-d) of F_1 and F_2
biometric_key, *otp_key*, *user_key*	Registration	PRNG-generated secrets: seed the feature obfuscation, the TOTP generator, and the authentication proof, respectively
BTP_1, BTP_2	Registration/validation	Obfuscated (protected) versions of BT_1, BT_2, produced by our BTP function, and used as the AES key/IV source
user_hash	Registration	Hash(BTP_1); stored so the server can verify a proof without ever seeing BTP_1 in plaintext
proof	Registration	$E(BTP_1, user\_key + user\_hash)$; the single server-side object that stands in for the enrolled identity
delta	Validation	$\mathrm{dist}(BT_1, BT_2)$; lets the server recompute BT_1 from a fresh BT_2 without F_1 ever being sent again
$unique\_code_i$	Validation	TOTP output at step i, valid for 30 s
MAC	Validation	HMAC over *user_key* and a server-issued random nonce; the "challenge" the client must answer
P_1, P_2	Validation	MAC encrypted with, respectively, the current TOTP code and BTP_2; these are what actually cross the network
threshold	Validation	Similarity threshold above which BT_1 and BT_2 are declared a match

Note: Objects are grouped by the phase in which they are first produced.

the database does not, on its own, expose F_1. Because F_1 itself is discarded rather than transformed once and stored, it can be reused to mint an unlimited number of unrelated, unlinkable proofs. Table 3 lists which entity holds which object once registration completes.

A virtual private network (VPN) secures the channel between client and authentication server. To authenticate, the client submits a new face image F_2 together with its *username* over this channel. The authentication server first challenges the client, which must respond using its *opt_key*, producing Proof 1 (P_1).

ietbm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 | Attributes required in the registration phase.

Parameters/Entity ↓ →	Registration server	Authentication server	Client
F_1	×	×	✓
username	×	✓	✓
otp_key	×	✓	✓
biometric_key	×	✓	✓
user_hash	×	✓	×
user_key	×	✓	×
proof	×	✓	×

The client then extracts biometric features from F_2, applies its *biometric_key*, and produces Proof 2 (P_2), which the server uses to confirm that the client correctly derived BTP_2 from F_2. The client also computes a parameter we call *delta*, which lets the authentication server recompute the registered features without F_1 ever being resent (details are given in Section 3.3). The server then processes these recomputed features and decrypts the *proof* to confirm authenticity. If the templates derived from F_1 and F_2 are sufficiently similar, above a predetermined threshold, and every cryptographic check passes, authentication succeeds.

Where previous work using BTs typically relies on a single distance or similarity function, our protocol also validates similarity but folds that comparison into a chain of cryptographic checks specifically so that BTs are never stored or transmitted in a form an eavesdropper or database attacker could use directly. Table 4 lists the parameters required during validation.

Authentication is valid only if all four of the following hold:

1. Biometric match: The similarity between the registration features BT_1 and the validation features BT_2 exceeds *threshold*.
2. Proof of correct obfuscation: The client's P_2 shows that BTP_2 was correctly derived from BT_2 using the client's own *biometric_key*.
3. Consistent recomputation: The server uses *delta* to recompute BT_1, and from it BTP_1, since neither is ever stored server-side.
4. Valid proof decryption: The server uses the recomputed BTP_1 to decrypt and validate the stored *proof*.

Algorithms 2 and 3 give a stepwise summary of the two phases described in prose above and formalized with equations in Sections 3.2 and 3.3; each algorithm line is annotated with the equation it corresponds to, so that the mapping between the workflow and the mathematics reviewers asked us to clarify is explicit rather than implicit.

The registration and validation protocols are formalized in Sections 3.2 and 3.3, respectively, including the exact messages exchanged between entities.

TABLE 4 | Attributes required in the validation phase.

Parameters/Entity ↓ →	Authentication server	Client
username	✓	✓
otp_key	✓	✓
P_1	✓	✓
biometric_key	✓	✓
F_1	×	✓
BT_1	×	✓
BTP_1	×	✓
F_2	✓	✓
BT_2	×	✓
BTP_2	×	✓
delta	✓	✓
P_2	✓	✓
BT_2'	✓	×
BTP_2'	✓	×
BTP_1'	✓	×
proof	✓	×
user_hash	✓	×
user_key	✓	×
BT_1'	✓	×
threshold	✓	×

3.2 | Registration Phase

Registration begins when the client contacts the registration server, which is the entity both the client and the authentication server rely on: The registration server decides whether a submitted face image meets the required security criteria and produces the proofs that the authentication server will later trust. Because both other parties depend on its output, the registration server's procedures need to be transparent and auditable, a point we return to in Section 4.2.

To decompose biometric data into a fixed-size code, the *authentication proof*, the registration server draws a set of random parameters, *biometric_key*, *otp_key*, and *user_key*, from a PRNG (Algorithm 2, lines 3).

Registration proceeds as follows once the client submits F_1. Face detection and feature extraction are applied to the image, followed by a quality check against ICAO-style criteria: adequate lighting, absence of shadows, full visibility of the face, and an acceptable expression. A neural network validates the face and produces a feature vector; a CNN then reduces this to a 1024-float embedding BT_1. This embedding is the object we need to protect.

We protect BT_1 by decomposing and transforming it so that it is obfuscated, in a way that behaves like a hash function: Any change to BT_1 or to *biometric_key* produces a completely different output, but, unlike a conventional hash, the transformed data

ietbm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ALGORITHM 2 | Registration phase (executed by the registration server).

1: **Input:** face image F_1 from Client
2: Detect face in F_1 and check against ICAO quality checklist; abort if it fails
3: Generate $biometric\_key, otp\_key, user\_key \leftarrow \mathrm{PRNG}(\,)$
4: $BT_1 \leftarrow \mathrm{CNN}(F_1)$ ▷ Equation (2)
5: $BTP_1 \leftarrow \mathrm{BTP}(biometric\_key, BT_1)$ ▷ Equation (3), PolyProtect-style decomposition, Equation (1)
6: $user\_hash \leftarrow \mathrm{Hash\_function}(BTP_1)$ ▷ Equation (4)
7: $proof \leftarrow E(BTP_1,\ user\_key + user\_hash)$ ▷ Equation (5)
8: Send $\{username, otp\_key, biometric\_key, user\_hash, proof\}$ to Authentication Server
9: Send $\{username, otp\_key, biometric\_key\}$ to Client
10: Discard F_1 and every locally generated object

ALGORITHM 3 | Validation phase (client and authentication server).

1: **Input:** *username*, face image F_2 from Client
2: **Server:** look up $user\_key, otp\_key$ for *username*; compute $\mathrm{MAC} \leftarrow \mathrm{HMAC}(user\_key, \mathrm{PRNG}(\,))$ ▷ Equation (6)
3: **Server:** $unique\_code_i \leftarrow \mathrm{TOTP}(otp\_key)$; $P_1 \leftarrow E(unique\_code_i, \mathrm{MAC})$; send P_1 ▷ Equations (7), (8)
4: **Client:** recover $\mathrm{MAC} \leftarrow D(unique\_code_i, P_1)$ using its own $\mathrm{TOTP}(otp\_key)$ ▷ Equations (9), (10)
5: **Client:** $BT_1 \leftarrow \mathrm{CNN}(F_1)$, $BT_2 \leftarrow \mathrm{CNN}(F_2)$, $delta \leftarrow \mathrm{dist}(BT_1, BT_2)$ ▷ Equations (11)–(13)
6: **Client:** $BTP_2 \leftarrow \mathrm{BTP}(biometric\_key, BT_2)$; $P_2 \leftarrow E(BTP_2, \mathrm{MAC})$ ▷ Equations (14), (15)
7: **Client:** send $P_2, delta$ to server
8: **Server:** $\overline{BT_2} \leftarrow \mathrm{CNN}(F_2)$; $\overline{BTP_2} \leftarrow \mathrm{BTP}(biometric\_key, \overline{BT_2})$ ▷ Equations (16), (17)
9: **Server:** $\overline{\mathrm{MAC}} \leftarrow D(\overline{BTP_2}, P_2)$; **if** $\overline{\mathrm{MAC}} \neq \mathrm{MAC}$ **then abort** ▷ Equations (18), (19)
10: **Server:** $\overline{BT_1} \leftarrow \mathrm{dist}(\overline{BT_2}, delta)$; $\overline{BTP_1} \leftarrow \mathrm{BTP}(biometric\_key, \overline{BT_1})$ ▷ Equations (20), (21)
11: **Server:** decrypt *proof* with $\overline{BTP_1}$; **if** result $\neq user\_key + user\_hash$ **then abort** ▷ Equation (22)
12: **Server: accept** iff $\mathrm{similarity}(\overline{BT_1}, \overline{BT_2}) > threshold$ ▷ Equation (23)

remain usable for downstream facial recognition. Concretely, we use a modified version of PolyProtect [40] to derive an AES-256 key and initialization vector (IV) directly from BT_1 and *biometric_key*; this transformation increases entropy and produces a fixed code whose distribution differs from that of the original features, and it is deterministic, so identical inputs always yield identical outputs, a property we rely on to keep the recognition step in Section 3.3 accurate.

Our implementation of PolyProtect partitions the 1024-value embedding V into $m = 32$ nonoverlapping sets (matching the nonoverlapping configuration of the original PolyProtect algorithm [40]) and then maps each set using random coefficients c and exponents e derived from *biometric_key* via the PRNG. This produces 32 terms; Equation (1) shows the first two. From each resulting float, only the most significant nonzero digit is retained (e.g., 0.00005678 contributes the digit 5); if that digit is 0, the PRNG, seeded with *biometric_key*, supplies a random replacement digit instead. Concatenating the 32 retained digits yields the AES-256 key and IV.

$$T_1 = \sum_{i=1}^{m} c_i \cdot V_i^{e_i},\ T_2 = \sum_{i=m+1}^{2m} c_i \cdot V_i^{e_i}. \tag{1}$$

The registration server extracts BT_1 (Equation 2) and transforms it into the protected template BTP_1 (Equation 3) using our *BTP* function, initialized with *biometric_key* and built on the PolyProtect-style decomposition above.

$$BT_1 = \mathrm{CNN}(valid\_frame), \tag{2}$$

$$BTP_1 = \mathrm{BTP}(biometric\_key, BT_1). \tag{3}$$

Hashing BTP_1 produces *user_hash* (Equation 4). The *proof* is then the encryption of *user_key* concatenated with *user_hash*, using BTP_1 itself as the encryption key (Equation 5).

$$user\_hash = \mathrm{Hash\_function}(BTP_1), \tag{4}$$

$$proof = E(BTP_1, user\_key + user\_hash). \tag{5}$$

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The authentication server stores *username*, *otp_key*, *biometric_key*, *user_hash*, and *proof* but never learns the biometric features themselves, so it cannot decrypt *proof* on its own; decryption is only possible once the validation phase reconstructs BTP_1 from a fresh, live face image. Once registration succeeds, the client retains *username*, *otp_key*, and *biometric_key*, and the registration server discards every parameter it generated.

3.3 | Validation Phase

Validation verifies the identity of an already-registered client. The client submits its *username* together with a new face image F_2, distinct from the enrollment image F_1, over a secure channel. The authentication server retrieves the corresponding *user_key* and *otp_key* and generates a MAC via HMAC over *user_key* and a PRNG-generated nonce (Equation 6).

$$\mathrm{MAC} = \mathrm{HMAC}(user\_key,\ \mathrm{PRNG}(\,)). \tag{6}$$

The server encrypts this MAC using a disposable TOTP code seeded with the client's *otp_key* (Equation 7), producing P_1 (Equation 8). TOTP codes rotate every 30 s, which bounds the window during which a captured P_1 remains useful to an attacker.

$$unique\_code_i = \mathrm{TOTP}(otp\_key), \tag{7}$$

$$P_1 = \mathrm{E}(unique\_code_i, \mathrm{MAC}). \tag{8}$$

The client decrypts P_1 using the same TOTP code, independently derived from its own *otp_key* (Equations 9, 10), then runs the CNN on both F_1 and F_2 to obtain BT_1 (Equation 11) and BT_2 (Equation 12). It computes *delta* as the distance between the two embeddings (Equation 13), applies *BTP* to BT_2 to obtain BTP_2 (Equation 14), and uses BTP_2 to encrypt the MAC into P_2 (Equation 15). P_2 and *delta* are then sent to the authentication server.

$$\mathrm{MAC} = \mathrm{D}(unique\_code_i,\ P_1), \tag{9}$$

$$unique\_code_i = \mathrm{TOTP}(otp\_key), \tag{10}$$

$$BT_1 = \mathrm{CNN}(F_1), \tag{11}$$

$$BT_2 = \mathrm{CNN}(F_2), \tag{12}$$

$$delta = \mathrm{dist}(BT_1, BT_2), \tag{13}$$

$$BTP_2 = \mathrm{BTP}(biometric\_key, BT_2), \tag{14}$$

$$P_2 = \mathrm{E}(BTP_2,\ \mathrm{MAC}). \tag{15}$$

The authentication server independently computes $\overline{BT_2}$ from F_2 (Equation 16) and $\overline{BTP_2}$ from $\overline{BT_2}$ (Equation 17) and then decrypts P_2 with $\overline{BTP_2}$ (Equation 18) and checks the result against the MAC it generated earlier (Equation 19); a mismatch aborts the protocol immediately, since there is no reason to continue if the client cannot demonstrate correct use of its *biometric_key*. If the MAC matches, the server applies *delta* to $\overline{BT_2}$ to recover $\overline{BT_1}$ (Equation 20), obfuscates it into $\overline{BTP_1}$ (Equation 21), and uses $\overline{BTP_1}$ to decrypt the stored *proof* (Equation 22). Authentication is granted only once this decryption yields the expected *user_key* + *user_hash* and the similarity between $\overline{BT_1}$ and $\overline{BT_2}$ exceeds *threshold* (Equation 23).

$$\overline{BT_2} = \mathrm{CNN}(F_2), \tag{16}$$

$$\overline{BTP_2} = \mathrm{BTP}\big(biometric\_key, \overline{BT_2}\big), \tag{17}$$

$$\overline{MAC} = \mathrm{D}\big(\overline{BTP_2},\ P_2\big), \tag{18}$$

$$\mathrm{MAC} \Longleftrightarrow \overline{MAC}, \tag{19}$$

$$\overline{BT_1} = \mathrm{dist}\big(\overline{BT_2}, delta\big), \tag{20}$$

$$\overline{BTP_1} = \mathrm{BTP}\big(biometric\_key,\ \overline{BT_1}\big), \tag{21}$$

$$\overline{user\_key} + \overline{user\_hash} = \mathrm{D}\big(\overline{BTP_1},\ proof\big), \tag{22}$$

$$threshold < \mathrm{similarity}\big(\overline{BT_1},\ \overline{BT_2}\big). \tag{23}$$

Neither $\overline{BT_1}$ nor $\overline{BT_2}$ is ever written to the server database; both exist only transiently, for the duration of a single validation. Because the same F_1 can be recombined with a freshly generated key set, revocability is essentially free: Generating a new, unlinkable proof for the same enrolled face requires no re-enrollment, only a new key. The protocol is agnostic to the specific hashing and encryption primitives used, provided they meet standard security notions; our choices are described in Section 3.5.

3.4 | Threat Model and Security Analysis

This section makes explicit the adversary model implicit in Sections 3.2 and 3.3 and walks through, property by property, why the design satisfies irreversibility, unlinkability, and revocability, and where it remains exposed. We do not offer a formal cryptographic proof of these properties, which would require a reduction to the security of AES-256, SHA-512, and the PRNG under standard assumptions, which is outside the scope of this paper, but we give a structured, falsifiable argument for each property, in place of the purely qualitative treatment of the earlier draft of this manuscript.

3.4.1 | Adversary Capabilities

We consider three adversary classes: (A1) a passive network eavesdropper without database access, (A2) an adversary with full read access to the authentication server's database (a breach scenario), and (A3) an adversary who additionally compromises the client device or otherwise obtains F_1, *otp_key*, and *biometric_key*. We assume that TLS/VPN is correctly deployed on the client-authentication server channel (Section 3.1), consistent with the rest of the protocol description.

3.4.2 | Irreversibility

Against A1, the channel is encrypted, so eavesdropping alone yields nothing. Against A2, the database contains *user_hash* and *proof*, both derived from BTP_1, never BT_1 or BTP_1 in plaintext; recovering BT_1 from *user_hash* requires inverting SHA-512, and recovering it from *proof* requires either the AES-256 key (which is BTP_1 itself, i.e., circular) or breaking AES-256. This is

ietbm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

the same property that Otroshi Shahreza and Marcel show fails for *unprotected* deep face templates under 3D reconstruction attacks [51]; our design is specifically constructed so that no equivalent reconstruction target is ever stored. A2 therefore cannot recover BT_1 without also breaking a standard primitive, which is the strongest claim we can make without a formal reduction.

3.4.3 | Unlinkability

Two registrations of the same face, using independent *biometric_key* draws, produce statistically unrelated BTP_1, *user_hash*, and *proof* values, since *biometric_key* seeds both the polynomial coefficients/exponents in Equation (1) and the digit-selection PRNG. An adversary holding two such records (A2) cannot determine they originated from the same underlying face without also recovering *biometric_key* for at least one of them, which reduces to the irreversibility argument above.

3.4.4 | Revocability

Revoking a compromised proof requires only a fresh *biometric_key* (and, if desired, fresh *otp_key*/*user_key*) and the original enrollment image F_1, still held only by the client (Table 4); no re-registration with a new document or biometric capture is required, which is a practical advantage over schemes that treat revocation as re-enrollment.

3.4.5 | Known Attack Scenarios

3.4.5.1 | Spoofing/Presentation Attacks. The protocol validates ICAO-style image quality at registration but, as acknowledged already in Section 4.2, performs no active liveness check on F_2 at validation time, so a high-quality static image or print of a legitimate user's face is not distinguishable from a live capture by this protocol alone; this is why we flag liveness detection as the highest-priority future addition rather than an optional one.

3.4.5.2 | Replay Attacks. TOTP's 30-s validity window bounds how long a captured P_1 can be replayed, and the MAC nonce is freshly drawn per session, so replaying an old P_2 against a new challenge fails the MAC check in Equation (19).

3.4.5.3 | Template Inversion. This is addressed above under irreversibility.

3.4.5.4 | Man-in-the-Middle. Mitigated by TLS/VPN on the client-authentication server channel; without it, an on-path attacker able to intercept and relay F_2 in real time, that is, a live relay rather than a stored-template attack, remains a residual risk common to essentially all challenge-response biometric protocols, ours included.

3.4.5.5 | Adversarial Examples Against the CNN. Adversarial examples are out of scope for this paper, since they target the recognition backbone (MagFace) rather than the protection layer; a backbone-level defense (e.g., adversarial training) would compose with our protocol unchanged, since *BTP* operates downstream of whatever embedding the CNN produces.

3.4.6 | Comparison With HE/Secure-Enclave Approaches

HE-based schemes (Section 2) offer comparison directly in the encrypted domain, which sidesteps the need to ever decrypt features server-side, at the cost of substantial per-comparison computational overhead and, as noted above, full recoverability if the HE key is exposed. Secure enclaves (e.g., Intel SGX-style trusted execution) move the comparison into hardware-isolated memory, which protects against a software-only database breach (A2) but not against a compromised or malicious enclave operator and introduces a hardware trust assumption our design avoids entirely. We see these as complementary, not competing, directions: An enclave could host our authentication server logic to further reduce the operator's own attack surface, which we note as a natural extension in Section 5.

3.5 | Implementation Details

We implemented the proposed approach in Python and evaluated it on a server running Ubuntu 20.04 with an AMD Ryzen 7 5700G processor and 58 GB of RAM. Random parameters are generated with Python's *SystemRandom*, a cryptographically secure PRNG. Symmetric encryption uses AES-256 from the *cryptography* library; hashing uses SHA-512 from *hashlib*. The BTP function itself (Sections 3.2 and 3.3) is our own implementation, built on the strategy described in [40].

To make the pipeline reproducible, we summarize the configuration choices left implicit in earlier sections: The embedding size is fixed at 1024 floats (MagFace's native output); PolyProtect decomposition uses $m = 32$ nonoverlapping sets, so 32 coefficient/exponent pairs are drawn per key; TOTP codes use the standard 30-s validity window; and the similarity *threshold* used for the accept/reject decision in Equation (23) is the empirically optimal threshold reported per dataset in Tables 5, 6, and 7 (obtained by maximizing Youden's J statistic on the ROC curve for that dataset and similarity metric). Face images are cropped and aligned to 112×112 pixels following the ArcFace alignment protocol [53] before being passed to the CNN.

We emphasize that this is an *evaluation*, not a *training*, study: MagFace is used as a frozen, pretrained backbone (checkpoint released by the MagFace authors, trained on MS1M-V2 [53]), and no fine-tuning is performed. Consequently, there is no train/validation/test split to report for the CNN itself; the six benchmarks (LFW, AgeDB-30, CALFW, CPLFW, RFW, and XQLFW) are used exactly as their respective standard evaluation protocols specify, each providing 6000 predefined genuine/impostor pairs, and every pair is used for evaluation only. Preprocessing beyond alignment is limited to discarding images that fail the same ICAO-style quality checklist enforced at registration (Section 3.2); we did not otherwise filter or rebalance the benchmark pairs, so as to keep our numbers directly comparable to the published ArcFace/MagFace protocol.

Distance and similarity are evaluated in this work using cosine, Euclidean (L2), and Manhattan (L1) metrics, since the protection step in Equations (1)–(3) changes the geometry of the embedding

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 | Comparison of baseline and proposed method with respect to optimal threshold, sensitivity, and 1-specificity, cosine similarity.

	Baseline			BTP		
Datasets	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**
AGEDB30	0.29	96.73	1.60	0.29	96.73	1.60
CALFW	0.39	91.27	0.83	0.39	91.29	0.83
LFW	0.40	99.70	0.00	0.40	99.70	0.00
RFW African	0.50	95.40	2.97	0.50	95.40	2.94
RFW Asian	0.49	94.17	3.07	0.49	94.16	3.07
RFW Caucasian	0.42	97.40	0.97	0.42	97.40	0.97
RFW Indian	0.50	94.77	2.57	0.50	95.73	3.53

Note: The near-identical baseline/BTP columns are the component-level ablation referenced in Section 4.1.

TABLE 6 | Comparison of baseline and proposed method with respect to optimal threshold, sensitivity, and 1-specificity, Euclidean distance.

	Baseline			BTP		
Datasets	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**
AGEDB30	1.19	96.73	1.60	1.19	96.73	1.60
CALFW	1.10	91.27	0.83	1.10	91.26	0.83
LFW	1.10	99.70	0.00	1.10	99.70	0.00
RFW African	1.00	95.40	3.03	1.00	95.40	3.01
RFW Asian	1.01	94.17	3.07	1.01	94.16	3.07
RFW Caucasian	1.07	97.40	0.97	1.07	97.40	0.97
RFW Indian	1.00	94.77	2.57	1.00	95.73	3.54

TABLE 7 | Comparison of baseline and proposed method with respect to optimal threshold, sensitivity, and 1-specificity, Manhattan distance.

	Baseline			BTP		
Datasets	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**	**Threshold**	**Sensitivity (%)**	**1-Specificity (%)**
AGEDB30	30.21	96.60	1.57	30.21	96.60	1.57
CALFW	28.12	91.20	1.03	28.12	91.20	1.03
LFW	28.02	99.70	0.00	28.40	99.70	0.00
RFW African	25.19	95.53	3.27	25.19	94.77	2.50
RFW Asian	25.65	94.20	3.23	25.65	94.23	3.24
RFW Caucasian	27.52	97.57	1.50	25.42	97.57	1.50
RFW Indian	25.38	94.50	2.27	25.38	94.50	2.27

space and we wanted to confirm that this did not silently favor one metric's assumptions over another's. Section 4 reports the effect this has on recognition accuracy.

4 | Experimental Results and Analysis

This section reports the evaluation plan (Section 4.1), presents and analyzes results including an accuracy/equal error rate (EER) discussion and a component-level ablation (Section 4.1), critically evaluates the design (Section 4.2), and compares similarity metrics (Section 4.4).

4.1 | Experiment Details and Results

4.1.1 | Datasets

We selected datasets based on availability and on a preference for lower label/image noise, since the backbone was trained on MS1M-V2 [53], which is known to be cleaner than alternatives such as MS-Celeb-1M [56]. We use six standard benchmarks: LFW [34], AgeDB-30 [35], CALFW [36], CPLFW [37], RFW [38], and XQLFW [39]. Because these are collected in the wild, some frames are of low quality, blurred, pixelated, or show closed eyes, which is a property of the benchmarks rather than an artifact of our pipeline but one that we return to below when

ietbm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

explaining the weaker results on XQLFW and CALFW. Images are 112×112 pixels, aligned per the ArcFace protocol [53], and each dataset provides 6000 evaluation pairs, used exactly as released (Section 3.5).

4.1.2 | Network Details

We use MagFace as the CNN backbone because it reported state-of-the-art accuracy among openly available checkpoints at the time of this study (Section 2 lists more recent alternatives, including AdaFace [29], ElasticFace [30], and the transformer-based TransFace [31], which we did not have the opportunity to rebenchmark against; see Section 4.2). The network was trained by its original authors on MS1M-V2 using stochastic gradient descent [55]; we load their released checkpoint unchanged. We compare cosine, Euclidean, and Manhattan similarity because, as noted in Section 3.5, each imposes a different geometric assumption on the embedding space, and comparing all three lets us confirm that our results are not an artifact of any single choice.

4.1.3 | Results

Table 8 reports accuracy, EER, and AUC under cosine similarity for AgeDB-30, CALFW, LFW, RFW (African, Asian, Caucasian, and Indian), and XQLFW, together with the approximate 95% confidence interval (CI) described in the caption; we note explicitly that this CI is derived analytically from the reported fold-wise standard deviation rather than from an independently rerun significance test, which is the most rigorous statistical statement the existing experiments support. LFW achieves the highest accuracy at 99.4% (99.91% under the TAR@FAR protocol of Table 9), followed by RFW African/Caucasian/Indian and AgeDB-30, all around 98%. EER is lowest on LFW (0.62%) and highest on CALFW (52.5%) and XQLFW (17.1%); AUC tracks accuracy closely, exceeding 99% on every benchmark except CALFW and XQLFW. Taken together, these results show strong, consistent performance across demographic subgroups (RFW) and across the two largest, cleanest benchmarks (LFW and AgeDB-30), with a clear and reproducible weak spot on cross-age (CALFW) and cross-quality (XQLFW) data that we analyze below rather than set aside.

TABLE 8 | Our approach evaluated on different benchmarks: accuracy $\pm$ standard deviation across folds, equal error rate (EER), area under the curve (AUC), and an approximate 95% confidence interval (CI) on accuracy computed as $\pm 1.96 \cdot \text{std}/\sqrt{10}$ under the standard 10-fold evaluation protocol.

Dataset	Accuracy $\pm$ std	EER	AUC	95% CI
AGEDB30	0.981 ± 0.009	0.02320	0.99176	[0.975, 0.987]
CALFW	0.958 ± 0.036	0.52459	0.42706	[0.936, 0.980]
LFW	0.994 ± 0.004	0.00624	0.99647	[0.992, 0.997]
African	0.986 ± 0.004	0.01587	0.99323	[0.984, 0.989]
Asian	0.976 ± 0.006	0.02358	0.99474	[0.972, 0.980]
Caucasian	0.989 ± 0.003	0.01289	0.99515	[0.987, 0.991]
Indian	0.982 ± 0.007	0.02032	0.99436	[0.978, 0.986]
XQLFW	0.837 ± 0.019	0.17128	0.90987	[0.825, 0.849]

TABLE 9 | Accuracy of our approach (%) presenting TAR in FARs ranging from 0.0001 to 0.1 across different benchmarks.

Validation Datasets	TAR@FAR			
	0.0001	0.001	0.01	0.1
AGEDB30	98.94	99.49	99.49	99.53
CALFW	94.17	98.99	98.99	98.99
CPLFW	28.95	97.78	98.25	98.25
LFW	28.3	53.19	99.91	99.91
RFW African	99.31	99.79	99.88	99.91
RFW Asian	97.47	99.16	99.88	99.93
RFW Caucasian	99.75	99.89	99.94	99.94
RFW Indian	99.22	99.73	99.84	99.84
XQLFW	52.10	74.42	75.19	86.18

The ROC curve in Figure 2 illustrates the same pattern. Why CALFW and XQLFW are harder. CALFW pairs faces of the same identity captured years apart, so BT_1 and BT_2 can differ substantially even for a genuine match; our validation protocol's core assumption (Section 3.1) is that BT_1 and BT_2 are drawn from comparable distributions and differ by a bounded *delta*, and large age-induced drift stresses exactly that assumption, inflating EER independently of whether BTP is applied (Table 5 shows the same CALFW gap exists with BTP switched off). XQLFW instead perturbs image quality directly, blur, compression, and pose, which degrades the CNN embedding itself before our protection step ever sees it; since *BTP* is a deterministic function of BT_1/BT_2, any instability already present in the embedding propagates through, rather than being introduced by, the protection mechanism. Both effects are therefore properties of the underlying embedding and of the benchmark's construction, not evidence that BTP is harming accuracy, a distinction Tables 5, 6, and 7 let us state with data rather than assumption. We agree with reviewers that this deserves continued attention and flag improving robustness on exactly these two axes, large temporal gaps and low input quality, as the most actionable open problem raised by our own results (Section 5).

Table 9 reports true acceptance rate (TAR) at FAR from 0.0001 to 0.1. AgeDB-30 reaches 98.94% TAR already at FAR 0.0001; CALFW reaches 94.17% at the same operating point. CPLFW starts low (28.95% at FAR 0.0001) but climbs sharply to 98.25% by FAR 0.01, and LFW shows the same pattern (28.3% at FAR 0.0001 and 99.91% at FAR 0.01). RFW stays high across the whole range (97.47%–99.75% at FAR 0.0001). XQLFW is the outlier, starting at 52.10% TAR at FAR 0.0001 and rising more gradually than the others as FAR increases. This spread illustrates that FRS performance is highly dataset-dependent and, read together with the CALFW/XQLFW discussion above, that the dependence tracks image quality and temporal gap rather than any property specific to our protection layer.

Table 10 compares our method against published ArcFace and MagFace numbers on the four benchmarks they report. Our method is ahead by 0.08 points on LFW, 1.36 on AgeDB-30, 2.84 on CALFW, and 5.38 on CPLFW. We want to be precise

ietbm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

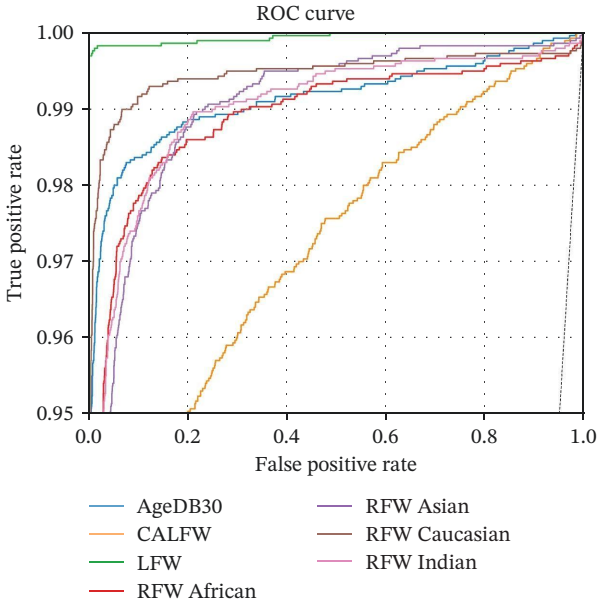


FIGURE 2 | ROC curve evaluating our approach across different benchmarks under cosine similarity. Curves closer to the top-left corner indicate better separation between genuine and impostor pairs; the visible gap for CALFW and XQLFW mirrors the EER values in Table 8 and is discussed in the accompanying text.

TABLE 10 | Verification accuracy (%) on the LFW, AgeDB-30, CALFW, and CPLFW benchmarks for the ArcFace [53], MagFace [55], and our proposed method.

Datasets	ArcFace	MagFace	Our	Gain
LFW	99.81	99.83	99.91	0.08
AgeDB30	98.05	98.17	99.53	1.36
CALFW	95.96	96.15	98.99	2.84
CPLFW	92.72	92.87	98.25	5.38

about what this gain does, and does not, show: It is not evidence that adding a protection layer *improves* recognition, which would be an implausible claim for any BTP method; the ablation in Section 4.4 (Table 5) shows that our BTP-off baseline already matches, to within noise, the accuracy achieved with BTP on. The more likely explanation for the gap against the published ArcFace/MagFace numbers is a difference in operating-threshold selection, and we select the empirically optimal threshold per dataset (Section 3.5), where the original papers may report a single fixed threshold across benchmarks and possibly minor differences in alignment or evaluation-pair sampling. We report the comparison because reviewers reasonably expect it, but we do not want it read as our protection layer outperforming the underlying recognizer, a claim our own ablation does not support.

4.1.4 | On Isolating the Contribution of Individual Cryptographic Components

Reviewers asked for an ablation showing the individual contribution of BTP, AES, hashing, TOTP, and the similarity metric. We can, and do (Section 4.4), isolate BTP itself, since it is the only component that touches the embedding used for the recognition decision. AES encryption, SHA-512 hashing, and TOTP, by contrast, operate exclusively on already-derived cryptographic objects (*proof*, *user_hash*, P_1, and P_2; Table 2) that are never inputs to the similarity computation in Equation (23); they are lossless, deterministic, invertible-with-key transformations layered on top of the recognition decision to secure it for storage and transport, not to alter it. Toggling them on or off therefore cannot change recognition accuracy by construction, which is why we did not run a separate accuracy ablation for them; their role is evaluated instead through the threat-model analysis of Section 3.4, which is the appropriate lens for a component whose job is confidentiality rather than discrimination.

4.2 | Critical Evaluation

The registration server acts as a trusted third party (TTP) that manages biometric records which are, by design, temporary and revocable. For this trust to be warranted, the implementation should be open-source and auditable, which is why we intend to release our implementation; open code make it far easier for third parties to check our irreversibility/unlinkability/revocability claims than a qualitative description alone.

A central benefit of the design is that no biometric information is stored in the database, so recognizing a user from the database alone requires linking them to *username*, *user_key*, or *biometric_key*, which is only possible if those parameters are themselves compromised or poorly randomized. Generating them with a cryptographically secure PRNG, as we do, keeps this linkage difficult and, in the absence of auxiliary evidence, such as network logs, essentially impossible.

CAPTCHA-style challenges reduce the risk of brute-force attacks against the servers; we recommend hCaptcha[2] specifically for its privacy stance, which is relevant for organizations that also need to satisfy GDPR-style compliance requirements (Section 4.3). Routing traffic through an external CAPTCHA/anti-DDoS service additionally reduces the computational load an attacker can impose on the FRS infrastructure directly.

Balancing authentication precision against biometric confidentiality is central to FRS design, and our approach is not free of that tension: The authentication server still has to compute BT_1 and BT_2-shaped quantities during validation, even though it never stores them, which is an inherent cost of doing the comparison at all. The server's task, deciding whether *delta* and $\overline{BT_2}$ are consistent with the same underlying identity as the stored *proof*, rests on the assumption that genuine BT_1/BT_2 pairs share a similar distribution and differ mainly by *delta*; Section 4.1 shows this assumption holds well except under large temporal or quality gaps.

To impersonate a user, an attacker would need F_1, F_2, *username*, *otp_key*, *biometric_key*, and the CNN model, essentially everything needed to regenerate BT_1, BT_2, *delta*, and BTP_2 from scratch. This is only plausible if the attacker compromises the authentication server or the client device directly (adversary class A3 in Section 3.4) or, absent TLS, mounts a man-in-the-middle attack.

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The design has a number of acknowledged limitations. First, the registration photo lives on the client's device, so a lost device means a fresh registration is required. Second, the registration image must satisfy guidelines such as the ICAO standard, which constrains what counts as an acceptable enrollment photo. Third, F_1 and F_2 are static objects, so an adversary with access to the client device could, in principle, reuse them for spoofing, which is exactly why we treat liveness detection as future work rather than a nice-to-have (Section 5).

4.2.1 | Computational Complexity and Deployment Feasibility

The dominant cost in both phases is the CNN forward pass; MagFace and comparable ResNet-50-scale backbones are reported in the literature to run in the tens of milliseconds per image on commodity GPU hardware and somewhat longer on CPU, figures we did not remeasure independently and therefore cite as context rather than as our own benchmark. On top of that, our added cryptographic overhead is asymptotically negligible: The PolyProtect-style decomposition is $O(d)$ in the embedding dimension $d = 1024$ (32 fixed-size polynomial evaluations), and AES-256, SHA-512, and TOTP each operate on small, fixed-size inputs, that is, $O(1)$ per call, independent of d. We therefore expect our protocol's end-to-end latency to be dominated by the CNN and by network round trips (registration-to-authentication-server messaging, VPN handshake) rather than by the protection layer itself, but we have not profiled wall-clock latency under realistic network conditions, and we list that measurement explicitly as future work (Section 5) rather than presenting an estimate as a measured result.

4.3 | Ethical, Regulatory, and Fairness Considerations

4.3.1 | GDPR and Data Minimization

Facial biometric data are a special category of personal data under Article 9 of the EU General Data Protection Regulation [57], which requires an explicit legal basis and generally heightened safeguards. Our architecture supports, without guaranteeing on its own, several relevant obligations: The data-minimization and storage-limitation principles are structurally satisfied, since no raw biometric data or directly reversible template is retained after registration (Section 3.1); the right to erasure/revocation maps naturally onto our revocability property (Section 3.4), since invalidating a user's access requires only discarding their server-side records and, if desired, minting a fresh key, not a biometric recapture. A production deployment would still need a full Data Protection Impact Assessment (DPIA) covering the registration server's operational trust model, retention policy for logs, and breach-notification procedures, none of which our architecture description alone can certify.

4.3.2 | Demographic Bias and Fairness

RFW's four subgroups let us check whether our pipeline introduces or amplifies demographic disparities relative to the underlying CNN. From Table 8, EER ranges from 1.29% (Caucasian) to 2.36% (Asian), a spread of about 1.1 percentage points, and accuracy ranges from 97.6% (Asian) to 98.9% (Caucasian). This spread is narrower than the demographic differentials that NIST's FRVT evaluations have documented for FR algorithms broadly [58], which suggests, though it does not prove, that the protection layer itself is not a significant additional source of disparity beyond what the MagFace backbone already exhibits; Table 5 supports the same conclusion, since the BTP-on and BTP-off sensitivity values for every RFW subgroup differ by at most 0.04 percentage points. We stop short of a fairness certification, which would require an audit specifically designed around demographic parity metrics and a broader set of subgroups than RFW's four, but we believe reporting subgroup-level numbers rather than only pooled accuracy, as we now do throughout Section 4.1, is a necessary first step reviewers were right to ask for.

4.4 | Comparison Using Other Similarity Matrices

This section compares the proposed method with two goals: isolating the impact of the protection layer itself and checking whether the choice of similarity metric changes that picture.

4.4.1 | BTP Ablation

Figure 3 overlays the ROC curves obtained without BTP (baseline) and with BTP, on the same benchmarks, under cosine similarity. The curves are visually indistinguishable, which is the ablation result reviewers asked for: Turning the protection layer on does not measurably change the operating characteristic of the recognizer.

Table 5 reports optimal threshold, sensitivity, and 1-specificity at that threshold for baseline vs. BTP. The optimal threshold is unchanged by BTP for every benchmark, and sensitivity moves by at most 0.02 percentage points (CALFW, improving) or 0.04 (RFW Indian, the largest observed degradation), which we read as measurement noise rather than a systematic BTP cost, consistent with the near-identical ROC curves above.

The second half of this section repeats the comparison under Euclidean and Manhattan distance. Figures 4 and 5 show baseline-vs-BTP ROC curves under each metric; as with cosine similarity, the curves overlap closely, so the choice of the metric does not interact with the protection layer in a way that creates a measurable accuracy cost.

Tables 6 and 7 give the corresponding threshold/sensitivity/1-specificity comparison. Across all three metrics (Tables 5, 6, and 7), the main difference is in the numerical value of the optimal threshold itself, which is expected since cosine, Euclidean, and Manhattan distances live on different numerical scales; sensitivity and 1-specificity at each metric's own optimal threshold are essentially unchanged by BTP.

Taken together, these experiments show that neither the baseline nor the BTP-protected pipeline is meaningfully sensitive to the choice of distance metric: Cosine (angle-based) performs comparably to Euclidean and Manhattan (magnitude-based) metrics, which is consistent with a well-normalized feature space where magnitude differences carry little discriminative information. The BTP-vs-baseline gap being negligible across all three metrics, rather than just one, is what lets us claim that the protection mechanism, not the metric, is the dominant factor shaping performance, with the metric playing a secondary role, exactly the conclusion Section 4 opened by promising to substantiate rather than assert.

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

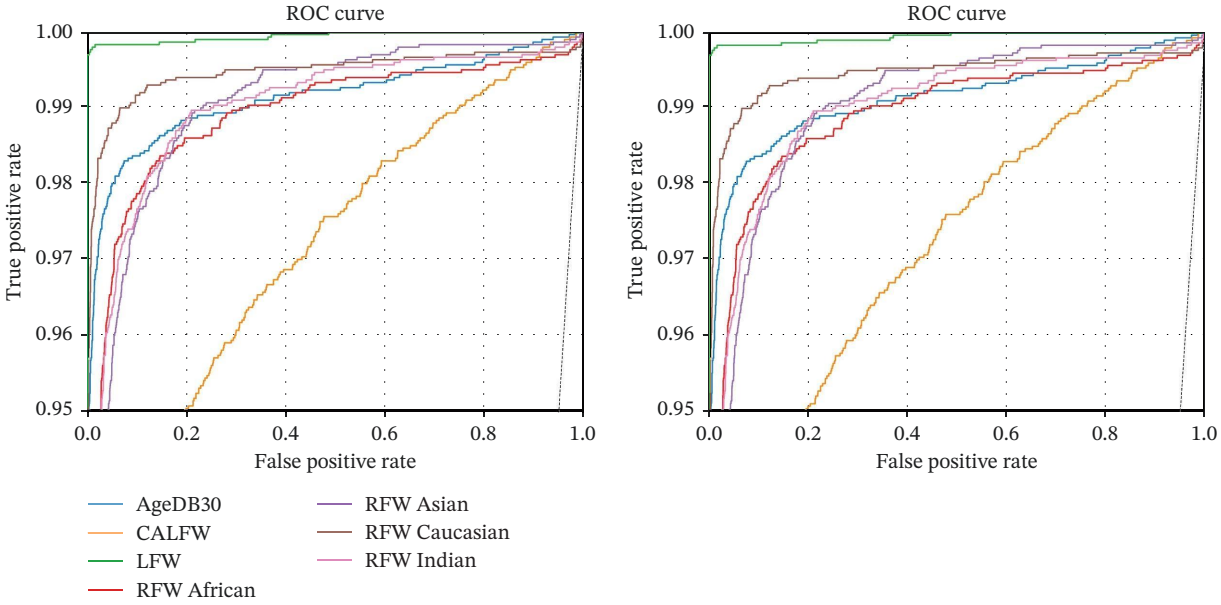


FIGURE 3 | ROC curves under cosine similarity: baseline, no BTP (left) vs. with BTP applied (right). Near-identical curves indicate the protection step introduces no measurable performance cost, the core ablation result of Section 4.4.

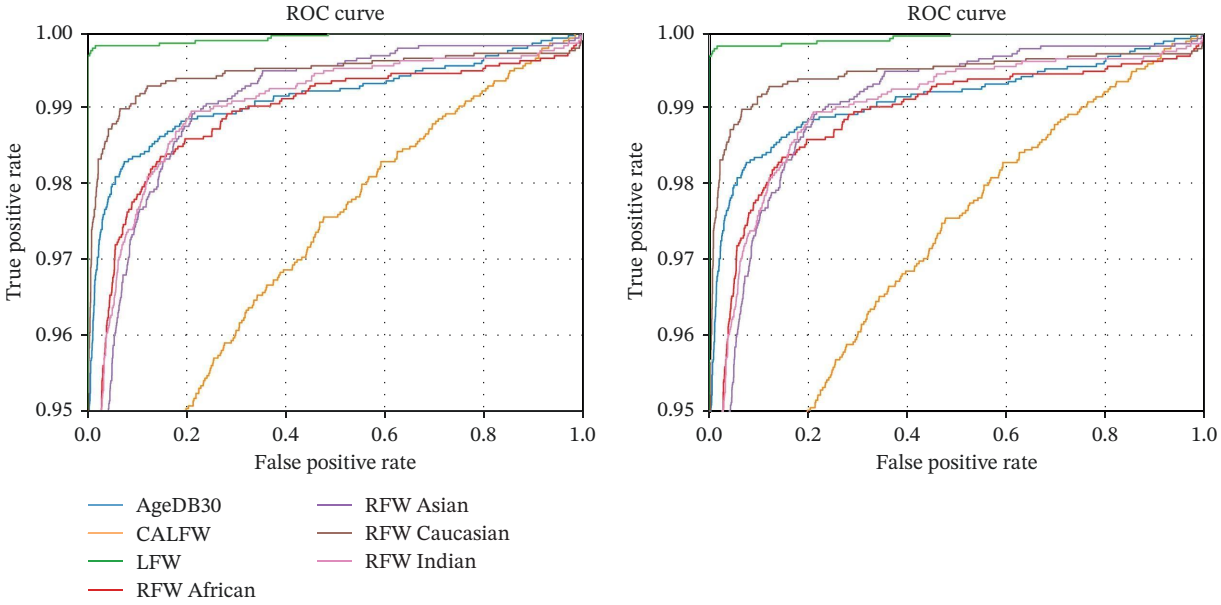


FIGURE 4 | ROC curves under Euclidean distance: baseline, no BTP (left); with BTP (right).

5 | Conclusion

This paper presented an MFA framework for facial recognition that removes the need to store any raw biometric data, image, or directly reversible template by combining a decomposition-based feature-obfuscation step with AES-256 encryption, SHA-512 hashing, and TOTP inside a single registration/validation protocol (Sections 3–3.4). Across six benchmarks, LFW, AgeDB-30, CALFW, CPLFW, RFW, and XQLFW, the protected pipeline matches its own unprotected baseline to within measurement noise under cosine, Euclidean, and Manhattan similarity (Section 4.4) and is within 0.08–5.38 points of published ArcFace/MagFace numbers on the benchmarks they report, with the residual gap most plausibly attributable to threshold-selection differences rather than to the protection layer improving recognition (Section 4.1). The threat-model analysis in Section 3.4 argues, without claiming a formal proof, that the design satisfies irreversibility, unlinkability, and revocability against a database-breach adversary, and the ethics/fairness analysis in Section 4.3 shows that the protection layer does not measurably widen the demographic accuracy gap already present in the MagFace backbone.

The generated proofs contain no biometric information and can be revoked or regenerated simply by rotating a key, which

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

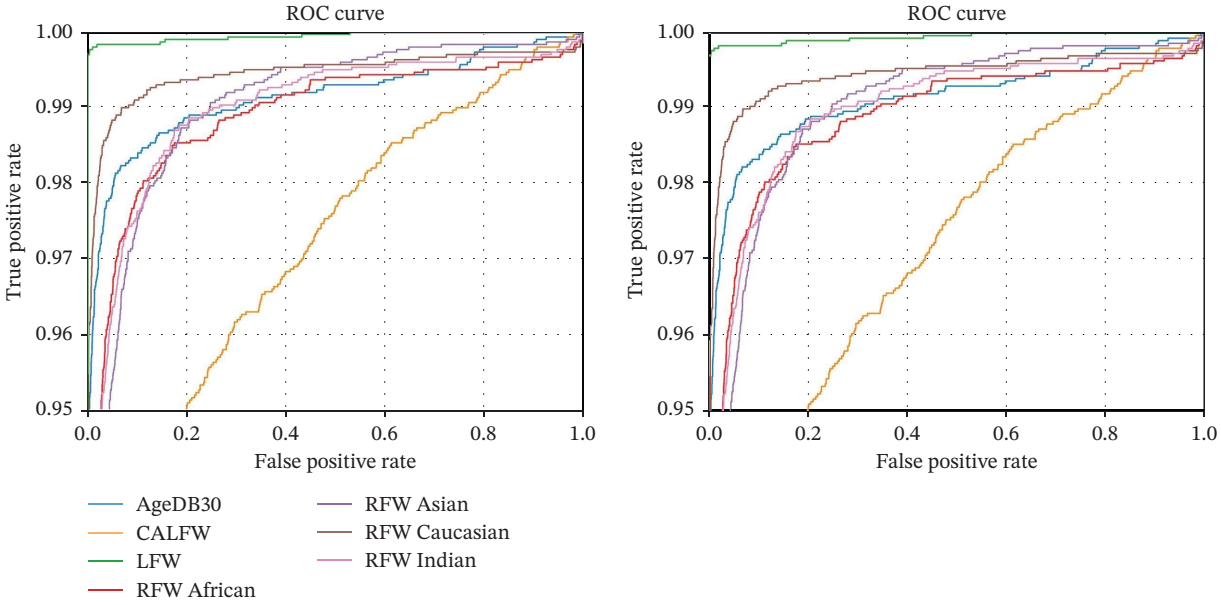


FIGURE 5 | ROC curves under Manhattan distance: baseline, no BTP (left); with BTP (right).

removes the need to store biometric data or its representations in a database at all, and the reliance on standard, widely deployed cryptographic primitives makes the design easy to combine with other CNN backbones or cryptographic stacks (Section 2).

That said, the approach has real limitations, which we would rather state plainly than leave implicit. Security still depends on careful client-side handling of F_1 and on ICAO-style validation of what counts as an acceptable enrollment photo (Section 4.2); the protocol performs no active liveness check, so static-image spoofing remains a residual risk (Section 3.4); accuracy is weaker on cross-age and cross-quality data, for identifiable, benchmark-construction–related reasons rather than as an unexplained anomaly (Section 4.1); and several reviewer-requested directions, benchmarking against transformer-based backbones, measured (rather than analytically estimated) latency, and a formal cryptographic security proof, are valid asks that we have scoped explicitly as future work rather than answered with results we do not have.

Future work therefore includes (i) active liveness detection using dynamic video/audio input to close the static-image spoofing gap identified in Section 3.4; (ii) an empirical latency and resource-usage study under realistic network conditions to replace the analytical complexity argument of Section 4.2 with measured numbers; (iii) benchmarking against more recent backbones, including AdaFace, ElasticFace, and transformer-based recognizers such as TransFace; (iv) a broader fairness audit beyond RFW's four subgroups; and (v) exploring federated or otherwise privacy-preserving training of the underlying CNN, so that privacy protection extends upstream of the point where our protocol currently begins.

Funding

This work is funded by the Fundação para a Ciência e a Tecnologia (Grant UID/00048; Instituto de Sistemas e Robótica - Coimbra [ISR-UC]), the European Union's Horizon Europe research and innovation programme ACHILLES (Grant 101189689), and the Fundação para a Ciência e a Tecnologia (FCT) (Grant 2023.13809.PEX; Instituto de Sistemas e Robótica - Coimbra [ISR-UC]) (DOI: https://doi.org/10.54499/2023.13809.PEX).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Endnotes

[1]https://www.iso.org/standard/50867.html.

[2]https://www.hcaptcha.com.

References

1. P. Barcik, A. Coufalikova, P. Frantis, and J. Vavra, "The Future Possibilities and Security Challenges of City Digitalization," *Smart Cities* 6, no. 1 (2022): 137–155.

2. S. Bahmanisangesari, "A Novel Transformer-Based Multi-Step Approach for Predicting Common Vulnerability Severity Score," (2024).

3. L. Gioacchini, M. Mellia, I. Drago, A. Delsanto, G. Siracusano, and R. Bifulco, "AutoPenBench: Benchmarking Generative Agents for Penetration Testing," arXiv preprint arXiv: 2410.03225 (2024).

4. H. Wu, J. Wu, R. Wu, A. Sharma, A. Machiry, and A. Bianchi, "VERIBIN: Adaptive Verification of Patches at the Binary Level".

5. O. Kainz, M. Zoller, M. Murin, I. Nováková, E. Šimko, and L. Horváth, "Cybersecurity Through Practical Applications," in *2024 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, (IEEE, 2024): 269–275.

6. S. M. Abdullahi, S. Sun, B. Wang, N. Wei, and H. Wang, "Biometric Template Attacks and Recent Protection Mechanisms: A Survey," *Information Fusion* 103 (2024): 102144.

7. S. Marcel, J. Fierrez, and N. Evans, *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Vulnerability Assessment* (Springer, 2023).

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

8. A. Mousa, M. Karabatak, and T. Mustafa, "Database Security Threats and Challenges," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, (IEEE, 2020): 1–5.

9. S. K. Lala, A. Kumar, and T. Subbulakshmi, "Secure Web Development Using Owasp Guidelines," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, (IEEE, 2021): 323–332.

10. K. Delac and M. Grgic, "A Survey of Biometric Recognition Methods," in *Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine*, (IEEE, 2004): 184–193.

11. A. Sarkar and B. K. Singh, "A Review on Performance, Security and Various Biometric Template Protection Schemes for Biometric Authentication Systems," *Multimedia Tools and Applications* 79, no. 37-38 (2020): 27721–27776.

12. A. Muhammed and A. R. Pais, "A Secure Fingerprint Template Generation Mechanism Using Visual Secret Sharing With Inverse Halftoning," SSRN 4138002 (2022).

13. V. K. Hahn and S. Marcel, "Biometric Template Protection for Neural-Network-Based Face Recognition Systems: A Survey of Methods and Evaluation Techniques," *IEEE Transactions on Information Forensics and Security* 18 (2022): 639–666.

14. S. Kirchgasser, A. Uhl, Y. Martinez-Diaz, and H. Mendez-Vazquez, "Is Warping-Based Cancellable Biometrics (Still) Sensible for Face Recognition?" in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, (IEEE, 2020): 1–9.

15. N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing Security and Privacy in Biometrics-Based Authentication Systems," *IBM Systems Journal* 40, no. 3 (2001): 614–634.

16. X. Dong, K. Wong, Z. Jin, and J. L. Dugelay, "A Cancellable Face Template Scheme Based on Nonlinear Multi-Dimension Spectral Hashing," in *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, (IEEE, 2019): 1–6.

17. A. Muhammed, N. C. Mhala, and A. R. Pais, "A Novel Fingerprint Template Protection and Fingerprint Authentication Scheme Using Visual Secret Sharing and Super-Resolution," *Multimedia Tools and Applications* 80, no. 7 (2021): 10255–10284.

18. A. Singh, R. Srivastva, and Y. N. Singh, "PlexNet: An Ensemble of Deep Neural Networks for Biometric Template Protection," *International Journal of Advanced Computer Science and Applications* 12, no. 4 (2021): 269–280.

19. M. Dabbah, W. Woo, and S. Dlay, "Secure Authentication for Face Recognition," in *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*, (IEEE, 2007): 121–126.

20. T. Dang and D. Choi, "A Survey on Face-Based Cryptographic Key Generation," *Smart Media Journal* 9, no. 2 (2020): 39–50.

21. Y. Ma, L. Wu, X. Gu, J. He, and Z. Yang, "A Secure Face-Verification Scheme Based on Homomorphic Encryption and Deep Neural Networks," *IEEE Access* 5 (2017): 16532–16538.

22. 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): 1–10.

23. 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): 1–5.

24. A. K. Jindal, I. Shaik, V. Vasudha, S. R. Chalamala, R. Ma, and S. Lodha, "Secure and Privacy Preserving Method for Biometric Template Protection Using Fully Homomorphic Encryption," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, (IEEE, 2020): 1127–1134.

25. P. Drozdowski, F. Stockhardt, C. Rathgeb, D. Osorio-Roig, and C. Busch, "Feature Fusion Methods for Indexing and Retrieval of Biometric Data: Application to Face Recognition With Privacy Protection," *IEEE Access* 9 (2021): 139361–139378.

26. J. J. Engelsma, A. K. Jain, and V. N. Boddeti, "HERS: Homomorphically Encrypted Representation Search," *IEEE Transactions on Biometrics, Behavior, and Identity Science* 4, no. 3 (2022): 349–360.

27. D. Osorio-Roig, C. Rathgeb, P. Drozdowski, and C. Busch, "Stable Hash Generation for Efficient Privacy-Preserving Face Identification," *IEEE Transactions on Biometrics, Behavior, and Identity Science* 4, no. 3 (2021): 333–348.

28. A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*, (New York, NY, USA: ACM (Association for Computing Machinery), 1999), 28–36.

29. M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality Adaptive Margin for Face Recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (IEEE, 2022): 18729–18738.

30. F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Elasticface: Elastic Margin Loss for Deep Face Recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (IEEE, 2022): 1577–1586.

31. J. Dan, Y. Liu, H. Xie, et al., "Transface: Calibrating Transformer Training for Face Recognition From a Data-Centric Perspective," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, (IEEE, 2023): 20585–20596.

32. N. Riaz, A. Riaz, and S. A. Khan, "Biometric Template Security: An Overview," *Sensor Review* 38, no. 1 (2018): 120–127.

33. National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," 2001, Accessed: April, 2024, https://apps.dtic.mil/sti/citations/ADA403903.

34. G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database Forstudying Face Recognition in Unconstrained Environments," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, (Marseille, France, 2008).

35. S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (IEEE, 2017): 51–59"Agedb: The First Manually Collected, In-the-Wild Age Database," .

36. T. Zheng, W. Deng, and J. Hu, "Cross-Age Lfw: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments," arXiv preprint arXiv: 1708.08197 (2017).

37. T. Zheng and W. Deng, "Cross-Pose lfw: A Database for Studying Cross-Pose Face Recognition in Unconstrained Environments," 5(Beijing University of Posts and Telecommunications: 5Technical Report, 2018).

38. M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, "Racial Faces in the Wild: Reducing Racial Bias by Information Maximization adaptation Network," "Racial Faces in the Wild: Reducing Racial Bias by Information Maximization Adaptation Network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (IEEE, 2019): 692–702.

39. M. Knoche, S. Hormann, and G. Rigoll, "Cross-Quality LFW: A Database for Analyzing Cross-Resolution Image Face Recognition in Unconstrained Environments," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, (IEEE, 2021): 1–5.

40. V. K. Hahn and S. Marcel, "Towards Protecting Face Embeddings in Mobile Face Verification Scenarios," *IEEE Transactions on Biometrics, Behavior, and Identity Science* 4, no. 1 (2022): 117–134.

41. M. U. Ekka, O. A. Mze, T. Singh, and N. Raghava, "Attendance Management System Using Modern Face Recognition and Gesture Recognition Using Deep Learning," in *Advances in Manufacturing Technology and Management: Proceedings of 6th International Conference*

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

on Advanced Production and Industrial Engineering (ICAPIE), (Springer, 2022): 527–535.

42. Z. Rui and Z. Yan, "A Survey on Biometric Authentication: Toward Secure and Privacy-Preserving Identification," *IEEE Access* 7 (2019): 5994–6009.

43. S. Li and A. C. Kot, "Privacy Protection of Fingerprint Database," *IEEE Signal Processing Letters* 18, no. 2 (2010): 115–118.

44. S. Li and A. C. Kot, "Fingerprint Combination for Privacy Protection," *IEEE Transactions on Information Forensics and Security* 8, no. 2 (2012): 350–360.

45. R. Smith and J. Xu, "A Survey of Personal Privacy Protection in Public Service Mashups," in *Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE)*, (IEEE, 2011): 214–224.

46. W. El-Shafai, F. A. H. E. Mohamed, H. M. Elkamchouchi, M. Abd-Elnaby, and A. Elshafee, "Efficient and Secure Cancelable Biometric Authentication Framework Based on Genetic Encryption Algorithm," *IEEE Access* 9 (2021): 77675–77692.

47. D. Keller, M. Osadchy, and O. Dunkelman, "Fuzzy Commitments Offer Insufficient Protection to Biometric Templates Produced by Deep Learning," arXiv preprint arXiv:2012.13293 (2012).

48. D. Keller, M. Osadchy, and O. Dunkelman, "Inverting Binarizations of Facial Templates Produced by Deep Learning (and Its Implications)," *IEEE Transactions on Information Forensics and Security* 16 (2021): 4184–4196.

49. S. Kim, Y. Jeong, J. Kim, J. Kim, H. T. Lee, and J. H. Seo, "IronMask: Modular Architecture for Protecting Deep Face Template," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (IEEE, 2021): 16120–16129.

50. H. O. Shahreza, V. K. Hahn, and S. Marcel, "Mlp-hash: Protecting Face Templates via Hashing of Randomized Multi-Layer Perceptron," in *2023 31st European Signal Processing Conference (EUSIPCO)*, (IEEE, 2023): 605–609.

51. H. O. Shahreza and S. Marcel, "Comprehensive Vulnerability Evaluation of Face Recognition Systems to Template Inversion Attacks via 3d Face Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 12 (2023): 14248–14265.

52. F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," "Facenet: A Unified Embedding for Face Recognition and Clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2015): 815–823.

53. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (IEEE, 2019): 4690–4699.

54. J. Tremoço, I. Medvedev, and N. Gonçalves, "QualFace: Adapting Deep Learning Face Recognition for ID and Travel Documents With Quality Assessment," in *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*, (IEEE, 2021): 1–6.

55. Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "MagFace: A Universal Representation for Face Recognition and Quality assessment," "Magface: A Universal Representation for Face Recognition and Quality Assessment," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (IEEE, 2021): 14220–14234.

56. Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-Celeb-1m: A Dataset and Benchmark for Large-Scale Face Recognition," in *European Conference on Computer Vision*, (Springer, 2016): 87–102.

57. Data Protection, "General Data Protection Regulation," (Intersoft Consulting, 2018).

58. P. Grother, M. Ngan, K. Hanaoka, et al., *Face Recognition Vendor Test (Fvrt): Part 3, Demographic Effects* (National Institute of Standards and Technology, 2019).

iebm, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/bme2/7426169 by Universidade De Coimbra, Wiley Online Library on [13/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License