Deep Learning-based Biometric Cryptographic Key Generation with Post-Quantum Security

Oleksandr Kuznetsov

kuznetsov@karazin.ua

Department of Political Sciences, Communication and International Relations, University of Macerata, Macerata, Via Crescimbeni, 30/32, Macerata, 62100, MC, Italy

Dmytro Zakharov

zamdmytro@gmail.com

Department of Applied Mathematics, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine

Emanuele Frontoni

emanuele.frontoni@unimc.it

Department of Political Sciences, Communication and International Relations, University of Macerata, Macerata, Via Crescimbeni, 30/32, Macerata, 62100, MC, Italy

Research Article

Keywords: cryptographic keys, deep learning models, convolutional neural networks, fuzzy extractor, biometric face images, code-based cryptosystems

Posted Date: May 10th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2913502/v1

Version of Record: A version of this preprint was published at Multimedia Tools and Applications on December 14th, 2023. See the published version at https://doi.org/10.1007/s11042-023-17714-7.

Deep Learning-based Biometric Cryptographic Key Generation with Post-Quantum Security

Alexandr Kuznetsov[1,2*†], Dmytro Zakharov[2†] and Emanuele Frontoni[1,3†]

[1*]Department of Political Sciences, Communication and International Relations, University of Macerata, Macerata, Via Crescimbeni, 30/32, Macerata, 62100, MC, Italy.
[2]Department of Applied Mathematics, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine.
[3]Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, Ancona, 60131, AN, Italy.

*Corresponding author(s). E-mail(s): kuznetsov@karazin.ua;
Contributing authors: zamdmytro@gmail.com;
emanuele.frontoni@unimc.it;
†These authors contributed equally to this work.

Abstract

In contemporary digital security systems, the generation and management of cryptographic keys, such as passwords and pin codes, often rely on stochastic random processes and intricate mathematical transformations. While these keys ensure robust security, their storage and distribution necessitate sophisticated and costly mechanisms. This study explores an alternative approach that leverages biometric data for generating cryptographic keys, thereby eliminating the need for complex storage and distribution processes. The paper investigates biometric key generation technologies based on deep learning models, specifically utilizing convolutional neural networks to extract biometric features from human facial images. Subsequently, code-based cryptographic extractors are employed to process the primary extracted features. The performance of various deep learning models and the extractor is evaluated by considering Type 1 and Type 2 errors. The optimized algorithm parameters yield an error rate of less than **10%**, rendering the generated keys suitable for biometric authentication. Additionally, this study demonstrates that the application of code-based cryptographic extractors provides a post-quantum level of security, further enhancing the practicality and effectiveness of biometric key generation technologies in modern information security systems. This research contributes to the ongoing efforts towards secure, efficient, and user-friendly authentication and encryption methods, harnessing the power of biometric data and deep learning techniques.

Keywords: cryptographic keys, deep learning models, convolutional neural networks, fuzzy extractor, biometric face images, code-based cryptosystems

1 Introduction

The rapid advancements in technology and the growing reliance on digital systems for various day-to-day activities have led to an increasing demand for secure and user-friendly authentication mechanisms. One of the most promising directions in this regard is the application of biometric technologies, which are quickly gaining traction in multiple domains of computer science [1–3]. Biometrics utilizes unique physiological or behavioral characteristics, such as facial features, fingerprints, and iris patterns, to establish an individual's identity with high accuracy [4, 5].

The appeal of biometrics lies in its inherent security, as these unique traits are difficult to forge or reproduce, and its user-friendliness, as users do not need to remember complex passwords or carry additional tokens for authentication [4]. Consequently, biometric technologies have been integrated into a diverse array of applications, ranging from access control in enterprises and criminal identification by law enforcement agencies to personalized services in medicine, banking, and advertising [3, 5].

However, the widespread adoption of biometrics has also given rise to new challenges and concerns. Most biometric systems are heavily reliant on large databases containing an extensive collection of images [6–8], including fingerprints, facial photos, iris scans, and more. The storage and management of such massive amounts of personal data introduce significant risks, such as data breaches and malicious use of stolen biometric information [9–11]. These issues highlight the need for innovative solutions that address the shortcomings of traditional biometric authentication and personal identification methods.

In parallel, cryptographic key-based access control methods, employing passwords, pin codes, and other secret keys [12, 13], have long been used to secure digital systems. These cryptographic keys are generated using specialized algorithms that rely on stochastic random processes and complex mathematical transformations [14–17]. While offering a robust security mechanism, the use of cryptographic keys introduces the need for secure storage and distribution systems, which can be expensive and cumbersome to maintain. Additionally, the loss of a cryptographic key could lead to the loss of access to critical resources, such as bank accounts or customer service accounts.

The advent of quantum computing has further intensified the need for advanced cryptographic methods, as traditional encryption techniques may become vulnerable to attacks by powerful quantum computers. Post-quantum cryptography seeks to develop cryptographic algorithms that are secure against both classical and quantum computational attacks [18, 19]. One promising direction in post-quantum cryptography is the use of code-based cryptosystems, which are known to be resistant to quantum attacks.

In light of the aforementioned challenges, this paper explores an approach that combines the advantages of biometrics and cryptographic keys while incorporating post-quantum cryptographic security. We propose a novel method for generating cryptographic keys from biometric facial images using deep learning models and code-based cryptographic extractors. By deriving cryptographic keys directly from biometric data, we eliminate the need for storing and distributing secret keys, while maintaining the security and user-friendliness of biometric authentication.

Furthermore, our approach employs code-based cryptographic extractors to process the primary extracted biometric features, ensuring a post-quantum level of security. We experiment with various deep learning models and assess the performance of the extractor based on Type 1 (False Reject Rate, FRR) and Type 2 (False Accept Rate, FAR) errors. The optimized parameters of our algorithms yield an FRR and FAR of less than 10%, enabling the use of the generated keys for biometric authentication.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of related studies on biometric key generation, cryptographic extractors, and post-quantum cryptography. Section 3 discusses the feature vector extractors employed in our methodology. Section 4 presents the feature vector converter for processing the extracted features. Section 5 details the code-based fuzzy extractor, which ensures post-quantum security. Section 6 reports the experimental results, while Section 7 compares our approach with existing methods. Finally, Section 8 concludes the paper and discusses potential avenues for future research in this area.

2 Related studies

This paper explores new technologies for generating cryptographic keys from biometric images using deep learning methods. This direction is developed in many related works. For example, in [1, 8, 20], the issues of liveness detection and user authentication are studied. In [9, 21–23] and many others, AI methods are examined in relation to solving the problem of biometric cryptography. The works [24–27] explore fuzzy extractors, which are developed on the basis of the previous *Fuzzy Commitment* [28–30] and *Fuzzy Vault Schemes* [31–33].

Key generation issues are considered in [34–36]. In [37], it was proposed to use a code-based extractor to generate keys. This is one of the areas of code-based cryptography [38–42], which, as shown in [18, 19, 38], provides high resistance to both classical and quantum cryptoanalysis. Code-based cryptography is considered a reliable and secure alternative to modern public-key cryptosystems [43–45]; some algorithms have already been selected by NIST USA for post-quantum standardization [46].

Thus, a cryptographic extractor based on codes is a promising direction for further research [4, 47]. At the same time, to use such an extractor, the initial biometric data must be converted into binary strings with a low error rate (no more than 25%). Errors can occur for various reasons, for example, due to errors in the processing of biometric images or due to deliberate distortion. To ensure low FRR values, one must minimize the error rate in the binary strings of a single user. Biometric images of another person (for example, an intruder) are also converted into binary strings with errors. To ensure a low FAR value, the error rate for the biometrics of different users must be very high. Thus, the central gap in modern research is estimating the frequency of errors in binary strings of different users. In this work, we investigate this problem. We use different deep-learning models to generate binary strings and estimate the error probabilities FRR and FAR before and after applying the code-based extractor. We optimize the extractor parameters based on the codes to provide small FRR and FAR. Our experiments with biometric facial images confirm this.

3 Feature vector extractors

In our research, we restrict ourselves to biometric images of faces and explore techniques for generating strong keys. We explore various deep-learning techniques for extracting biometric features. In particular, we consider one of the most accurate models *Keras Facenet* [48] and *Face Recognition* [49] with accuracy 99.63% and 99.38%, respectively. We explore the possibilities of using them to generate binary strings.

3.1 Feature vector extraction

Keras Facenet and *Face Recognition* feature extraction algorithms, given an image X, return a real-valued feature vector $\boldsymbol{f} = \phi(X) \in \mathbb{R}^{n_f}$ of fixed size n_f (here and hereafter by $\phi : \mathcal{I} \to \mathbb{R}^{n_f}$ we denote this transformation from an image to a feature vector).

Feature vectors of two similar people must be "close" to each other while feature vectors of two different people must be "far away" from each other. There are different ways how to encapsulate this rule strictly, but in case of *Keras Facenet* and *Face Recognition* packages the *vector distance* function $d_v(\cdot, \cdot)$ is considered:

$$d_v(\boldsymbol{x}, \boldsymbol{y}) \triangleq \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

This way, if we have two images of people, say, X and Y, then if $d_v(\phi(X), \phi(Y))$ is small enough than most likely X and Y are images of the same person, whereas if $d_v(\phi(X), \phi(Y))$ is large enough, we conclude that X and Y are images of two different people.

Therefore, it is relevant to introduce the term *distance between images* as defined in paper [20] for the *Siamese neural network* (where each vector element f_k has the same weight). The distance $d_{\mathcal{I}}(X, Y)$ between images $X, Y \in \mathcal{I}$ is a vector distance between the corresponding feature vectors, i.e.:

$$d_{\mathcal{I}}(X, Y) \triangleq d_v(\phi(X), \phi(Y)) \tag{1}$$

As said before, both *Keras Facenet* and *Face Recognition* use deep learning approach: firstly the set of triplets $\mathcal{T} \subset \mathcal{I}^3$ is being formed where each element has a form $\{A, P, N\}$, with A and P being the images of the same person, and A with N

of two different people (here letters A, P, N correspond to the terms *anchor*, *positive*, and *negative* which represent aforementioned relations). Then the algorithm tries to find such parameters of the neural network to minimize the following loss function (where β is a positive hyperparameter):

$$\hat{\mathcal{L}} = \sum_{A,P,N \in \mathcal{T}} \operatorname{ReLU}\left(d_{\mathcal{I}}(A, P) - d_{\mathcal{I}}(A, N) + \beta\right).$$

In particular, this distance helps us to differentiate when two images correspond to a single person (we denote such relation as $X \equiv Y$) and when to two different people (we denote such relation as $X \not\equiv Y$).

Consider the Figure 1. On this figure pairs of images are displayed with a corresponding distance between them. As one might see, when 2 images correspond to two different people, the distance between them is much larger than in case when two images correspond to a single person.

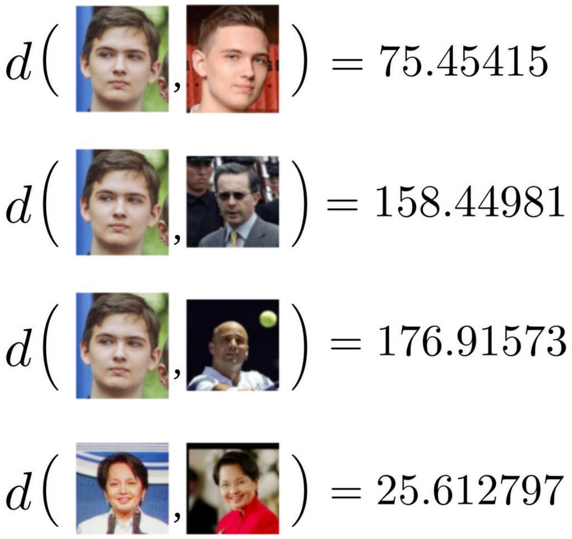
$d\left(\cdot , \cdot \right) = 75.45415$
$d\left(\cdot , \cdot \right) = 158.44981$
$d\left(\cdot , \cdot \right) = 176.91573$
$d\left(\cdot , \cdot \right) = 25.612797$

Fig. 1 Distances between 4 pairs of images. Images (besides the author's ones) are taken from the *CelebA dataset*[26]

Speaking more strictly, suppose we have certain hyperparameter τ, which characterizes the boundary between classification 'single person' and 'different people', that is:

- $d_{\mathcal{I}}(X, Y) \leq \tau \implies X \equiv Y$
- $d_{\mathcal{I}}(X, Y) > \tau \implies X \not\equiv Y$

For instance, if we put $\tau = 100$ in the example above, the first and the fourth pairs will be identified as the same person, but the second and third ones as different people, which would correspond to the correct result.

3.2 Binary strings extraction

However, as mentioned in the section 1, we need to form not a real-valued vector, but a binary string $s \in \Sigma^{n_s}$ of a fixed length $n_s = 128$ (here and hereafter, $\Sigma \equiv \{0, 1\}$).

In our research we decided, given a vector $\boldsymbol{f} = \phi(X)$, extracted from the image $X \in \mathcal{I}$, using existing feature extractors, form the binary string s directly, without changing used neural network parameters. In other words, we will apply a certain function which we call *feature vector converter* $\psi : \mathbb{R}^{n_f} \to \Sigma^{n_s}$ on vector $\boldsymbol{f}$, yielding the binary string $s = \psi(\boldsymbol{f})$.

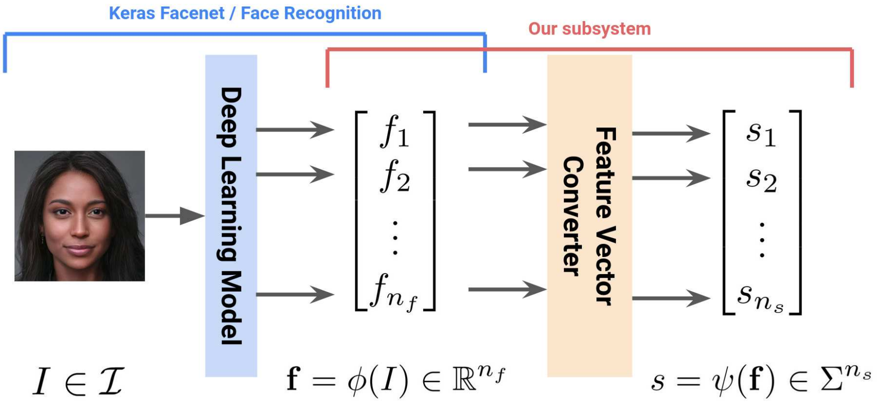
Keras Facenet / Face Recognition
Our subsystem
Deep Learning Model
f_1
f_2
f_{n_f}
Feature Vector Converter
s_1
s_2
s_{n_s}
$I \in \mathcal{I}$
$\mathbf{f} = \phi(I) \in \mathbb{R}^{n_f}$
$s = \psi(\mathbf{f}) \in \Sigma^{n_s}$

Fig. 2 Structure of our system as a whole: firstly applying transformation ϕ from an image to a real-valued vector and then applying ψ, yielding a binary string.

All things considered, we retrieve the composition of functions ϕ and ψ, yielding the desired function $\Psi : \mathbb{R}^{n_f} \to \Sigma^{n_s}$ from an image to a binary string: $\Psi = \psi \circ \phi$.

3.3 Dataset for the feature extractor evaluation

For a feature vector extractor accuracy evaluation, we will use [50, 51] datasets.

At first, we split the images into batches where in each batch only the images of one single person are stored. That way, if we take 2 images from the same batch, we will have the images of a same person, whereas if 2 images are taken from 2 different batches, we'll have images of different people.

Binary strings extracted from images of the same batch should differ by no more than 25%. The frequency of mismatches (error rate) of elements' binary strings from one batch is decisive for the calculation of FRR.

Binary strings extracted from images of different batches should differ as much as possible. The mismatch frequency (error rate) of elements' binary strings from different batches is decisive for the FAR calculation.

Apart from splitting the images into batches, for the subsequent accuracy evaluation we would also need to split images into pairs. We will denote such a set of pairs as $\mathcal{P} \subset \mathcal{I} \times \mathcal{I}$.

4 Feature vector converter

As mentioned in the subsection 3.2, feature vector converter should, given a real-valued feature vector $\boldsymbol{f} \in \mathbb{R}^{n_f}$, form a binary string $s \in \Sigma^{n_s}$ of length n_s. Since in our case $n_f = n_s = 128 =: N$, that significantly simplifies the task.

Suppose we have a vector $\boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \in \mathbb{R}^N$ and we need to form $s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix} \in \Sigma^N$ according to some rule $\psi : \mathbb{R}^N \to \Sigma^N$. Let us define ψ in the following way:

$$\psi \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} \mathbb{1}(f_1 > 0) \\ \mathbb{1}(f_2 > 0) \\ \vdots \\ \mathbb{1}(f_N > 0) \end{bmatrix} =: \mathring{\mathbb{1}}(\boldsymbol{f} > \mathbf{0}_N), \tag{2}$$

where we denoted by $\mathring{\mathbb{1}}(\boldsymbol{a} > \boldsymbol{b})$ a vector that has 1 on i^{th} position if $a_i > b_i$ and 0 otherwise.

This result yielded relatively good results since, in average, feature vectors tend to be distributes around $\mathbf{0}_N$. However, one might improve results by applying a simple adjustment.

To illustrate an issue with the definition 2, consider Figure 3 with $N = 2$ for simplicity. If the data is distributed as it is shown on the left, function $\mathring{\mathbb{1}}(\mathbf{x} > \mathbf{0}_N)$ yields high accuracy as it distinguishes all the clusters with different people. However, applying the same function to the dataset shown on the right would not distinguish any pairs since all of them are located in the same quadrant. To prevent this issue, we could shift the axes to the expected value of this distribution as a whole.

Suppose that $\mathbf{X}$ is a random vector with an expectation value of $\boldsymbol{\mu} := \mathbb{E}[\mathbf{X}]$. Thus, we define the new feature vector converter as

$$\psi(\mathbf{x}) = \mathring{\mathbb{1}}(\mathbf{x} > \boldsymbol{\mu}) \tag{3}$$

Given the dataset $\mathcal{D} = \{I_k, y_k\}_{k=0}^{n_D}$, we retrieve a set of feature vectors $\{\phi(I_k)\}_{k=1}^{n_D}$ and approximate the expectation value as simply this set's mean, that is $\boldsymbol{\mu} \approx \frac{1}{n_D} \sum_{k=1}^{n_D} \phi(I_k)$.

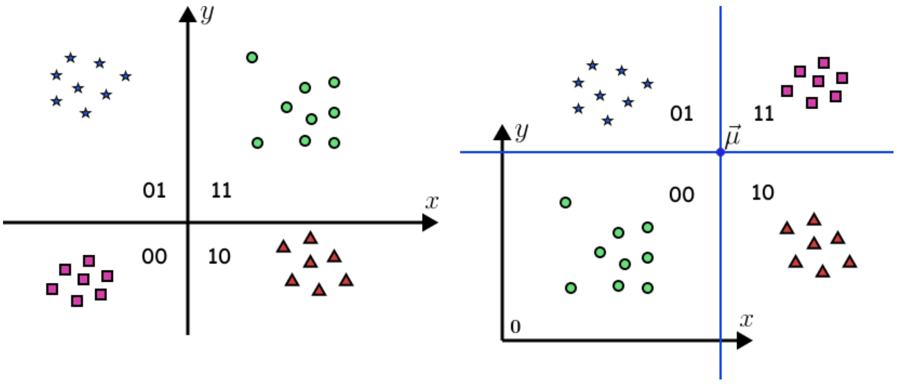

Fig. 3 Applying two indicator functions to two different datasets with 4 people and $N = 2$. **Left:** applying $\mathring{\mathbb{1}}(\mathbf{x} > \mathbf{0}_N)$. **Right:** applying $\mathring{\mathbb{1}}(\mathbf{x} > \boldsymbol{\mu})$.

5 Code based Fuzzy Extractor

In our implementation, we employ a fuzzy extractor based on the code construction presented in [37]. This cryptoprimitive utilizes the principles of the McEliece code-based cryptosystem, as detailed in [43].

We begin by considering a linear block $(n, k, d = 2t+1)$ code defined over the finite field $\mathrm{GF}(q)$. This code possesses a fast (polynomial complexity) decoding algorithm. Let $\mathbf{G}$ denote the generative $k \times n$ matrix associated with this code. In the context of the McEliece cryptosystem, the public key is given by a $k \times n$ matrix, expressed as:

$$\mathbf{G}_X = \mathbf{X} \cdot \mathbf{G} \cdot \mathbf{P} \cdot \mathbf{D} \tag{4}$$

Here, the private keys consist of the following matrices: $\mathbf{X}$, a nonsingular $k \times k$ matrix with elements from $\mathrm{GF}(q)$; $\mathbf{P}$, a permutation $n \times n$ matrix; and $\mathbf{D}$, a diagonal $n \times n$ matrix.

It is worth noting that the McEliece cryptosystem often employs binary Goppa codes [38, 39, 52]. In such cases, the matrices $\mathbf{X}$ and $\mathbf{P}$ contain elements from the field $\mathrm{GF}(2)$, and the matrix $\mathbf{D}$ is omitted.

The ciphertext is represented by a vector of length n, which is computed according to the following rule:

$$\mathbf{c}_X^* = \mathbf{I} \cdot \mathbf{G}_X + \mathbf{e} \tag{5}$$

where $\mathbf{c}_X = \mathbf{I} \cdot \mathbf{G}_X$ - codeword of masked code with a generator matrix $\mathbf{G}_X$, $\mathbf{I}$ - k-bit public text, vector $\mathbf{e}$ - secret error vector with Hamming weight (number of non-zero positions) that equals to $w_H(\mathbf{e}) = t$.

An authorized user, who possesses the secret key, can calculate the vector

$$\overline{\mathbf{c}}^* = \mathbf{c}_X^* \cdot \mathbf{D}^{-1} \cdot \mathbf{P}^{-1} = \mathbf{I}' \cdot \mathbf{G} + \mathbf{e}' \tag{6}$$

and decode it to obtain $\mathbf{I}'$. The public text is then computed using the following relation:

$$\mathbf{I} = \mathbf{I}'\mathbf{X}^{-1} \tag{7}$$

An attacker, who lacks knowledge of the secret key, cannot compute Equation 6 and utilize a fast (polynomial complexity) decoding algorithm. The attacker can only rely on Equation 4, which implies the use of highly complex decoders (exponential complexity). By carefully selecting parameters $(n, k, d = 2t + 1)$, we can achieve an extremely high decoding complexity for potential adversaries. Furthermore, code-based cryptography offers quantum-resistant security, which can be attained by significantly increasing the code parameters $(n, k, d = 2t + 1)$.

In the fuzzy extractor from the paper [37], binary strings extracted from biometric data are interpreted as $\mathbf{c}_X^*$ in Equation 5, and the formed cryptographic password is interpreted as the vector $\mathbf{I}$. The calculation of $\mathbf{I}$, based on the known $\mathbf{c}_X^*$, is associated with decoding an $(n, k, d = 2t+1)$ code with the generative matrix in Equation 4. This implies that the solution to this problem is accessible only to someone who possesses the secret key (matrices $\mathbf{X}$, $\mathbf{P}$, and $\mathbf{D}$).

It is essential to note that decoding a randomly generated sequence may not always be successful. For example, in the binary case, there are 2^n distinct sequences of length n. In this scenario, only $2^k \sum_{i=0}^{t} C_n^i$ sequences will be successfully decoded, resulting in a success probability that decreases proportionally to $t!$:

$$\frac{2^k \sum_{i=0}^{t} C_n^i}{2^n} \approx \frac{1}{t!} \tag{8}$$

This issue can be addressed by generating a reference biometric dataset without decoding, as suggested in [37]. For example, let's assume that we form the reference binary string $\mathbf{c}_X^* = \mathbf{B}$ as a result of multiple biometric scans and averaging the acquired data. We presume that a randomly chosen subset of k positions in this vector remains undistorted, denoted by $\mathbf{B}_k$. We then create the password $\mathbf{I}$ through matrix inversion:

$$\mathbf{I} = \mathbf{B}_k \cdot \mathbf{G}_{X_1}^{-1} \tag{9}$$

In this case, the matrix $\mathbf{G}_{X_1}$ is composed of k columns of the matrix $\mathbf{G}_X$, with column numbers corresponding to the randomly selected k positions of the vector $\mathbf{B}$.

The paper [37] also proposed a method for generating a non-secret helper string, which serves as public information. The use of a helper string significantly reduces the impact of errors in binary biometric strings. The non-secret helper string is formed as the check part of the codeword:

$$\mathbf{P}_{n-k} = \mathbf{I} \cdot \mathbf{G}_{X_2} = \mathbf{B}_k \cdot \mathbf{G}_{X_1}^{-1} \cdot \mathbf{G}_{X_2} \tag{10}$$

where the matrix $\mathbf{G}_{X_2}$ is constructed from the remaining $n - k$ columns of the matrix in Equation 4.

Now, we interpret each newly formed biometric binary vector as a word:

$$\mathbf{B}^* = \mathbf{I} \cdot \mathbf{G}_X + \mathbf{e}^* \tag{11}$$

When using a helper string, we assume that the word in Equation 11 consists of the distorted part $\mathbf{B}_k^*$ affected by the error vector $\mathbf{e}^*$ and the non-secret, undistorted helper string $\mathbf{P}_{n-k}$. In other words, we presume that all non-zero positions of the error vector $\mathbf{e}^*$ are concentrated on the k positions of the vector $\mathbf{B}_k^*$. This approach allows us to effectively address the challenges posed by errors in biometric data while maintaining the security and efficiency of the cryptographic process.

6 Experiments

6.1 Evaluation parameters

To assess accuracy of forming binary strings, we define the *images binary distance* $\delta_{\mathcal{I}}(X, Y)$ between images $X, Y \in \mathcal{I}$ as a ratio of pairwise mismatched spots in images' binary string representations (recall that $\Psi = \psi \circ \phi$):

$$\delta_{\mathcal{I}}(X, Y) \triangleq \frac{1}{N} \cdot \|\Phi(X) - \Phi(Y)\|_1 \tag{12}$$

Similarly, we define *images binary similarity* $\sigma_{\mathcal{I}}(X, Y)$ as simply:

$$\sigma_{\mathcal{I}}(X, Y) \triangleq 1 - \delta_{\mathcal{I}}(X, Y) \tag{13}$$

Let us see which result we get when applying ψ to some images from our dataset. As one might see from the Figure 4, two binary strings of the same person almost coincide. According to our definition, similarity between these two images is approximately 90%, which is a relatively good result. Yet for two different people, for example, as depicted on the Figure 5, binary strings differ significantly and in the given example similarity equals 30%, which is a relatively small value, as expected.

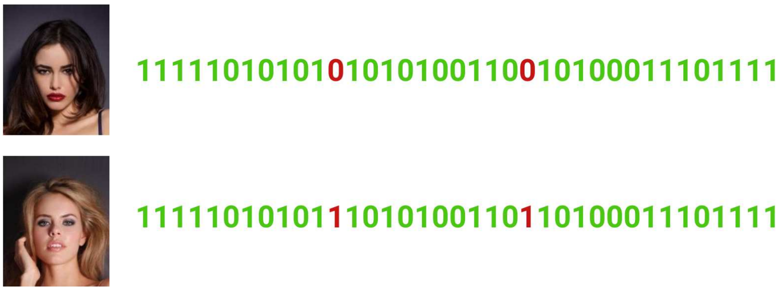

Fig. 4 Binary string for the pair of images of the same person taken from *Celeba* dataset. In green we marked the same characters, whereas in red different ones. For demonstation purposes we included only 37 string characters.

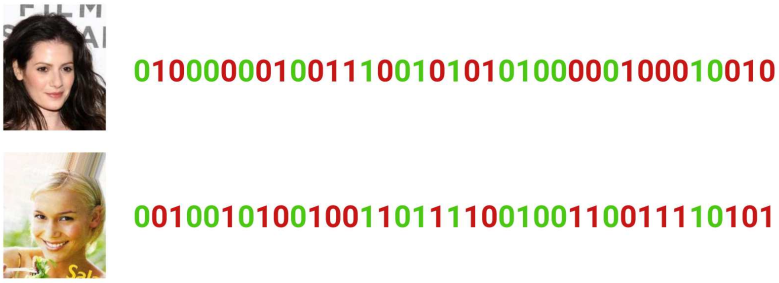

Fig. 5 Binary string for the pair of images of two different people. In green we marked the same characters, whereas in red different ones. For demonstation purposes we included only 37 string characters.

Now let us evaluate the accuracy of such converter on the larger dataset. Firstly, we propose to split set of pairs $\mathcal{P}$ into two other sets: $\mathcal{P}_{\text{same}} = \{(X, Y) \in \mathcal{P} \mid X \equiv Y\}$ – set of image pairs of the same person, and $\mathcal{P}_{\text{diff}} = \{(X, Y) \in \mathcal{P} \mid X \not\equiv Y\}$ – set of pairs of different people.

Then for the accuracy evaluation we will use two values: $\hat{\sigma}_{\text{same}}$ – average similarity between binary strings, formed for the set of pairs of a single person, and $\hat{\sigma}_{\text{diff}}$ for the set of pairs of different people. We define their values as follows:

$$\hat{\sigma}_{\text{diff/same}} = \frac{1}{|\mathcal{P}_{\text{diff/same}}|} \sum_{X,Y \in \mathcal{P}_{\text{diff/same}}} \sigma_{\mathcal{I}}(X, Y) \tag{14}$$

Our goal is to maximize the difference $\hat{\sigma}_{\text{same}} - \hat{\sigma}_{\text{diff}}$ while keeping $\hat{\sigma}_{\text{same}}$ as large as possible.

6.2 Results

As described in section 4, we considered two feature vector converters: without offset $\psi(\boldsymbol{x}) = \mathring{\mathbb{1}}(\boldsymbol{x} > \boldsymbol{0}_N)$ and with offset $\psi(\boldsymbol{x}) = \mathring{\mathbb{1}}(\boldsymbol{x} > \boldsymbol{\mu})$. Now let us evaluate how both work and which results we get.

6.2.1 Vector converter without offset

In Tables 1 and 2 we included cumulative similarities evaluation described earlier for two datasets (*lfw* and *CelebA*) and two models (*Keras Facenet* and *Face Recognition*) using vector converter without offset. For *lfw* we used 1000 images, for *CelebA* approximately 100000 (thus, almost the same number of pairs).

Table 1 Cumulative similarity σ_{same} for pair of images of a same person

Model	Dataset	
	lfw	CelebA
Keras Facenet	0.773	0.737
Face Recognition	0.888	0.885

Table 2 Cumulative similarity σ_{diff} for pair of images of different people

Model	**Dataset**	
	lfw	**CelebA**
Keras Facenet	0.508	0.517
Face Recognition	0.797	0.809

The obtained experiment results show that the binary strings extracted by the *Keras Facenet* and *Face Recognition* models differ in error rates. For example, binary strings for a single person (obtained from images of the same batch) differ by 7-9% when using the *Face Recognition* model and by 22-26% when using the *Keras Facenet* model. In fact, this means that the *Keras Facenet* model is of little use for generating binary strings with such vector converter, because the error rate is at the edge of the corrective power of the correction codes. The *Face Recognition* model, on the contrary, has a high potential for use and a low error rate potentially allows very low FRR values to be achieved.

6.2.2 Vector converter with offset

Since *CelebA* dataset contains much more images than *lfw*, we decided to use it to compare a vector converter with and without an offset.

After applying a new feature vector converter for *Face Recognition* model, we get results depicted in the Table 3.

Table 3 Vector converter comparison based on *Face Recognition* model

Feature converter	**Evaluation parameters**			
	$\hat{\sigma}_{\text{same}}$	**# of same pairs**	$\hat{\sigma}_{\text{diff}}$	**# of diff pairs**
$\mathbb{1}(\mathbf{x} > \mathbf{0}_N)$	0.890	186682	0.802	139481
$\mathbb{1}(\mathbf{x} > \boldsymbol{\mu})$	0.745	186682	0.529	139481

In turn, applying the same method to a *Keras Facenet* model almost did not change similarities values.

Although applying this method to a *Face Recognition* decreased $\hat{\sigma}_{\text{same}}$, difference $\hat{\sigma}_{\text{same}} - \hat{\sigma}_{\text{diff}}$ increased from approximately 0.076 to 0.216 which is a drastic change which makes using *Face Recognition* even a better option.

6.3 Evaluation and comparison of FRR and FAR

For evaluating FRR and FAR, we will consider two cases. Suppose that as a result of scanning and processing of the biometric data, we formed the binary string 11, where the Hamming weight (number of non-zero positions) of an error vector $\mathbf{e}^*$ characterizes possible differences of $\mathbf{B}^*$ with a reference biometric set $\mathbf{B}$. If the number of these differences does not exceed t (corrective ability $(n, k, d = 2t + 1)$ of a code), using manipulations 5 and 6 will allow to correctly generate the same password $\mathbf{I}$.

Number of non-zero position of a vector $\mathbf{e}^*$ is determined by the probability of a non-zero character occurrence in $\mathbf{e}^*$, i.e. probability of distortion of one character of the codeword $\mathbf{c}_X = \mathbf{I} \cdot \mathbf{G}_X$. For an authorized and unauthorized user, these probabilities are different.

Consider the first case.

Suppose that the vector 11 belongs to the authorized user. We denote the probability of one character distortion in $\mathbf{c}_X$ as p_0. Then the value of FRR can be estimated by the formula [37]:

$$\text{Without using helper string: } \text{FRR} = 1 - \sum_{i=0}^{t} C_n^i p_0^i (1-p_0)^{n-i} \tag{15}$$

$$\text{Using helper string: } \text{FRR} = 1 - \sum_{i=0}^{t} C_k^i p_0^i (1-p_0)^{k-i} \tag{16}$$

Consider the second case.

Suppose that the vector 11 belongs to the unauthorized user. We denote the probability of one character distortion as p_1. Then the value FAR can be evaluated according to the formula:

$$\text{Without using helper string: } \text{FAR} = \sum_{i=0}^{t} C_n^i p_1^i (1-p_1)^{n-i} \tag{17}$$

$$\text{Using helper string: } \text{FAR} = \sum_{i=0}^{t} C_k^i p_1^i (1-p_1)^{k-i} \tag{18}$$

In Tables 1 and 3, we present the empirical estimation of $\hat{\sigma}_{\text{same}}$, which represents the average similarity of extracted binary vectors for the same individual using different deep learning models. We utilize these values to calculate $p_0 = 1 - \hat{\sigma}_{\text{same}}$. Our focus is primarily on the results obtained from the *CelebA* model, as it comprises a significantly larger number of images. Consequently, we obtain $p_0 = 0.263$ for the *Keras Facenet* model and $p_0 = 0.255$ for the *Face Recognition* model.

In a similar manner, we assess the value of p_1 based on the experimental results for $\hat{\sigma}_{\text{diff}}$. We derive the following values: $p_1 = 0.483$ for the *Keras Facenet* model and $p_0 = 0.471$ for the *Face Recognition* model. These findings provide crucial insights into the performance of various deep learning models in generating binary vectors for biometric authentication and highlight the potential for further optimization and improvement in this domain.

The extractor task is to minimize FRR and FAR for various lengths of generated passwords and various probabilities p_0 and p_1.

The considered extractor is based on the use of code-based cryptosystems that use a linear block $(n,k,d) = (2^m, 2^m - mt, 2t+1)$ code with a fast (of polynomial complexity) decoding. The safest option is considered to be use the binary Goppa code with parameters

$$(n,k,d) = (2^m, 2^m - mt, 2t+1) \tag{19}$$

for some $m \in \mathbb{Z}^+$.

In our experiments, we formed binary strings of length $n = 128$, i.e. for $m = 7$. In the Tables 4 and 5 we show parameters k, d of Goppa codes for different values of

t. The tables also show the calculated values of FRR and FAR for various cases. The third column contains parameter 2^{-k} - probability of guessing a password of length k bits.

On the Figures 7 and 6 we plotted dependencies of FRR and FAR on different values of t while using helper string.

As evident from the presented data, without the use of a helper string, the fuzzy extractor (for the case of $n = 128$) is unable to achieve low FRR. This observation holds true for both deep learning models, *Keras Facenet* and *Face Recognition*. However, when a helper string is employed, the situation changes dramatically. We observe that for larger values of t, it is possible to select code parameters that result in both FRR and FAR taking on acceptable values.

The importance of these indicators cannot be overstated, as they directly influence the overall performance and security of biometric authentication systems. Low FRR ensures that genuine users are not rejected by the system, thereby providing a smooth and hassle-free authentication experience. On the other hand, low FAR is crucial for preventing unauthorized access by impostors, ensuring the integrity and confidentiality of sensitive data. Striking a balance between FRR and FAR is essential for the development and implementation of reliable and robust biometric authentication systems. The use of a helper string, as demonstrated in our study, has proven to be a promising approach to achieve this balance and enhance the effectiveness of fuzzy extractors in the context of biometric authentication.

Table 4 FRR and FAR estimates for various Goppa codes of length 128 (without using helper string)

				Keras Facenet		*Face Recognition*	
t	k	2^{-k}	d	FRR	FAR	FRR	FAR
1	121	3.76×10^{-37}	3	1.0000	0.0000	1.0000	0.0000
2	114	4.81×10^{-35}	5	1.0000	0.0000	1.0000	0.0000
3	107	6.16×10^{-33}	7	1.0000	0.0000	1.0000	0.0000
4	100	7.89×10^{-31}	9	1.0000	0.0000	1.0000	0.0000
5	93	1.01×10^{-28}	11	1.0000	0.0000	1.0000	0.0000
6	86	1.29×10^{-26}	13	1.0000	0.0000	1.0000	0.0000
7	79	1.65×10^{-24}	15	1.0000	0.0000	1.0000	0.0000
8	72	2.12×10^{-22}	17	1.0000	0.0000	1.0000	0.0000
9	65	2.71×10^{-20}	19	1.0000	0.0000	1.0000	0.0000
10	58	3.47×10^{-18}	21	1.0000	0.0000	1.0000	0.0000
11	51	4.44×10^{-16}	23	1.0000	0.0000	1.0000	0.0000
12	44	5.68×10^{-14}	25	1.0000	0.0000	1.0000	0.0000
13	37	7.28×10^{-12}	27	1.0000	0.0000	1.0000	0.0000
14	30	9.31×10^{-10}	29	1.0000	0.0002	1.0000	0.0002
15	23	1.19×10^{-7}	31	1.0000	0.0012	0.9999	0.0011
16	16	1.53×10^{-5}	33	0.9999	0.0078	0.9998	0.0069
17	9	1.95×10^{-3}	35	0.9997	0.0489	0.9994	0.0411
18	2	0.25	37	0.9994	0.2854	0.9988	0.2288

To date, acceptable indicators of biometric authentication based on the image of a person's face are: FRR ≈ 25%, FAR ≈ 10% [53–55]. At the same time, significant

Table 5 FRR and FAR estimates for various Goppa codes of length 128 (using helper string)

				Keras Facenet		*Face Recognition*	
t	k	2^{-k}	d	FRR	FAR	FRR	FAR
1	121	3.76×10^{-37}	3	1.0000	0.0000	1.0000	0.0000
2	114	4.81×10^{-35}	5	1.0000	0.0000	1.0000	0.0000
3	107	6.16×10^{-33}	7	1.0000	0.0000	1.0000	0.0000
4	100	7.89×10^{-31}	9	1.0000	0.0000	1.0000	0.0000
5	93	1.01×10^{-28}	11	1.0000	0.0000	1.0000	0.0000
6	86	1.29×10^{-26}	13	1.0000	0.0000	1.0000	0.0000
7	79	1.65×10^{-24}	15	0.9999	0.0000	0.9998	0.0000
8	72	2.12×10^{-22}	17	0.9987	0.0000	0.9979	0.0000
9	65	2.71×10^{-20}	19	0.9877	0.0000	0.9825	0.0000
10	58	3.47×10^{-18}	21	0.9262	0.0000	0.9054	0.0000
11	51	4.44×10^{-16}	23	0.7229	0.0000	0.6781	0.0001
12	44	5.68×10^{-14}	25	0.3663	0.0036	0.3210	0.0058
13	37	7.28×10^{-12}	27	**0.0830**	**0.0744**	**0.0666**	**0.0972**
14	30	9.31×10^{-10}	29	0.0047	0.5023	0.0034	0.5551
15	23	1.19×10^{-7}	31	0.0000	0.9673	0.0000	0.9749

progress has been made in facial recognition technologies in recent years. For example, according to studies of NIST [56] the best face recognition algorithm has an error rate of only about 0.08% (under ideal conditions). At the same time, for systems for generating biometric passwords, the FAR values should be as small as possible, preferably comparable to the probability of guessing a password. In the Table 4 we highlighted the case with FRR $\approx$ FAR $<$ 10%. On Figure 8 we considered dependencies of FRR and FAR in these cases for different values of p_0 and p_1.

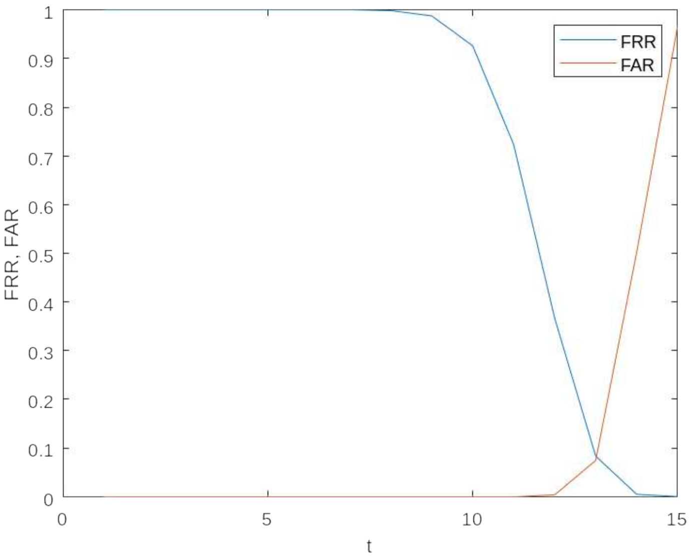

Fig. 6 Dependencies FRR(t) and FAR(t) using helper string, *Keras Facenet* deep learning model.

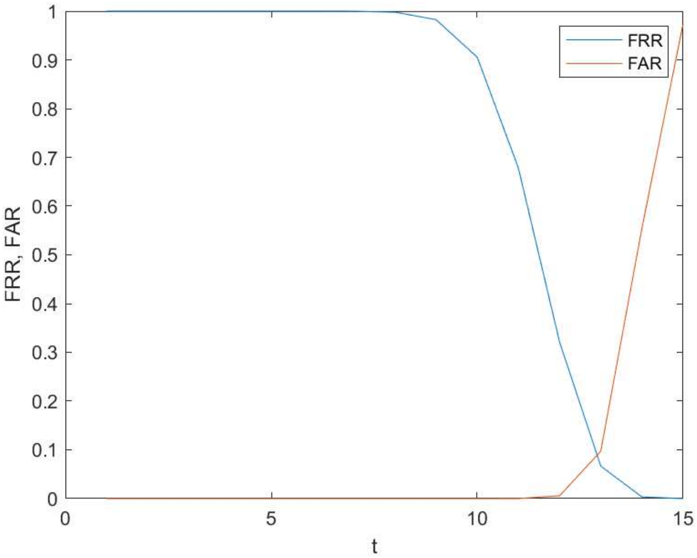

Fig. 7 Dependencies FRR(t) and FAR(t) using helper string, *Face Recognition* deep learning model

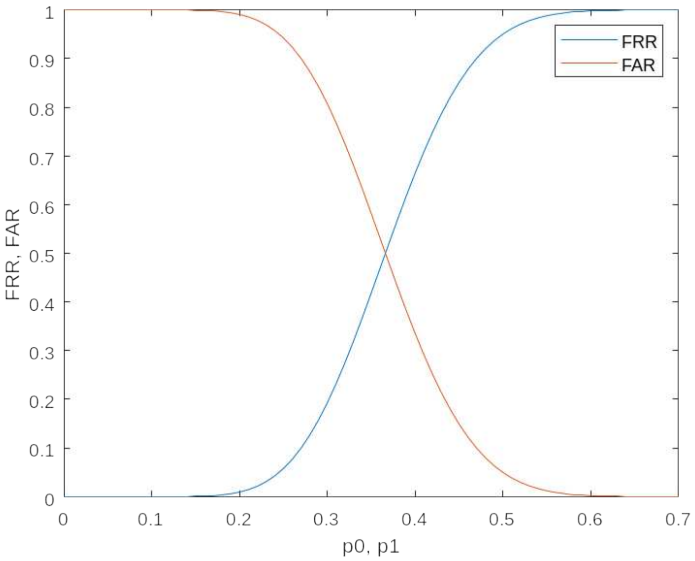

Fig. 8 Dependencies FRR(p_0) and FAR(p_1) without using helper string, Goppa code with parameters $(n, k, d) = (128, 37, 27)$

7 Comparison results

In this study, we have investigated methods for generating cryptographic keys from biometric facial images. This is an important research area that has been evolving in many related works. A major gap in the field of research has been the assessment of error rates in the extracted binary strings, which are vital for the subsequent evaluation of FRR and FAR.

In our research, we employed deep learning models, *Keras Facenet* and *Face Recognition*, to assess error rates for input images from different clusters:

- Images of faces of the same individual;
- Images of faces of different individuals.

Binary strings corresponding to the same person should have minimal differences to ensure a low FRR. Conversely, for binary strings extracted from images of different individuals, the error rates should be as high as possible to achieve low FAR values.

The final comparison of FRR and FAR is presented in Tables 4 and 5. The results indicate that both investigated models are roughly equivalent in performance. It is evident that these findings warrant further refinement and validation through independent testing. Nonetheless, our study suggests that the deep learning models utilized in this research hold significant potential for generating cryptographic keys using a code-based fuzzy extractor.

The development of efficient and secure methods for generating cryptographic keys from biometric data is essential for the broader implementation of robust biometric authentication systems. As the field continues to advance, the importance of validating and optimizing the performance of deep learning models in key generation will remain a critical aspect of research. By harnessing the potential of deep learning models and code-based fuzzy extractors, we can further strengthen the security and reliability of biometric authentication systems, ultimately enhancing data protection and user privacy in various applications.

8 Conclusion

Biometrics has become a popular choice for authentication purposes due to its inherent advantages, such as the uniqueness and non-transferability of an individual's physical or behavioral traits. However, traditional biometric authentication methods necessitate the storage of reference biometric images, which poses several challenges. Storing and processing biometric data require expensive infrastructure and elaborate methods to protect personal information. These complexities hinder the widespread adoption and development of biometric authentication systems.

An alternative approach to address these issues involves the generation of biometric keys that can be created ”on-the-fly” without the need for storing reference biometric images. These keys should satisfy all the requirements for cryptographic strength while eliminating the need for storage or distribution. Biometric data is always readily available, enabling the generation of cryptographic keys at any given moment. This innovative method offers the potential for more secure, convenient, and efficient authentication processes, paving the way for a new era in biometric-based security systems.

In this study, we have presented a novel approach to generate secure cryptographic keys from biometric data, leveraging deep learning models and code-based cryptography for enhanced security. Our method combines the advantages of biometrics, such as the inherent uniqueness and non-transferability of biometric features, with the robustness and post-quantum security provided by code-based cryptographic extractors.

We have demonstrated the effectiveness of our approach by experimenting with various deep learning models, such as convolutional neural networks, to extract biometric features from facial images. Our optimized algorithm parameters achieve an FRR and FAR below 10%, ensuring the generated keys are suitable for biometric authentication. Moreover, the use of code-based cryptographic extractors offers a post-quantum level of security, making our method resistant to potential future quantum computing attacks.

Our study also addresses the challenge of errors in biometric data by introducing a non-secret helper string, which significantly reduces the impact of errors in binary biometric strings. This allows for a more reliable and accurate authentication process while maintaining the security of the generated keys.

In conclusion, our proposed method offers a promising and secure alternative to traditional cryptographic key generation and storage, simplifying the process and enhancing security by utilizing biometric data and post-quantum cryptography. Future research could explore the applicability of our method to other biometric modalities, such as fingerprint or iris recognition, and investigate the potential for further optimization of deep learning models and cryptographic extractors to improve the overall performance and security of the system.

Declarations

- Funding
  1. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101007820.
  2. This publication reflects only the author's view and the REA is not responsible for any use that may be made of the information it contains.
- Conflict of interest/Competing interests
  The authors have no conflicts of interest to declare that are relevant to the content of this article.
- Ethics approval
  Not applicable.
- Consent to participate
  Not applicable.
- Consent for publication
  Not applicable.
- Availability of data and materials/Data Accessibility Statement/Code availability
  Our manuscript has no associated data or code.
- Authors' contributions
  These authors contributed equally to this work.

All authors have read and agreed to the published version of the manuscript.

References

[1] Chakraborty, S., Das, D.: An Overview of Face Liveness Detection. arXiv:1405.2227 [cs] (2014). arXiv: 1405.2227. Accessed 2021-02-12

[2] Rathgeb, C., Uhl, A.: A survey on biometric cryptosystems and cancelable biometrics. EURASIP Journal on Information Security **2011**(1), 3 (2011) https://doi.org/10.1186/1687-417X-2011-3

[3] Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric cryptosystems: issues and challenges. Proceedings of the IEEE **92**(6), 948–960 (2004) https://doi.org/10.1109/JPROC.2004.827372

[4] Lutsenko, M., Kuznetsov, A., Kiian, A., Smirnov, O., Kuznetsova, T.: Biometric Cryptosystems: Overview, State-of-the-art and Perspective Directions. Lecture Notes in Networks and Systems, vol. 152, p. 84. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-58359-0_5

[5] Jin, Z., Teoh, A.B.J., Goi, B.-M., Tay, Y.-H.: Biometric cryptosystems: A new biometric key binding and its implementation for fingerprint minutiae-based representation. Pattern Recognition **56**, 50–62 (2016) https://doi.org/10.1016/j.patcog.2016.02.024

[6] Pane, A., Chen, T.M., Nepomuceno, E.: In: Daimi, K., Francia III, G., Encinas, L.H. (eds.) Biometric Cryptography, pp. 3–28. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10706-1_1 . https://doi.org/10.1007/978-3-031-10706-1_1

[7] www.html-factory.cz, Jiřík, P.: The Future of Multi-Factor Biometric Authentication (2021). https://www.phonexia.com/blog/the-future-of-multi-factor-biometric-authentication/

[8] Rui, Z., Yan, Z.: A survey on biometric authentication: Toward secure and privacy-preserving identification. IEEE Access **7**, 5994–6009 (2019) https://doi.org/10.1109/ACCESS.2018.2889996

[9] hamme, T.V., Garofalo, G., Joos, S., Preuveneers, D., Joosen, W.: In: Batina, L., Bäck, T., Buhan, I., Picek, S. (eds.) AI for Biometric Authentication Systems. Lecture Notes in Computer Science, pp. 156–180. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98795-4_8 . https://doi.org/10.1007/978-3-030-98795-4_8

[10] Valderrama, W., Magadán, A., Vergara, O.O., Ruiz, J., Pinto, R., Reyes, G.: Detection of facial spoofing attacks in uncontrolled environments using elbp and color models. IEEE Latin America Transactions **20**(66), 875–883 (2022)

[11] Wang, G., Wang, Z., Jiang, K., Huang, B., He, Z., Hu, R.: Silicone mask face anti-spoofing detection based on visual saliency and facial motion. Neurocomputing **458**, 416–427 (2021) https://doi.org/10.1016/j.neucom.2021.06.033

[12] Menezes, A.J., Oorschot, P.C.v., Vanstone, S.A., Oorschot, P.C.v., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, ??? (2018). https://doi.org/10.1201/9780429466335 . https://www.taylorfrancis.com/books/9780429466335

[13] Klima, R.E., Klima, R., Sigmon, N.P., Sigmon, N., Klima, R., Sigmon, N.P., Sigmon, N.: Cryptology: Classical and Modern. Chapman and Hall/CRC, ??? (2018). https://doi.org/10.1201/9781315170664 . https://www.taylorfrancis.com/books/9781315170664

[14] Rubinstein-Salzedo, S.: Cryptography. Springer Undergraduate Mathematics Series. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94818-8 . http://link.springer.com/10.1007/978-3-319-94818-8

[15] Kuznetsov, A.A., Gorbenko, Y., Kiian, A.K.A., Ulianovska, Y.V., Kuznetsova,

T.: Elliptic curve pseudorandom bit generator with maximum period sequences. International Journal of Computing, 494–505 (2021) https://doi.org/10.47839/ijc.20.4.2436

[16] Kuznetsov, A., Kiian, A., Smirnov, O., Cherep, A., Kanabekova, M., Chepurko, I.: Testing of code-based pseudorandom number generators for post-quantum application. In: 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), pp. 172–177 (2020). https://doi.org/10.1109/DESSERT50317.2020.9125045

[17] Delfs, H., Knebl, H.: Introduction to Cryptography. Information Security and Cryptography. Springer, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47974-2 . http://link.springer.com/10.1007/978-3-662-47974-2

[18] Post-Quantum Cryptography. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7 . http://link.springer.com/10.1007/978-3-540-88702-7

[19] Post-Quantum Cryptography. Lecture Notes in Computer Science, vol. 9606. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8 . http://link.springer.com/10.1007/978-3-319-29360-8

[20] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014). https://doi.org/10.1109/CVPR.2014.220

[21] Kinkiri, S., Keates, S.: Speaker identification: Variations of a human voice. In: 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE), pp. 1–4 (2020). https://doi.org/10.1109/ICACCE49060.2020.9154998

[22] Hsiao, C.-S., Fan, C.-P., Hwang, Y.-T.: Iris location and recognition by deep-learning networks based design for biometric authorization. In: 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech), pp. 144–145 (2021). https://doi.org/10.1109/LifeTech52111.2021.9391787

[23] Chuang, C.-W., Fan, C.-P.: Biometric authentication with combined iris and sclera information by yolo-based deep-learning network. In: 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), pp. 1–2 (2020). https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258253

[24] Qin, Z., Huang, G., Xiong, H., Qin, Z., Choo, K.-K.R.: A fuzzy authentication system based on neural network learning and extreme value statistics. IEEE Transactions on Fuzzy Systems **29**(3), 549–559 (2021) https://doi.org/10.1109/TFUZZ.2019.2956896

[25] Jana, A., Sarker, M.K., Ebrahimi, M., Hitzler, P., Amariucai, G.T.: Neural fuzzy extractors: A secure way to use artificial neural networks for biometric user authentication. arXiv:2003.08433 [cs] (2020). arXiv: 2003.08433

[26] Fuller, B., Reyzin, L., Smith, A.: When Are Fuzzy Extractors Possible? vol. 961, (2014). http://eprint.iacr.org/2014/961

[27] Kuznetsov, A., Zakharov, D., Frontoni, E., Romeo, L., Rosati, R.: Deep learning based fuzzy extractor for generating strong keys from biometric face images. In: 2022 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S T) (2022)

[28] Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceedings of the 6th ACM Conference on Computer and Communications Security. CCS '99, pp. 28–36. Association for Computing Machinery, New York, NY, USA (1999). https://doi.org/10.1145/319709.319714 . https://doi.org/10.1145/319709.319714

[29] Juels, A.: In: Tuyls, P., Skoric, B., Kevenaar, T. (eds.) Fuzzy Commitment, pp. 45–56. Springer, London (2007). https://doi.org/10.1007/978-1-84628-984-2_3 . https://doi.org/10.1007/978-1-84628-984-2_3

[30] Chauhan, S., Sharma, A.: A generalized approach for the fuzzy commitment scheme. Journal of Cyber Security Technology **3**(4), 189–204 (2019) https://doi.org/10.1080/23742917.2019.1631429

[31] Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) Audio- and Video-Based Biometric Person Authentication. Lecture Notes in Computer Science, pp. 310–319. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/11527923_32

[32] Juels, A., Sudan, M.: A fuzzy vault scheme. Designs, Codes and Cryptography **38**(2), 237–257 (2006) https://doi.org/10.1007/s10623-005-6343-z

[33] Frassen, T., Zhou, X., Busch, C.: Fuzzy vault for 3d face recognition systems. In: 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1069–1074 (2008). https://doi.org/10.1109/IIH-MSP.2008.315

[34] Banerjee, S., Odelu, V., Das, A.K., Srinivas, J., Kumar, N., Chattopadhyay, S., Choo, K.-K.R.: A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment. IEEE Internet of Things Journal **6**(5), 8739–8752 (2019) https://doi.org/10.1109/JIOT.2019.2923373

[35] Jakobsson, M., Liu, D.: In: Jakobsson, M. (ed.) Your Password is Your New PIN. SpringerBriefs in Computer Science, pp. 25–36. Springer, New York, NY (2013). https://doi.org/10.1007/978-1-4614-4878-5_3 . https://doi.org/10.1007/

978-1-4614-4878-5_3

[36] Álvarez, F.H., Encinas, L.H.: Security efficiency analysis of a biometric fuzzy extractor for iris templates. In: Herrero, , Gastaldo, P., Zunino, R., Corchado, E. (eds.) Computational Intelligence in Security for Information Systems. Advances in Intelligent and Soft Computing, pp. 163–170. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04091-7_20

[37] Kuznetsov, A., Kiyan, A., Uvarova, A., Serhiienko, R., Smirnov, V.: New code based fuzzy extractor for biometric cryptography. In: Int. Sci.-Pract. Conf. Probl. Infocommunications Sci. Technol., PIC S T - Proc., pp. 119–124. Institute of Electrical and Electronics Engineers Inc., ??? (2019). https://doi.org/10.1109/INFOCOMMST.2018.8632040 . journalAbbreviation: Int. Sci.-Pract. Conf. Probl. Infocommunications Sci. Technol., PIC S T - Proc.

[38] Overbeck, R., Sendrier, N.: In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Code-based cryptography, pp. 95–145. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_4 . https://doi.org/10.1007/978-3-540-88702-7_4

[39] Wang, W., Szefer, J., Niederhagen, R.: Fpga-based niederreiter cryptosystem using binary goppa codes. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. Lecture Notes in Computer Science, pp. 77–98. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_4

[40] Bardet, M., Chaulet, J., Dragoi, V., Otmani, A., Tillich, J.-P.: Cryptanalysis of the mceliece public key cryptosystem based on polar codes. In: Takagi, T. (ed.) Post-Quantum Cryptography. Lecture Notes in Computer Science, pp. 118–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_9

[41] Maurich, I., Heberle, L., Güneysu, T.: Ind-cca secure hybrid encryption from qc-mdpc niederreiter. In: Takagi, T. (ed.) Post-Quantum Cryptography. Lecture Notes in Computer Science, pp. 1–17. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_1

[42] Moody, D., Perlner, R.: Vulnerabilities of “mceliece in the world of escher”. In: Takagi, T. (ed.) Post-Quantum Cryptography. Lecture Notes in Computer Science, pp. 104–117. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_8

[43] McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report **44**, 114–116 (1978)

[44] Sendrier, N.: In: Tilborg, H.C.A., Jajodia, S. (eds.) Niederreiter Encryption Scheme, pp. 842–843. Springer, Boston, MA (2011). https://doi.org/10.1007/978-1-4419-5906-5_385 . https://doi.org/10.1007/978-1-4419-5906-5_385

[45] Classic McEliece: NIST submission. https://classic.mceliece.org/nist.html

[46] Classic McEliece: Intro. https://classic.mceliece.org/index.html

[47] Lutsenko, M., Kuznetsov, A., Gorbenko, Y., Oleshko, I., Pronchakov, Y., Kotukh, Y.: Key generation from biometric data of iris. In: 2019 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), pp. 1–6 (2019). https://doi.org/10.1109/UkrMiCo47782.2019.9165457

[48] Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015). https://doi.org/10.1109/CVPR.2015.7298682

[49] Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning — by Adam Geitgey — Medium. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[50] Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. (2008). https://hal.inria.fr/inria-00321923

[51] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild (arXiv:1411.7766) (2015) https://doi.org/10.48550/arXiv.1411.7766 . arXiv:1411.7766 [cs]

[52] Lint, J.H., Geer, G.: In: Lint, J.H., Geer, G. (eds.) Classical Goppa codes. DMV, pp. 22–24. Birkhäuser, Basel (1988). https://doi.org/10.1007/978-3-0348-9286-5_5 . https://doi.org/10.1007/978-3-0348-9286-5_5

[53] Hua, G.: In: Schintler, L.A., McNeely, C.L. (eds.) Facial Recognition Technologies, pp. 475–479. Springer, Cham (2022). https://doi.org/10.1007/978-3-319-32010-6_93 . https://doi.org/10.1007/978-3-319-32010-6_93

[54] Libby, C., Ehrenfeld, J.: Facial recognition technology in 2021: Masks, bias, and the future of healthcare. Journal of Medical Systems **45**(4), 39 (2021) https://doi.org/10.1007/s10916-021-01723-w

[55] Gates, K.A.: Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance. NYU Press, ??? (2011)

[56] Grother, P.J., Ngan, M.L., Hanaoka, K.K.: Ongoing face recognition vendor test (frvt) part 2: Identification. NIST (2018). Last Modified: 2018-11-27T15:11-05:00