applied sciences

MDPI

Article

Post-Quantum Biometric Authentication Based on Homomorphic Encryption and Classic McEliece

Rosario Arjona *, Paula López-González, Roberto Román and Iluminada Baturone

Instituto de Microelectrónica de Sevilla (IMSE-CNM), Universidad de Sevilla-CSIC, 41092 Seville, Spain

* Correspondence: arjona@imse-cnm.csic.es

Abstract: Homomorphic encryption is a powerful mechanism that allows sensitive data, such as biometric data, to be compared in a protected way, revealing only the comparison result when the private key is known. This is very useful for non-device-centric authentication architectures with clients that provide protected data and external servers that authenticate them. While many reported solutions do not follow standards and are not resistant to quantum computer attacks, this work proposes a secure biometric authentication scheme that applies homomorphic encryption based on the Classic McEliece public-key encryption algorithm, which is a round 4 candidate of the NIST post-quantum standardization process. The scheme applies specific steps to transform the features extracted from biometric samples. Its use is proposed in a non-device-centric biometric authentication architecture that ensures user privacy. Irreversibility, revocability and unlinkability are satisfied and the scheme is robust to stolen-device, False-Acceptance Rate (FAR) and similarity-based attacks as well as to honest-but-curious servers. In addition to the security achieved by the McEliece system, which remains stable over 40 years of attacks, the proposal allows for very reduced storage and communication overheads as well as low computational cost. A practical implementation of a non-device-centric facial authentication system is illustrated based on the generation and comparison of protected FaceNet embeddings. Experimental results with public databases show that the proposed scheme improves the accuracy and the False-Acceptance Rate of the unprotected scheme, maintaining the False-Rejection Rate, allows real-time execution in clients and servers for Classic McEliece security parameter sets of 128 and 256 bits (mceliece348864 and mceliece6688128, respectively), and reduces storage requirements in more than 90.5% compared to the most reduced-size homomorphic encryption-based schemes with post-quantum security reported in the literature.

Keywords: biometric template protection; homomorphic encryption; post-quantum security

check for updates

Citation: Arjona, R.; López-González, P.; Román, R.; Baturone, I. Post-Quantum Biometric Authentication Based on Homomorphic Encryption and Classic McEliece. *Appl. Sci.* **2023**, *13*, 757. https://doi.org/10.3390/app13020757

Academic Editor: Javier Hernando

Received: 23 November 2022
Revised: 22 December 2022
Accepted: 28 December 2022
Published: 5 January 2023

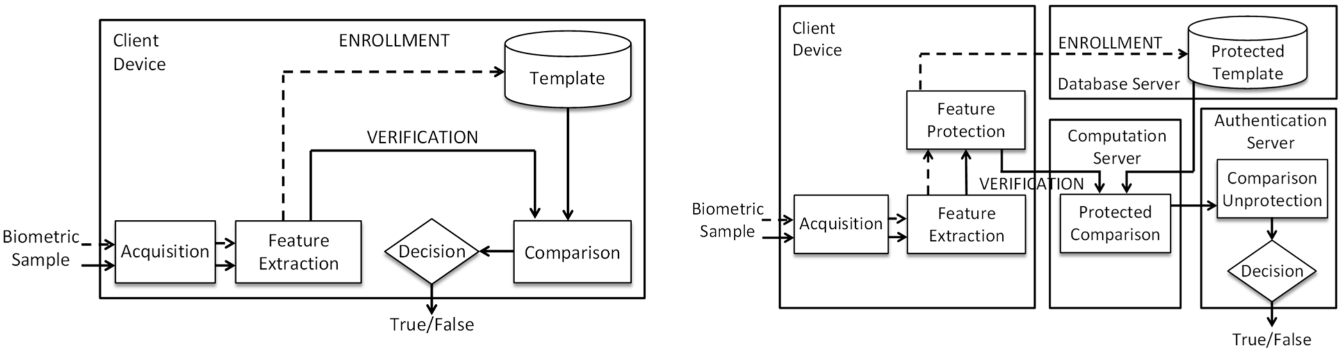
1. Introduction

In electronic transactions, people must be able to prove who they are online. Typically, the person proves that they: (a) know a unique secret ('what the person knows'), (b) have a unique possession ('what the person has') and/or (c) are a physical entity ('who the person is'). The physical entity of a person is defined by his/her biometric characteristics [1], which provide an intrinsic link between the person and the electronic entity without having to know or remember secrets and without the need for possessing a device or card (which can be stolen).

A biometric recognition scheme usually includes an enrollment phase and a verification phase [2]. At the enrollment phase, biometric characteristics of a person are acquired and discriminative features are extracted and stored as a reference. At the verification phase, features are extracted from biometric queries, which are matched to the stored reference. If during matching, the comparison result (typically a distance measurement) satisfies a threshold, the recognition decision determines that the person is successfully authenticated.

1.1. Biometric Authentication Architectures

In a device-centric authentication architecture (Figure 1a), all the phases (enrollment and verification) and, thus, all the operations (acquisition, feature extraction, storage, comparison and decision) of the recognition scheme are carried out locally in the same client device. The client device authenticates the user and, then, an external server authenticates the client device on behalf of the user. Advantages of the device-centric authentication architecture are that the security perimeter is very reduced (it is constrained to the device) and user's biometric data, which are sensitive data, as established by the data-protection regulations from several countries, do not go out of that perimeter. The drawback is that many applications require external evidence about the authenticity of the user and this architecture requires additional secure hardware.



(**a**) (**b**)

Figure 1. Biometric authentication architectures: (**a**) device-centric, (**b**) protected and non-device-centric.

In a non-device-centric authentication architecture, acquisition and feature extraction are performed on the client device, and storage, comparison and decision are performed on external servers. In addition to the user and the device, several authors model a generic non-device-centric biometric system with three entities, which act as database, computation and authentication servers [3]. At the enrollment and verification phases, the device captures and extracts the biometric features from its user and sends them to the external servers. At the enrollment phase, the database server stores the reference (template). At the verification phase, the client device provides the biometric queries and the database server provides the stored reference. The computation server compares the stored reference and the biometric queries. Then, the authentication server applies a threshold value to the comparison result to authenticate the user. In this case, there is an external authentication of the user's physical entity. However, since communication, storage and authentication are performed on the cloud, biometric features should be protected. Homomorphic encryption not only allows for features to be protected at communication and storage, but also whenever they are compared [4–10]. Only the comparison result is obtained in the unprotected domain by the authentication server (as illustrated in Figure 1b).

1.2. Biometric Data Protection

According to the ISO/IEC 24745 standard [11], the requirements of biometric protection schemes are irreversibility (or non-invertibility), unlinkability and revocability (or renewability). Irreversibility ensures that no sensitive information about biometric data is leaked from protected biometric features. Therefore, protected biometric features can be communicated and stored on an external server without revealing sensitive information. Revocability refers to obtaining different protected biometric features from the same biometric sample when the same biometric sample has been employed to enroll the user in different biometric systems with different databases. Unlinkability refers to obtaining different protected biometric features from different biometric samples from the same instance. If protected features are compromised, an adversary cannot recover any sensitive information or know the owner of the protected features. Consequently, the compromised protected features can be destroyed without affecting other parallel or new enrollments. Another issue to consider is that the distances between protected biometric features should not preserve the distances between unprotected features in order to avoid the success of similarity-based attacks based on the use of machine learning techniques [12].

In addition to security requirements, the recognition performance, the size of the protected features as well as the execution times of the authentication steps have to be evaluated [13]. The recognition performance of the unprotected scheme should be preserved or even improved, the verification phase should be carried out at real time and the size of the protected features should be minimized to reduce the storage and transfer operations. Storage and transfer of enrollment data are only performed once to enroll the user. In contrast, since the verification phase is performed many times, the transfer of protected features between the database, computation and authentication servers is higher. Nowadays, this issue is a challenge in reported non-device-centric architectures because, in general, the size of protected features is bigger than unprotected ones.

1.3. Post-Quantum Security

With the availability of large-scale quantum computers in the future, many of the current proposed biometric protection schemes will be broken. In order to provide long-term security, post-quantum cryptography should be considered. Currently, several quantum-resistant biometric protection schemes have been proposed based on lattice cryptography [6–10,14,15]. However, further research is needed to improve the performance and employ standard solutions.

In 2016, the National Institute of Standards and Technology (NIST) began a process with multiple evaluation rounds to develop standards for post-quantum cryptography. In July 2022, selected algorithms for standardization and round 4 candidates were announced [16]. Among public-key encryption and key-establishment algorithms, CRYSTALS- Kyber based on lattice-based cryptography using Module Learning with Errors (Module-LWE) was selected for standardization and Classic McEliece, which employs Goppa code-based cryptography, was selected as a candidate of the round 4. In the Third Round Status Report [17], NIST showed its interest to know specific use cases for which Classic McEliece would be a good solution. Although Classic McEliece requires higher key sizes and generation times than other proposals, this may not be a drawback for biometric recognition systems because keys only need to be generated and transmitted at the enrollment phase. On the other side, Classic McEliece provides the smallest protected data size of any of the NIST post-quantum cryptography candidates, which is interesting to reduce biometric data storage and transmission. These reasons motivated us to analyze Classic McEliece for biometric authentication in non-device-centric architectures.

1.4. Main Contributions

The main contributions of this paper are illustrated in Figure 2 and summarized as follows:

- We propose a new biometric authentication scheme with post-quantum security that conveniently adapts the Classic McEliece algorithm to a non-device-centric architecture with honest-but-curious servers.
- Taking advantage of a homomorphic property of Classic McEliece, the proposal is suitable for privacy-preserving non-device-centric architectures where the communication, storage and comparison of biometric features are performed in the protected domain.
- The recognition performance is greater than or equal to the authentication obtained with unprotected features and provides irreversibility, unlinkability, revocability and resistance to FAR and similarity-based attacks.

- The proposal provides protected biometric data with much smaller size than other solutions with similar security properties, which reduces storage and communication overheads.
- A practical implementation of the proposal in a facial authentication system based on an Android App (for smartphones that act as client devices), Python code (for computers that act as computation and authentication servers) and MySQL (for database servers) demonstrates that authentication is performed in real time.

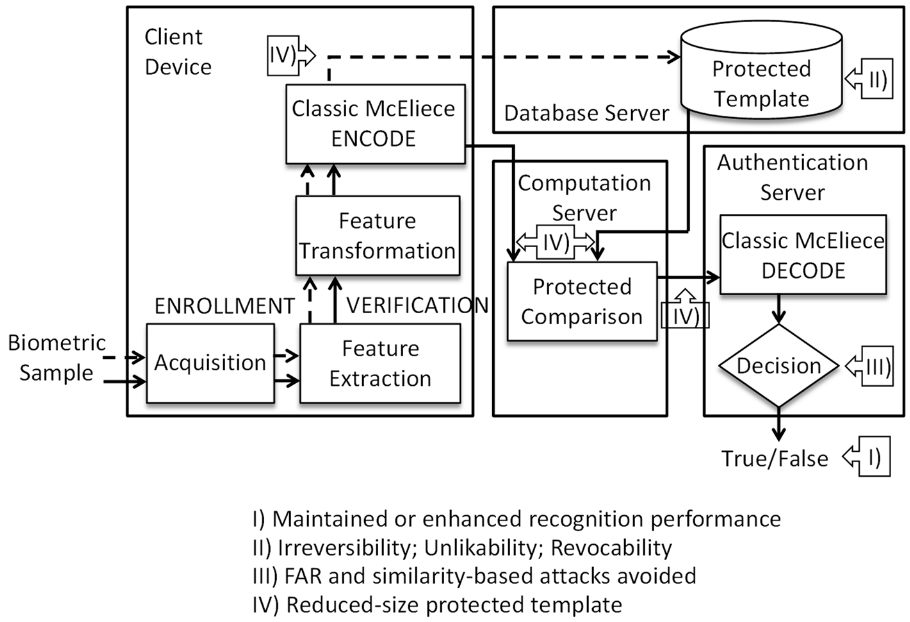


Figure 2. Overview of the proposal with the main contributions indicated.

The paper is structured as follows. Section 2 includes a review of the proposals in the literature for post-quantum-secure biometric schemes. Section 3 presents our proposal of a secure biometric scheme and a security analysis. Section 4 presents a practical realization of the proposed scheme for facial recognition using FaceNet [18]. Finally, Section 5 concludes the work.

2. Related Work

Traditionally, biometric protection schemes have been based on cancelable biometrics and biometric cryptosystems [13]. In cancelable biometrics, feature transformation techniques employ a transformation function f_t to modify the biometric features extracted at the enrollment phase b^R and at the verification phase b^V. Therefore, transformed features $f_t(b^R)$ and $f_t(b^V)$, respectively, are compared in the transformed domain. Generally, transformed features contain less information than the original ones, which degrades the recognition performance compared to the unprotected scheme. In salting techniques, the recognition performance is increased due to the combination of the biometric information and a user-specific secret key. However, the secret key is usually stored in the user device so that the security does not increase in the stolen-device scenario.

In biometric cryptosystems, secure sketches are employed as public data that allow for recovering enrolled biometric data by providing enough similar biometric data at verification. One of the most widely used secure sketches is the fuzzy commitment proposed in [19], which is based on error-correcting codes. In a fuzzy commitment, the public data $f_c(b^R, K^c)$ bind the biometric features b^R with a random secret K^c. The random secret is encoded with the encoding function of the error-correcting code and the result is XORed with the biometric features b^R, $f_c(b^R, K^c) = b^R \oplus encode(K^c)$. The result is stored as a template at the enrollment phase. At the verification phase, the operation

$decode\big(f_c\big(b^R, K^c\big) \oplus b^V\big) = decode\big(b^R \oplus encode(K^c) \oplus b^V\big)$ is performed. If the differences between the enrollment and verification biometric features $\big(b^R \oplus b^V\big)$ are small enough for the error-correcting code to recover the random secret K^c, the authentication is successful. In the case of a non-device-centric architecture, if the authentication server recovers the random secret K^c from the public data, it is also able to recover the biometric data b^R, which is a problem with honest-but-curious servers. The recognition performance of the biometric cryptosystems depends on the correction capability of the error-correcting codes. Consequently, it can be worse than the recognition performance of the unprotected system. Another problem of fuzzy commitments is to achieve unlinkability. Since any linear combination of codewords is also a codeword if using linear error-correcting codes, two fuzzy commitments derived from the same biometric features using different codewords can be linked. This is known as the decodability-based cross-matching attack [20] and limits the binding property. In addition, the low entropy in biometric characteristics (they are not quite random) limits the hiding property and, hence, irreversibility of the fuzzy commitments. Moreover, considering brute-force attacks, also known as FAR (False-Acceptance Rate) attacks when FAR is not 0, unimodal systems usually provide a security from 17 to 24 bits (for FAR values from 10^{-5} to 10^{-7}), which is very low compared to the security in a cryptographic system [21].

In the PQFC (Post-Quantum Fuzzy Commitment) scheme proposed in [14], a matrix A generated randomly and a vector K^c chosen randomly are multiplied. The resulting vector, $random(K^c)$, is added to the biometric data, $f_c\big(b^R, K^c\big) = b^R + random(K^c)$, at the enrollment phase. The $random(K^c)$ is stored in a smart card for the user and $f_c\big(b^R, K^c\big)$ is stored in a database server. At the verification phase, $f_c\big(b^V, K^c\big)$ is computed using the $random(K^c)$ provided by the smart card and it is compared with the stored $f_c\big(b^R, K^c\big)$. If the matching score resulting from the comparison falls within the system threshold, then the user is authenticated. Otherwise, authentication fails. The proposal in [22] is equivalent since the $random(K^c)$ can be seen as the result of applying a key derivation function to a knowledge-based (K^c is related to a password) or a possession-based factor. In [23], the $random(K^c)$ can be seen as provided by a possession-based factor, in particular by a Physically Unclonable Function (PUF) in the user's device. These proposals increase the unlinkability and irreversibility of fuzzy commitments but remain vulnerable to FAR attacks.

The proposal based on LPN (Learning Parity with Noise) commitments in [15] avoids the problems of fuzzy commitments. In this proposal, the accuracy of the unprotected system is preserved, providing irreversibility, revocability and unlinkability with a security level comparable to a cryptographic system, even with unimodal biometric systems. In this proposal, the privacy of the users is also preserved. In LPN commitments, the biometric data are encoded by using a random linear code, with some noise added to the codeword, $f_c\big(b^R, K^c\big) = K^c \oplus encode\big(b^R\big)$. The unknown noise, K^c, should be a low-weight uniformly random vector. At the verification phase, the operation $decode\big(f_c\big(b^R, K^c\big) \oplus f_c\big(b^V, K^c\big)\big)$ is performed. A decoding failure detects that K^c at enrollment and verification are not the same, avoiding FAR attacks. The disadvantage of LPN commitments is the slow execution of the decoding algorithm (4.44 and 50.13 s for a security of 128 and 256 bits, respectively, are reported in [15], although no optimization is applied).

More recently, homomorphic encryption has been applied to biometric recognition [4–10,13]. Several solutions are not resistant to attacks from quantum computers [4,5]. Most of the quantum-resistant solutions are based on lattice cryptography [6–10]. Many of them employ a device-centric architecture, where the biometric features b^R and b^V are encrypted at the client site with the public key of the client pk^C, resulting in $E(b^R, pk^C)$ and $E(b^V, pk^C)$, respectively. The comparison is performed by the server in the encrypted domain $E(b^R, pk^C) \diamond E(b^V, pk^C)$ by applying an evaluation operator $\diamond$. According to the homomorphic properties, $E(b^R \circ b^V, pk^C) = E(b^R, pk^C) \diamond E(b^V, pk^C)$, where $\circ$ is another or the same evaluation operator. From $D\big(E(b^R \circ b^V, pk^C), sk^C\big)$, $b^R \circ b^V$ is decrypted at the client site by using the private key of the client, sk^C. The server, which stores the encrypted templates and calculates the encrypted distance between protected data, cannot obtain information about biometric data because the private key to decrypt is in the client site [5–7]. However, as commented in the Introduction, we are interested in non-device-centric architectures, which are required by many applications. Other quantum-resistant solutions employ a non-device-centric architecture with two servers (authentication and computation servers) in addition to the database and the client [8–10]. The biometric features are encrypted at the client site with the public key of the authentication server. The computation server calculates the encrypted distance between protected data and sends the result to the authentication server. The computation server cannot obtain information about biometric data because the private key to decrypt is in the authentication site and the authentication server only decrypts the distance between the data. The limitation is that both servers cannot collude to avoid the leakage of biometric data, which may not be very realistic.

Among the quantum-resistant solutions commented above, only the proposal in [10] employs CRYSTALS-Kyber, which is the public-key encryption and key-establishment algorithm standardized by the NIST post-quantum standardization process [24]. Classic McEliece, BIKE and HQC are the round 4 candidate algorithms of the contest [16] and can be standardized in the future, after further analysis. Classic McEliece provides, for each security level, the lowest ciphertext size and, thus, the lowest ratio (ciphertext size/plaintext size). This motivated the work presented in this paper because such an advantage is very interesting for biometric applications.

The first McEliece cryptosystem was proposed in 1978 [25], specifically employing Goppa codes. Several proposals for biometric protection schemes are based on the application of the original McEliece cryptosystem to construct secure sketches by using the cryptosystem as the coding method [26–28]. The drawback of the original McEliece cryptosystem is that the size of the keys is large. In order to reduce the key sizes, several proposals employed variants [29]. However, structural attacks were successful for Reed Müller, Reed-Solomon, BCH, Low-Density Parity Check (LDPC) and Moderate-Density Parity Check (MDPC) codes. For variants that employed hyperelliptic curves, convolutional, quasi-cyclic and quasi-dyadic Goppa codes, algebraic attacks were reported. Another variant that applies Goppa codes based on structured error vectors with larger Hamming weights was attacked in [30]. The only variant that has remained stable together with the original McEliece cryptosystem is the Niederreiter cryptosystem proposed in 1994 [31] that also employed Goppa codes and is employed in Classic McEliece [32], as commented, recently proposed as a round 4 candidate of the NIST post-quantum standardization process.

3. Proposed Scheme Based on Homomorphic Encryption and Classic McEliece

3.1. Non-Device-Centric Biometric Authentication Context and Threat Model

We consider a non-device-centric authentication architecture composed of client devices, a database server, an authentication server and a computation server, as in [3]. It is assumed that each user employs his/her client device, such as a smartphone, that acquires the biometric characteristics and implements all the steps of the proposed scheme to protect the biometric data before sending them to the database server at enrollment and to the computation server at verification. The user provides his/her unique identifier ID to the client device. If the ID is sensitive, it is mapped to a public index, as proposed in [33]. At the enrollment phase, the client device maps the user ID to a non-sensitive index x that is stored in the database server. At the verification phase, the client device again provides the index x and the computation server uses a Private Information Retrieval (PIR) [34] to retrieve data from the database without revealing to the database which data are retrieved.

It is assumed that all the operations executed in the client device are carried out in a trusted way, for example, by using a Trusted Execution Environment (TEE). It is also assumed that no attacks can be carried out at the enrollment phase. All the servers are honest at the enrollment phase. If an attacker gains access to the client device at the verification phase, which is known as a stolen-device scenario, the attacker cannot manipulate the client device to carry out unauthorized operations or change the processed data because the client device is tamper-resistant. What the attacker can do is to provide impostor or synthetic biometric characteristics to the client device, known as presentation attacks.

The communication channels between the parties are assumed to be secure so that the messages are confidential and their integrity can be checked. Thus, an external adversary cannot extract or employ the information transmitted by the communication channels. Further, the servers are assumed to be authenticated so that an attacker cannot impersonate the servers.

The servers can be honest-but-curious, that is, they carry out their steps as expected but can try to obtain information about the users' biometric data from the protected data they process or store. It is assumed that this curiosity can lead the servers to collusion to try to obtain information.

3.2. Generation and Matching of Protected Biometric Features

Our proposal for biometric protection adapts the Classic McEliece NIST proposal [32] to the context of the non-device-centric biometric authentication scheme described above. The definition of Classic McEliece, which is based on the Niederreiter cryptosystem, is composed of three main algorithms: key generation (*KeyGen*), encoding (*ENCODE*) and decoding (*DECODE*). The parameter set specifies: $q = 2^m$, $n \leq q$ and $k = n - tm$. *KeyGen* has no inputs and generates the public and private keys. The public key T is the $(n-k) \times k$ random matrix over a finite field of order 2, $\mathbb{F}_2$, such that $I_{n-k} \mid T$ is the $(n-k) \times n$ parity-check matrix H for the Goppa code where I is a $(n-k) \times (n-k)$ identity matrix. The private key is $\Gamma = (\mathrm{g},\ \alpha_1, \alpha_2, \ldots,\ \alpha_n)$, where g is a monic irreducible polynomial in $\mathbb{F}_q[x]$ of degree t and $(\alpha_1, \alpha_2, \ldots,\ \alpha_n)$ are distinct elements of a finite field of order q, $\mathbb{F}_q$, generated from a uniform random string. *ENCODE* receives, as inputs, the public key T and a vector $e \in \mathbb{F}_2^n$ of Hamming weight t. The output of the algorithm is the vector $c \in \mathbb{F}_2^{n-k}$ computed as $c = H{\cdot}e$ with $H = I_{n-k} \mid T$. *DECODE* receives, as inputs, the private key Γ and a vector $c \in \mathbb{F}_2^{n-k}$. The first step of *DECODE* is to extend c to $v = (c, 0, \ldots, 0) \in \mathbb{F}_2^n$ by appending k zeros. The second step is to find the unique codeword *cod* in the Goppa code defined by Γ, which is at a distance t from v. An advantage of Classic McEliece is that there are fast algorithms to decode these codes, for example, the Berlekamp algorithm. The third step is to set $e = v + cod$. If the Hamming weight of e is t and $c = H{\cdot}e$, then e is returned.

3.2.1. Generation of Protected Biometric Features

The *ENCODE* algorithm is applied at enrollment and verification to obtain the vectors cb^R and cb^V. The pseudocode to generate protected biometric features is shown in Algorithm 1 and works as follows:

1. If biometric features bf are not binary, a binarization process, which preserves distances, is applied, such as Linearly Separable Subcode (LSSC) [35], thus, obtaining a binary biometric feature string b.
2. The binary string b is divided into N substrings, $b = (b_1, \ldots, b_N)$, with, respectively, $z_1, \ldots, z_N$ bits, where z_i is greater than or equal to t, with $i = 1, \ldots,\ N$.
3. Each substring eb_i, with length n, is composed of $(n-k)$ random bits from the position 1 to $(n-k)$, concatenated by the z_i bits of b_i and by $(k - z_i)$ random bits from the position $(n - k + z_i + 1)$ to n. Figure 3 shows the transformation of the binary biometric features b_i^R (Figure 3a) and b_i^V (Figure 3b).

   3.1. The $(n - z_i)$ random bits that perform as a padding come from a secret substring p_i. The secret string $p = (p_1, \ldots, p_N)$ is chosen randomly at enrollment to ensure that the Hamming weights of $eb_1, \ldots, eb_N$ are greater than t.

   - Depending on the realization, string p can be stored in the client device or reconstructed in some way whenever required (from what the user knows or from a seed the device reconstructs with a Physical Unclonable Function [23]).

3.2. The bits from position $(n-k+1)$ to n can be randomly permuted, as illustrated in Figure 3a. The permutation is the same at enrollment and verification, as illustrated in Figure 3b.

3.3. The client device checks that the Hamming weights of the substrings b_i are neither zero nor have a low value.

4. *ENCODE* algorithm is applied to the enrollment and verification substrings eb_i^R and eb_i^V, respectively, obtaining the ciphertexts $cb^R = (cb_1^R, \ldots, cb_N^R)$ and $cb^V = (cb_1^V, \ldots, cb_N^V)$, with $cb_i^R = H \cdot eb_i^R$ and $cb_i^V = H \cdot eb_i^V$.

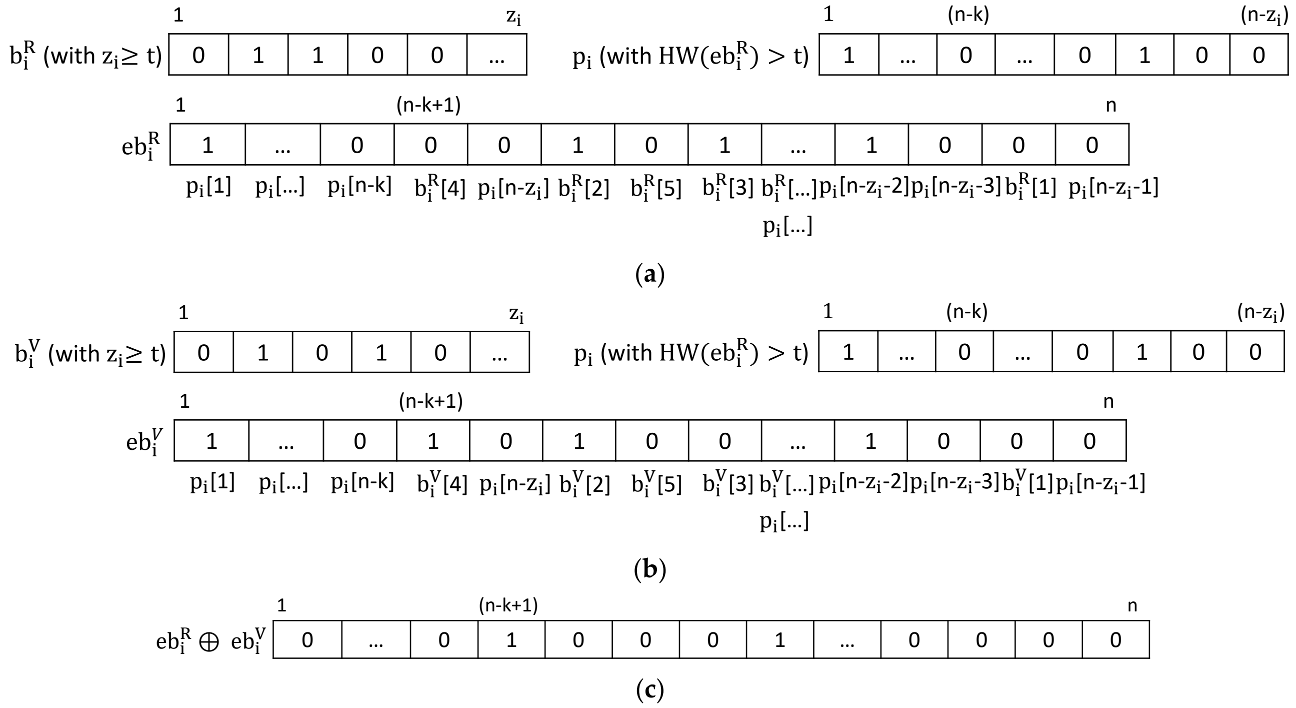

1
z_i
b_i^R (with z_i ≥ t)
0 1 1 0 0 …
1
(n-k)
(n-z_i)
p_i (with HW(eb_i^R) > t)
1 … 0 … 0 1 0 0
1
(n-k+1)
n
eb_i^R
1 … 0 0 0 1 0 1 … 1 0 0 0
p_i[1] p_i[…] p_i[n-k] b_i^R[4] p_i[n-z_i] b_i^R[2] b_i^R[5] b_i^R[3] b_i^R[…] p_i[n-z_i-2] p_i[n-z_i-3] b_i^R[1] p_i[n-z_i-1]
p_i[…]
(a)
1
z_i
b_i^V (with z_i ≥ t)
0 1 0 1 0 …
1
(n-k)
(n-z_i)
p_i (with HW(eb_i^R) > t)
1 … 0 … 0 1 0 0
1
(n-k+1)
n
eb_i^V
1 … 0 1 0 1 0 0 … 1 0 0 0
p_i[1] p_i[…] p_i[n-k] b_i^V[4] p_i[n-z_i] b_i^V[2] b_i^V[5] b_i^V[3] b_i^V[…] p_i[n-z_i-2] p_i[n-z_i-3] b_i^V[1] p_i[n-z_i-1]
p_i[…]
(b)
1
(n-k+1)
n
eb_i^R ⊕ eb_i^V
0 … 0 1 0 0 0 1 … 0 0 0 0
(c)

Figure 3. Biometric Feature Transformation: (**a**) at the enrollment phase and (**b**) at the verification phase. (**c**) shows the application of the XOR operation to the biometric features transformed in (**a**) and (**b**).

Algorithm 1: Pseudocode to generate protected biometric features.

```
    Inputs: bf, t, N, $T$, n, k, p = ($p_1$, … , $p_N$), seed
Output: cb
    IF bf is not binary THEN
b ← Apply preserving distances binarization algorithm to bf
END IF
$b_1$, … , $b_N$ ← Divide b into N substrings with length ≥ t
FOR i = 1 to N DO
$z_i$   ← Length of $b_i$
$eb_i'$ ← Concatenate ($p_i$[1:n − k], $b_i$, $p_i$[n − k + 1:n − $z_i$])
$eb_i$  ← Concatenate ($eb_i'$ [1:n − k], Permutation of $eb_i'$[n − k + 1:n] with seed)
$cb_i$  ← ENCODE($eb_i$, $T$)
END FOR
RETURN the set cb= ($cb_1$, … , $cb_N$) of protected biometric features
```

3.2.2. Matching of Protected Biometric Features

The *DECODE* algorithm is applied at verification to obtain the recognition decision. We propose the following procedure:

1. The XOR operation of the encoded substrings is obtained, that is, $cb_i^R \oplus cb_i^V = H \cdot eb_i^R \oplus H \cdot eb_i^V$.
2. $eb_i^R \oplus eb_i^V$ is recovered by the *DECODE* algorithm. Figure 3c shows the result of the application of the XOR operation to the transformed biometric features eb_i^R and eb_i^V.
   - By the homomorphic property of Classic McEliece, $ENCODE\left(eb_i^R \oplus eb_i^V\right) = ENCODE\left(eb_i^R\right) \oplus ENCODE\left(eb_i^V\right)$. Hence, $cb_i^R \oplus cb_i^V = H \cdot eb_i^R \oplus H \cdot eb_i^V = H \cdot \left(eb_i^R \oplus eb_i^V\right)$.
   - It is ensured that $eb_i^R \oplus eb_i^V$ has Hamming weight smaller than or equal to t. The same transformation is applied at the enrollment and verification phases for each individual. Therefore, the application of the *DECODE* algorithm to $cb_i^R \oplus cb_i^V$ can recover $eb_i^R \oplus eb_i^V$ because the padding bits from position 1 to $(n-k)$ and from $(n-k+z_i+1)$ to n are 0 s after the XOR application and the Hamming weight of the bits from $(n-k+1)$ to $(n-k+z_i)$, which correspond to $b_i^R \oplus b_i^V$, has a Hamming weight smaller than or equal to t. The simplest solution is to make z_i equal to t because, in this way, the Hamming weight of $b_i^R \oplus b_i^V$ is always smaller than or equal to t. However, biometric features allow for making z_i greater than t, ensuring the substrings b_i meet such a requirement. The latter solution is preferred to reduce the size of communicated and stored data.
   - Given that the substrings eb_i^R and eb_i^V are constructed with a Hamming weight greater than t, *DECODE* algorithm cannot recover eb_i^R and eb_i^V, from cb_i^R and cb_i^V, even with the knowledge of the private key Γ.
3. A distance measurement is applied to $eb_i^R \oplus eb_i^V$, for example, the Hamming distance, by computing the Hamming weight.
4. The sum of the partial distances when all the substrings $eb_i^R \oplus eb_i^V$ are decoded generates the final distance result by computing:

   $$HD = \sum_{i=1}^{N} Hamming\ weight\left(eb_i^R \oplus eb_i^V\right) \quad (1)$$

   The resulting Hamming Distance can be normalized by the number of bits of the binary string b.
5. The final result is compared to a threshold value to determine the recognition decision.

3.3. Proposed Non-Device-Centric Authentication Scheme

We employ the protection proposal described above in a non-device-centric authentication architecture, which considers the phases of setup, enrollment and verification, and, as entities, client devices and database, comparison and authentication servers.

3.3.1. Setup and Enrollment Phases

1. The authentication server A generates its pair of public and private keys $\left(T^A, \Gamma^A\right)$ by means of the Classic McEliece *KeyGen* algorithm. They are stored in a secure way.
2. The authentication server A sends the public key to the client device U and it is stored.
3. A threshold value th is set to determine the recognition decision.
4. The client device U acquires the user identifier ID^U and the biometric samples S^{UR}.
5. Binary biometric features $b^{UR} \in \mathbb{F}_2^l$ that preserve the distances are extracted from S^{UR}.
6. The transformation of the enrollment binary features is applied to b^{UR}, thus, obtaining the substrings $eb_i^{UR} \in \mathbb{F}_2^n$ with Hamming weight greater than t.
7. The *ENCODE* algorithm is employed to encode the substrings eb_i^{UR} by using the authentication server public key T^A, resulting the encoded substrings $cb_i^{UR} \in \mathbb{F}_2^{n-k}$.
8. The ID^U is mapped to an index x^U in a way that is only known by the client device.
9. The substrings cb_i^{UR} and the index x^U are sent to the database server to be stored.

The setup and enrollment phases are illustrated in Figure 4.

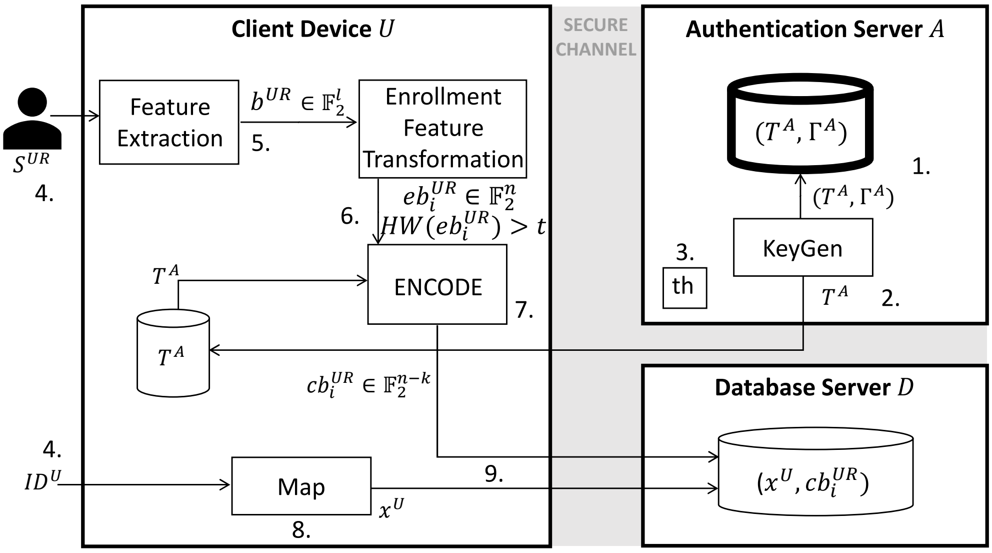


Figure 4. Setup and enrollment phases of the distributed authentication protocol based on the Classic McEliece protection proposed.

3.3.2. Verification Phase

1. The user identifier ID^U and the biometric samples S^{UV} are acquired by the client device U.
2. Binary biometric features $b^{UV} \in \mathbb{F}_2^l$, which preserve the distances, are extracted from S^{UV}.
3. The substrings eb_i^{UV} with Hamming weight greater than t are obtained by applying the same transformation as in the enrollment phase.
4. The *ENCODE* algorithm is employed to encode the substrings eb_i^{UV} by using the authentication server public key T^A, resulting the encoded substrings $cb_i^{UV} \in \mathbb{F}_2^{n-k}$.
5. The client device U generates the nonce $mU_i \in \mathbb{F}_2^{n-k}$.
6. The authentication server A generates the nonce $mA_i \in \mathbb{F}_2^{n-k}$.
7. The encoded substrings are XORed with the nonces generated, obtaining $cbn_i^{UV} \in \mathbb{F}_2^{n-k}$ as $cb_i^{UV} \oplus mU_i \oplus mA_i$.
8. The client device U maps the ID^U to the index x^U.
9. The index x^U and cbn_i^{UV} are sent to the computation server C.
10. The computation server C retrieves the protected template cb_i^{UR} from the database server by using a PIR protocol with x^U as input, that is, $PIR\left(x^U\right)$.
11. The computation server C applies a XOR operation to obtain $cb_i^{UR} \oplus cbn_i^{UV}$.
12. This result is sent to the authentication server A.
13. The authentication server performs the XOR operation $cb_i^{UR} \oplus cbn_i^{UV} \oplus mU_i \oplus mA_i$ to obtain $cb_i^{UR} \oplus cb_i^{UV}$.
14. $eb_i^{UR} \oplus eb_i^{UV}$ is decoded by using the private key Γ^A.
15. The final distance result is computed as in Equation (1).
16. The result is compared with the threshold value th to generate the recognition decision.

Figure 5 illustrates the verification phase.

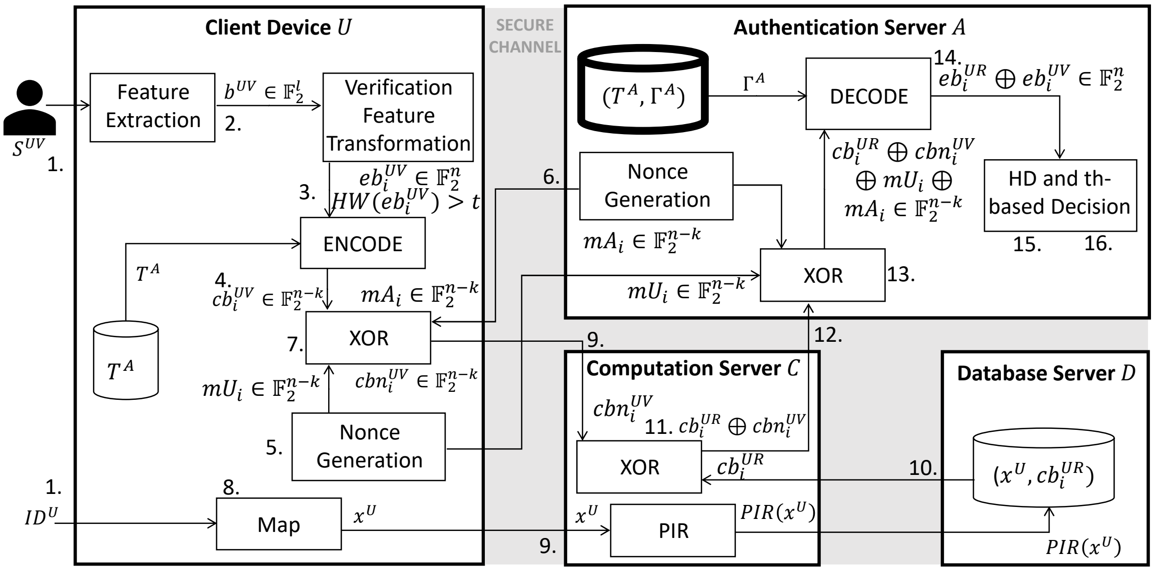

Client Device U
SECURE CHANNEL
Authentication Server A
S^{UV}
1.
Feature Extraction
$b^{UV} \in \mathbb{F}_2^l$
2.
Verification Feature Transformation
$eb_i^{UV} \in \mathbb{F}_2^n$
3. $HW(eb_i^{UV}) > t$
ENCODE
T^A
T^A
4. $cb_i^{UV} \in \mathbb{F}_2^{n-k}$
$mA_i \in \mathbb{F}_2^{n-k}$
7.
XOR
$mU_i \in \mathbb{F}_2^{n-k}$
$cbn_i^{UV} \in \mathbb{F}_2^{n-k}$
5.
Nonce Generation
1.
ID^U
8.
Map
x^U
6.
(T^A, Γ^A)
Γ^A
DECODE
14. $eb_i^{UR} \oplus eb_i^{UV} \in \mathbb{F}_2^n$
Nonce Generation
$cb_i^{UR} \oplus cbn_i^{UV} \oplus mU_i \oplus mA_i \in \mathbb{F}_2^{n-k}$
HD and th-based Decision
15. 16.
$mA_i \in \mathbb{F}_2^{n-k}$
$mU_i \in \mathbb{F}_2^{n-k}$
XOR
13.
9.
12.
Computation Server C
cbn_i^{UV}
11. $cb_i^{UR} \oplus cbn_i^{UV}$
XOR
cb_i^{UR}
10.
Database Server D
(x^U, cb_i^{UR})
x^U
PIR
$PIR(x^U)$
9.
$PIR(x^U)$

Figure 5. Verification phase of the distributed authentication protocol based on the Classic McEliece protection proposed.

3.4. Security Analysis

3.4.1. Biometric Security Requirements

Our proposal satisfies the security requirements (irreversibility, revocability and unlinkability) established in the ISO/IEC 24745 standard on biometric information protection [11].

Irreversibility of the protected templates cb_i^{UR} is satisfied since Classic McEliece is based on the NP problem of decoding random linear codes, Goppa codes able to correct up to t errors are employed and the number of errors in the eb_i^{UR} is larger than t. Hence, the protected templates cannot be decoded either by the authentication server, which knows the private key associated to the public key, employed in the encryption.

Revocability is ensured because a user can remove his/her protected template associated with an authentication server and later create a new one that is different from the previous one. The user can generate different protected templates from the same biometric sample S^{UR} by changing the secret padding $p = (p_1, \ldots, p_N)$. Due to the security provided by the *ENCODE* algorithm of Classic McEliece, the protected templates generated cannot be linked.

Unlinkability of the protected templates associated with several authentication servers, i.e., several services, is also ensured because different public keys of the authentication servers are employed. Therefore, if there was information leakage from the database server(s), it is not possible to distinguish the protected templates that are associated with the same user.

3.4.2. System Attacks

Replay attacks, where an attacker tries to achieve a successful authentication using a previous string, are avoided by including the nonces.

Regarding privacy, if the curiosity of the servers makes them exchange information, the authentication and database servers would only obtain the link between the protected biometric features and the index x^U. The authentication and database servers know if the same or a different user is being authenticated. Anyway, the security of biometric data remains, since the Hamming weight of the protected features is greater than t, so that they cannot be decoded. In addition, they do not know the user identifier ID^U, so that they do not know which particular user is being authenticated.

Similarity-based attacks [12] are possible if the distances between unprotected features are nearly the same as the distances between protected features. A search algorithm is applied to randomly generate first guesses, transform them into protected features, compute the distances with the protected features obtained, use the information to improve the probability of success with new guesses and repeat the process until a successful guess is reached. In order to carry out similarity-based attacks in the proposed scheme, the attacker should steal the client device to be able to generate the secret padding. In addition, the attacker should discover how the bits associated with the biometric data could be modified with the biometric sample provided S^{IV}, which is difficult since the client device is assumed to be resistant to manipulations. In addition, the attacker should break the security of the authentication server to be able to decode, if possible, the distance between plaintexts. Even in that case, since the bits from biometric features can be permuted by the client device, the ordered vector b with the biometric features would be more difficult to extract. In this sense, we consider that our proposal shows enough robustness to similarity-based attacks.

3.4.3. Stolen-Device Scenario

In the stolen-device scenario, an attacker can provide impostor samples S^{IV} at verification. Then, the client device generates the vector eb_i^{IV} from the impostor individual substrings b_i^{IV}, using the padding p_i^U of the genuine user. By using the authentication server public key, the vector cb_i^{IV} is obtained. Nonces mA_i and mU_i are applied and the vector cbn_i^{IV} is generated. The computation server obtains $cb_i^{UR} \oplus cbn_i^{IV}$ and the authentication server obtains $cb_i^{UR} \oplus cb_i^{IV}$ from $cb_i^{UR} \oplus cbn_i^{IV} \oplus mA_i \oplus mU_i$. Since the same padding p_i^U of enrollment is applied, when $eb_i^{UR} \oplus eb_i^{IV}$ is computed, the padding bits are 0. If the Hamming weight of $b_i^{UR} \oplus b_i^{IV}$ is greater than t for some i, $b_i^{UR} \oplus b_i^{IV}$ cannot be decoded and the impostor is rejected. If the Hamming weights of $b_i^{UR} \oplus b_i^{IV}$ for all the i are smaller than or equal to t, the comparison result is the same as in the unprotected scenario, so the attacker is rejected as in the unprotected scenario. The security of the scheme will be the same as the security of the unprotected biometric scheme.

Further, in the stolen-token scenario, the attacker could try to generate synthetic samples S^{IV} at verification to make the client device generate fake b_i^{IV}, such as null vectors or vectors with very low Hamming weight. Then, the attacker should also attack the authentication server to try decoding $b_i^{UR} \oplus b_i^{IV}$ and obtain information about b_i^{UR} because b_i^{IV} is almost null. However, this cannot be performed because the client device checks that the Hamming weights of the substrings b_i^{IV} are neither zero nor have a low value.

4. Practical Realization of the Proposed Biometric Protection Scheme for Facial Recognition

4.1. Implementation of a Non-Device-Centric Facial Authentication

A OnePlus5T smartphone was considered as the client device, which executes an Android App developed in Java. For the computation, authentication and database servers, an Intel Core i5-9400F was considered, which executes Python code in version 3.6.14 for the computation and authentication servers and MySQL for the database server to store the protected templates.

The client device uses the front camera to capture user faces. Then, BlazeFace [36], a convolutional neural network, was applied to detect faces and crop the input image. Subsequently, FaceNet [18], a convolutional neural network to extract floating-point embeddings, was employed as a feature extractor. The load and inference of BlazeFace and FaceNet pre-trained models [37] and [38], respectively, were supported by the library TensorFlowLite, which is suitable for implementations in mobile devices. With these pre-trained models, 128 floating-point feature elements, also referred to as embeddings, were obtained from 160 × 160 bits face images. The floating-point embeddings were discretized by means of Linearly Separable Subcode (LSSC) [35], which was implemented to obtain a binary representation with the codes 000, 001, 011 and 111. The feature space was segmented into four intervals: $(-\infty, -0.1)$, $[-0.1,\ 0.0)$, $[0.0,\ 0.1)$ and $[0.1, +\infty)$, in order to improve the recognition performance. After discretization, each embedding or feature vector was made up of 384 bits.

A shared library in C named as libClassicMcEliece was developed to include the *KeyGen*, *ENCODE* and *DECODE* functions from the Classic McEliece proposal [32] and the *Enrollment Feature Transformation* and *Verification Feature Transformation* functions from our proposal. On the client device, a Java Native Interface (JNI) was employed to allow for running C code in Java. In the computation, authentication and database servers, this functionality was achieved by the use of the libraries *ctypes* and *numpy*.

4.2. Experimental Results

4.2.1. Face Databases

In order to demonstrate the efficiency of our proposal, two public databases were selected: FERET and LFW. Both databases exhibit variations in pose, expression, sex, age and illumination. The FERET database [39], distributed by NIST, was considered because it is a standard database. Since the acquisition conditions in the FERET database were semi-controlled (the same physical setup was employed but it was reassembled for each session), the LFW database was also considered. The LFW database [40] was designed for unconstrained face recognition by using images collected from the web. Since, during the face detection and crop process, some samples are lost, the extracted embeddings were 8160 of 994 individuals in FERET and 12,770 embeddings of 5749 individuals in LFW databases.

4.2.2. Recognition Performance

For the evaluation of the recognition performance, we followed the FVC (Fingerprint Verification Competition) protocol [41]. For the genuine comparisons, each sample of the same individual is compared to the remaining samples of the same individual. For the impostor comparisons, the first sample of each individual is compared with the first sample of the remaining individuals. Symmetric comparisons are removed to avoid correlation. In the FERET database, the number of genuine comparisons was 52,708 and the number of impostor comparisons was 483,636. In the LFW database, the number of genuine comparisons was 235,254 and the number of impostor comparisons was 15,487,395.

The recognition performance in the protected domain was evaluated by considering the Classic McEliece parameter sets that provide security levels of 128 and 256 bits. These parameter sets are mceliece348864 (with $n = 3488$, $t = 64$, $m = 12$, and $k = 2720$) and mceliece6688128 (with $n = 6688$, $t = 128$, $m = 13$, and $k = 5024$), respectively. Embeddings of 384 bits (48 bytes) were divided into two 192-bit segments for mceliece348864 and into one 384-bit segment for mceliece6688128. Therefore, $z_i = 192$ for mceliece348864 and $z_i = 384$ for mceliece6688128. The padding was considered to have Hamming weight $t + 1$ in order to ensure that the Hamming weight of the substrings eb_i was greater than t. In this way, paddings with 65 1s were generated for mceliece348864 and paddings with 129 1s were generated for mceliece6688128.

The recognition performance was evaluated in terms of accuracy, defined as the ratio in percentage between the number of true recognition decisions and the total number of recognition decisions, False-Acceptance Rate (FAR) and False-Rejection Rate (FRR). Table 1 shows the recognition performance of FaceNet embeddings represented in floating-point and binary in the unprotected domain. Comparisons of floating-point embeddings were performed by using the Euclidean distance. Comparisons of binary embeddings were performed by using the Hamming distance. Although some information is lost by the binarization process, accuracy, FAR and FRR are not largely affected. Table 1 also includes the recognition performance when the proposed protection based on Classic McEliece is applied. The accuracy, FAR and FRR results prove that the recognition performance was improved. For the genuine distribution, the same transformation (same secret padding and same random positions) was applied for embeddings from different samples from the same individual. Then, the *DECODE* algorithm could recover $b_i^R \oplus b_i^V$ from $cb_i^R \oplus cb_i^V$ when the Hamming weight of $eb_i^R \oplus eb_i^V$ was smaller than or equal to t. For the impostor distribution, different transformations (different secret padding and different random positions) were applied for embeddings of different individuals. Then, the *DECODE* algorithm could not decode $cb_i^R \oplus cb_i^V$ because $eb_i^R \oplus eb_i^V$ had Hamming weight greater than t. For the accuracy indicator, true-positive decisions were obtained from genuine comparisons that could be decoded (impostor comparisons were always true-negative decisions). In the LFW database, almost all the genuine comparisons were decoded. For this reason, the accuracy was approximately 100%. FAR was 0% because impostor comparisons generated decoding fails. FRR was the same as in the unprotected domain because all genuine comparisons could be decoded so the recognition performance was not affected.

Table 1. Recognition performance of FaceNet embeddings.

Database	Representation	Protection	Accuracy (%)	FAR (%)	FRR (%)
FERET	Floating-point	No	99.2	1.15	1.15
	Binary	No	98.9	1.69	1.69
	Binary	mceliece348864	99.8	0	1.69
	Binary	mceliece6688128	99.8	0	1.69
LFW	Floating-point	No	99.3	0.83	0.83
	Binary	No	99.2	1.18	1.18
	Binary	mceliece348864	~100	0	1.18
	Binary	mceliece6688128	~100	0	1.18

4.2.3. Size and Execution Time Performance

Table 2 shows the size performance of our proposals based on Classic McEliece mceliece348864 and mceliece6688128 parameter sets compared to other post-quantum biometric protection schemes from the literature. These proposals are based on a combination of homomorphic encryption and lattice-based cryptography. Our proposal was the best one in terms of the ratio between protected and unprotected feature sizes. The last row illustrates the total size of protected features for 1000 users. This proves that the communication and storage overheads of protected features with our proposal are lower.

Table 2. Size performance comparison of post-quantum biometric protection proposals based on homomorphic encryption.

Proposal	Ideal Lattices [8]	R-LWE [8]	R-LWE [9]	M-LWE [10]	M-LWR [10]	mceliece348864 (Ours)	mceliece6688128 (Ours)
Security Level (bits)	80	80	128	128	256	128	256
Unprotected Feature Size (bytes)	256	256	48	48	48	48	48
Protected Feature Size (bytes)	19,456	31,744	5632	2181	3133	192	208
Protected Feature Size/ Unprotected Feature Size	76.0	124.0	117.3	45.4	62.3	4.0	4.3
Total Feature Size of 1000 Users (Kbytes)	19,000	31,000	5500	2130	3060	187.5	203.1

Table 3 shows the execution times required by the operations of mceliece348864 and mceliece6688128 parameter sets in the platforms described in Section 4.1. These results prove that the authentication of a user can be carried out in real time.

Table 3. Execution times (ms) for the authentication operations.

Operations	mceliece348864	mceliece6688128
Feature extraction, transformation and ENCODE	25.3	55.7
XOR, DECODE and threshold comparison	36.5	85.0
Total	61.8	140.7

4.2.4. Security Performance

The framework proposed in [42] was applied to evaluate unlinkability and revocability. Unlinkability is evaluated by considering the distributions of mated and non-mated instances. Revocability is evaluated by considering the distributions of the same sample and non-mated instances. Distributions of mated instances are generated by genuine comparisons of protected features from different samples from the same instance created by using different feature transformations (different embeddings from the same individual, different secret paddings and different random positions). Distributions of non-mated instances are generated by impostor comparisons of protected features from different instances created by using different feature transformations (different embeddings from different individuals, different secret paddings and different random positions). Distributions of same samples are generated by genuine comparisons of protected features from the same sample created by using different feature transformations (same sample, different secret paddings and different random positions). These distributions must coincide to prove unlinkability and revocability scenarios. Figure 6 shows that unlinkability and revocability were satisfied for the distributions in the LFW database by considering protection based on the Classic McEliece mceliece6688128 parameter set (similar results were obtained with mceliece348864 parameter set). The LFW database was selected for this evaluation because the numbers of samples and comparisons are higher.

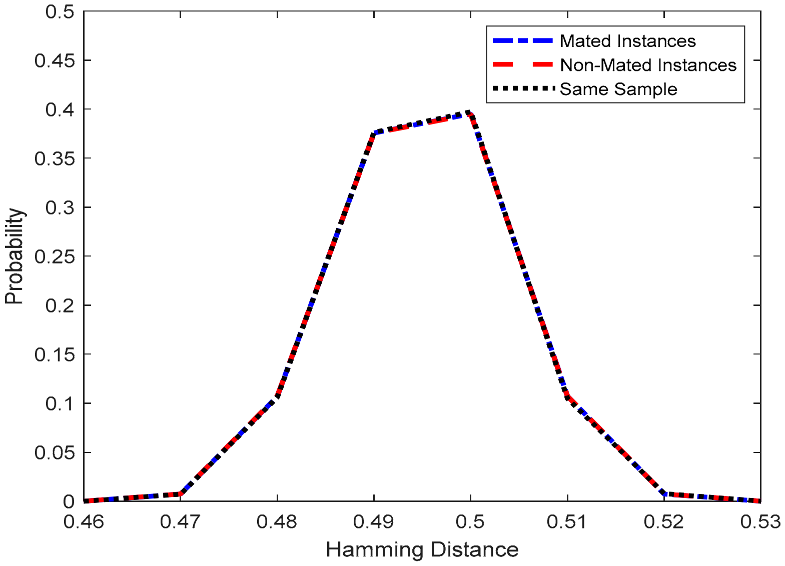


Figure 6. Revocability and unlinkability evaluation for the LFW database with Classic McEliece mceliece6688128 parameter set.

The resistance to similarity-based attacks in case protected features could be accessed was evaluated using the method described in [12]. This method considers that protected biometric features are secure if there is no correlation between the distances of the impostor comparisons in the protected and in the unprotected domains. The results obtained for the LFW database by considering protection based on the mceliece6688128 parameter set (similar results were obtained with mceliece348864 parameter set) are shown in Figure 7.

This figure depicts the distances of the impostor comparisons in the protected domain on the vertical axis and the distances of the impostor comparisons in the unprotected domain on the horizontal axis. The figure illustrates that the impostor-protected distances do not change with respect to their impostor-unprotected distances. Therefore, their correlation is quite small, which proves the resistance to similarity-based attacks of our proposal.

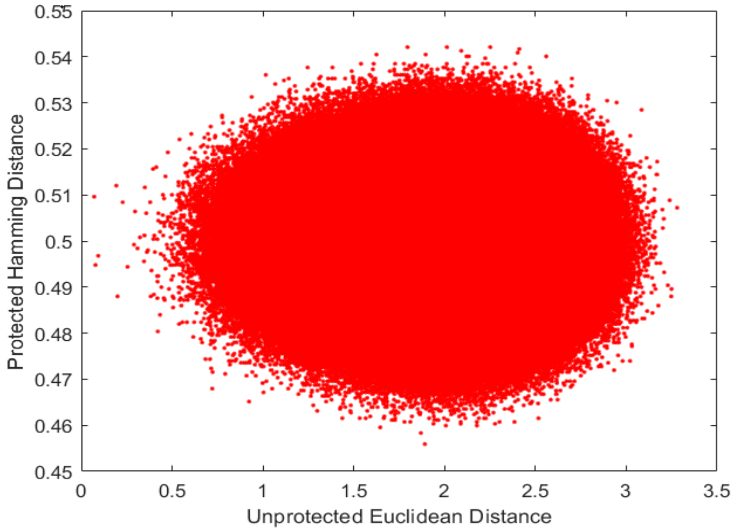


Figure 7. Evaluation of the resistance to similarity-based attacks for the LFW database with Classic McEliece mceliece6688128 parameter set.

Another issue to consider is the stolen-device scenario. The attacker employs his/her biometrics and the client device of a genuine user extracts the associated embeddings. These embeddings are transformed with the same secret padding and random positions employed to enroll the genuine user. In this scenario, as illustrated in Figure 8 with the comparison of FAR and FRR (DET curve) for the LFW database by considering protection based on the mceliece6688128 parameter set (similar results were obtained with mceliece348864 parameter set), the recognition performance was almost the same as in the unprotected approach.

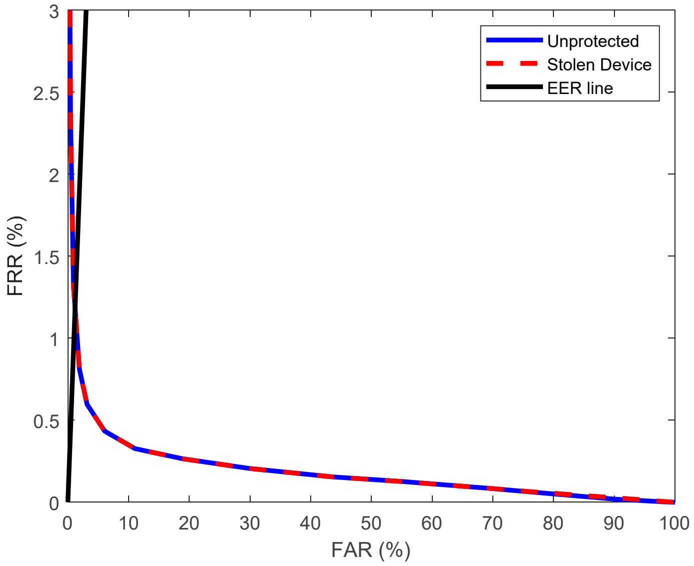


Figure 8. DET Curves for the unprotected and the stolen device scenario obtained from LFW database with Classic McEliece mceliece6688128 parameter set.

5. Discussion

In this work, we proposed a biometric protection scheme based on homomorphic encryption and Classic McEliece as well as its use in a non-device-centric biometric authentication architecture with honest-but-curious servers. Classic McEliece is one of the round 4 candidates of the NIST post-quantum cryptography standardization competition.

A specific transformation of features is proposed to meet security and Classic McEliece requirements. This precludes decoding-protected biometric features and impostor comparison results, even knowing the private key and even the servers exchanging information. Therefore, in an honest-but-curious adversarial model, the security is maintained and recognition performance is improved in a normal scenario and preserved in a stolen-device scenario. Parameter sets were selected to achieve security levels of 128 and 256 bits for Classic McEliece (mceliece6688128 and mceliece348864 parameter sets, respectively). Irreversibility is ensured by the impossibility of decoding ciphertexts with more errors than those permitted. Further, Classic McEliece-based protected features are random, even though they are generated from the same sample and, thus, revocability and unlinkability are satisfied. In addition, the scheme is robust to FAR and similarity-based attacks.

A practical realization is illustrated, which was performed by considering a smartphone as the device client and computers as the database, computation and authentication servers. FaceNet embeddings were selected as biometric features. The execution times obtained allow for real-time authentication. A relevant advantage of our proposal is that Classic McEliece generates protected data with sizes much lower than other approaches, while maintaining low computational cost.

6. Conclusions

In summary, the proposed non-device-centric biometric authentication scheme offers the following advantages:

- Post-quantum security, even with honest-but-curious servers;
- Privacy-preserving management of individual data;
- Recognition performance improved in a normal scenario and maintained in a stolen-device scenario;
- Practical realizations allowing for real-time authentication with low computational, storage and communication costs.

Author Contributions: Conceptualization, all authors; methodology, R.A. and I.B.; software, P.L.-G.; validation, R.A. and I.B.; formal analysis, all authors; investigation, all authors; resources, P.L.-G.; data curation, all authors; writing—original draft preparation, R.A.; writing—review, all authors; writing—editing, R.A. and I.B.; visualization, R.A.; supervision, R.A. and I.B.; project administration, I.B. and R.A.; funding acquisition, I.B., R.A. and R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted thanks to Grant PDC2021-121589-I00 funded by MCIN/AEI/ 10.13039/501100011033 and the "European Union NextGenerationEU/PRTR", and Grant PID2020-119397RB-I00 funded by MCIN/AEI/ 10.13039/501100011033. The work of Roberto Román was supported by VI Plan Propio de Investigación y Transferencia through the University of Seville.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Jain, A.K.; Flynn, P.; Ross, A.A. *Handbook of Biometrics*; Springer: New York, NY, USA, 2010.
2. Rane, S.; Wang, Y.; Draper, S.C.; Ishwar, P. Secure Biometrics: Concepts, Authentication Architectures, and Challenges. *IEEE Signal Process. Mag.* **2013**, *30*, 51–64. [CrossRef]
3. Simoens, K.; Bringer, J.; Chabanne, H.; Seys, S. A Framework for Analyzing Template Security and Privacy in Biometric Authentication Systems. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 833–841. [CrossRef]
4. Gómez-Barrero, M.; Maiorana, E.; Galbally, J.; Campisi, P.; Fierrez, J. Multi-biometric template protection based on homomorphic encryption. *Pattern Recognit.* **2017**, *67*, 149–163. [CrossRef]
5. Morampudi, M.K.; Prasad, M.V.N.K.; Raju, U.S.N. Privacy-preserving iris authentication using fully homomorphic encryption. *Multimed. Tools Appl.* **2020**, *79*, 19215–19237. [CrossRef]
6. Vallabhadas, D.K.; Sandhya, M. Securing multimodal biometric template using local random projection and homomorphic encryption. *J. Inf. Secur. Appl.* **2022**, *70*, 103339. [CrossRef]
7. Torres, W.A.A.; Bhattacharjee, N.; Srinivasan, B. Privacy-preserving biometrics authentication systems using fully homomorphic encryption. *Int. J. Pervasive Comput. Commun.* **2015**, *11*, 151–168. [CrossRef]
8. Yasuda, M. Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption. *Inf. Sec. J. Glob. Perspect.* **2017**, *26*, 85–103. [CrossRef]
9. Kolberg, J.; Drozdowski, P.; Gomez-Barrero, M.; Rathgeb, C.; Busch, C. Efficiency Analysis of Post-quantum-secure Face Template Protection Schemes based on Homomorphic Encryption. In Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 16–18 September 2020.
10. Román, R.; Arjona, R.; Baturone, I. A Quantum-Resistant Face Template Protection Scheme using Kyber and Saber Public Key Encryption Algorithms. In Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 14–16 September 2022.
11. *ISO/IEC 24745:2022*; Information Security, Cybersecurity and Privacy Protection—Biometric Information Protection. ISO: Geneva, Switzerland, 2022.
12. Chen, Y.; Wo, Y.; Xie, R.; Wu, C.; Han, G. Deep Secure Quantization: On secure biometric hashing against similarity-based attacks. *Signal Process.* **2019**, *154*, 314323. [CrossRef]
13. Nandakumar, K.; Jain, A.K. Biometric Template Protection: Bridging the performance gap between theory and practice. *IEEE Signal Process. Mag.* **2015**, *32*, 88100. [CrossRef]
14. Al-Saggaf, A.A.A. Post-Quantum Fuzzy Commitment Scheme for Biometric Template Protection: An Experimental Study. *IEEE Access* **2021**, *9*, 110952–110961. [CrossRef]
15. Arjona, R.; Baturone, I. A Post-Quantum Biometric Template Protection Scheme Based on Learning Parity With Noise (LPN) Commitments. *IEEE Access* **2020**, *8*, 182355–182365. [CrossRef]
16. NIST Post-Quantum Cryptography Round 4 Submissions. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions (accessed on 21 December 2022).
17. NIST IR 8413. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. July 2022. Available online: https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413.pdf (accessed on 21 December 2022).
18. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
19. Juels, A.; Wattenberg, M. A Fuzzy Commitment Scheme. In Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS), Singapore, 1 November 1999.
20. Kelkboom, E.J.C.; Breebaart, J.; Kevenaar, T.A.M.; Buhan, I.; Veldhuis, R.N.J. Preventing the decodability attack based crossmatching in a fuzzy commitment scheme. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 107–121. [CrossRef]
21. Murakami, T.; Ohki, T.; Takahashi, K. Optimal sequential fusion for multibiometric cryptosystems. *Inf. Fusion* **2016**, *32*, 93108. [CrossRef]
22. Abidin, A.; Argones-Rúa, E.; Preneel, B. An efficient entity authentication protocol with enhanced security and privacy properties. In Proceedings of the International Conference on Cryptology and Network Security (CANS), Dubai, United Arab Emirates, 13–16 November 2016; Springer: Cham, Switzerland, 2016; Volume 10052, pp. 1–16.
23. Arjona, R.; Prada-Delgado, M.Á.; Baturone, I.; Ross, A. Securing Minutia Cylinder Codes for Fingerprints through Physically Unclonable Functions: An Exploratory Study. In Proceedings of the IEEE International Conference on Biometrics (ICB), Gold Coast, Australia, 20–23 February 2018.
24. NIST Post-Quantum Cryptography Selected Algorithms. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022 (accessed on 21 December 2022).
25. McEliece, R.J. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *The Deep Space Network Progress Report*. 1978; pp. 114–116. Available online: https://tmo.jpl.nasa.gov/progress_report2/42-44/44N.PDF (accessed on 21 December 2022).
26. Chabanne, H. Method for Coding Biometric Data, Method for Controlling Identity and Devices for Carrying out Said Methods. U.S. Patent US7797606B2, 20 January 2016.
27. Sharma, A.; Ojha, D.B. Biometric Template Security Using Code Base Cryptosystem. *Inf. Secur. Int. J.* **2013**, *26*, 49–60.

28. Kuznetsov, A.; Kiyan, A.; Uvarova, A.; Serhiienko, R.; Smirnov, V. New Code Based Fuzzy Extractor for Biometric Cryptography. In Proceedings of the International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 9–12 October 2018.
29. Singh, H. Code based Cryptography: Classic McEliece. *arXiv* **2019**, arXiv:1907.12754.
30. Lee, Y.; Cho, J.; Kim, Y.-S.; No, J.-S. Cryptanalysis of the Ivanov-Kabatiansky-Krouk-Rumenko Cryptosystems. *IEEE Commun. Lett.* **2020**, *24*, 2678–2681. [CrossRef]
31. Li, Y.X.; Deng, R.H.; Wang, X.M. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Trans. Inf. Theory* **1994**, *40*, 271–273.
32. Classic McEliece NIST Proposal. Available online: https://classic.mceliece.org/ (accessed on 21 December 2022).
33. Abidin, A.; Mitrokotsa, A. Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-LWE. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Atlanta, GA, USA, 3–5 December 2014.
34. Chor, B.; Kushilevitz, E.; Goldreich, O.; Sudan, M. Private information retrieval. *J. ACM* **1998**, *45*, 965–981. [CrossRef]
35. Lim, M.; Teoh, A.B.J. A Novel Encoding Scheme for Effective Biometric Discretization: Linearly Separable Subcode. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 300–313. [CrossRef]
36. Bazarevsky, V.; Kartynnik, Y.; Vakunov, A.; Raveendran, K.; Grundmann, M. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs. *arXiv* **2019**, arXiv:1907.05047.
37. BlazeFace Model. Available online: https://github.com/tensorflow/tfjs-models/tree/master/blazeface (accessed on 21 December 2022).
38. David Sanberg's Facenet Model. Available online: https://drive.google.com/file/d/0B5MzpY9kBtDVZ2RpVDYwWmxoSUk/edit?resourcekey=0-xi62SLMG3gMyC6wTkl9Q0A (accessed on 21 December 2022).
39. Phillips, P.J.; Moon, H.; Rizvi, S.A.; Rauss, P.J. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1090–1104. [CrossRef]
40. Huang, G.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, 17 October 2008.
41. Fingerprint Verification Competition (FVC). Available online: https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx (accessed on 21 December 2022).
42. Gomez-Barrero, M.; Galbally, J.; Morales, A.; Fierrez, J. Privacy-Preserving Comparison of Variable-Length Data with Application to Biometric Template Protection. *IEEE Access* **2017**, *5*, 8606–8619. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.