applied sciences

MDPI

Article

A Secure Biometric Key Generation Mechanism via Deep Learning and Its Application

Yazhou Wang [1], Bing Li [1,2,3,*], Yan Zhang [3], Jiaxin Wu [1] and Qianya Ma [1]

[1] School of Microelectronics, Southeast University, Nanjing 210096, China; 230179339@seu.edu.cn (Y.W.); jx_wu@seu.edu.cn (J.W.); 220206031@seu.edu.cn (Q.M.)
[2] Shenzhen Research Institute, Southeast University, Shenzhen 518000, China
[3] School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China; yanzhang930807@seu.edu.cn
* Correspondence: bernie_seu@seu.edu.cn; Tel.: +86-153-6504-5432

Abstract: Biometric keys are widely used in the digital identity system due to the inherent uniqueness of biometrics. However, existing biometric key generation methods may expose biometric data, which will cause users' biometric traits to be permanently unavailable in the secure authentication system. To enhance its security and privacy, we propose a secure biometric key generation method based on deep learning in this paper. Firstly, to prevent the information leakage of biometric data, we utilize random binary codes to represent biometric data and adopt a deep learning model to establish the relationship between biometric data and random binary code for each user. Secondly, to protect the privacy and guarantee the revocability of the biometric key, we add a random permutation operation to shuffle the elements of binary code and update a new biometric key. Thirdly, to further enhance the reliability and security of the biometric key, we construct a fuzzy commitment module to generate the helper data without revealing any biometric information during enrollment. Three benchmark datasets including ORL, Extended YaleB, and CMU-PIE are used for evaluation. The experiment results show our scheme achieves a genuine accept rate (GAR) higher than the state-of-the-art methods at a 1% false accept rate (FAR), and meanwhile satisfies the properties of revocability and randomness of biometric keys. The security analyses show that our model can effectively resist information leakage, cross-matching, and other attacks. Moreover, the proposed model is applied to a data encryption scenario in our local computer, which takes less than 0.5 s to complete the whole encryption and decryption at different key lengths.

Keywords: biometrics; security; privacy; deep learning

Citation: Wang, Y.; Li, B.; Zhang, Y.; Wu, J.; Ma, Q. A Secure Biometric Key Generation Mechanism via Deep Learning and Its Application. *Appl. Sci.* **2021**, *11*, 8497. https://doi.org/10.3390/app11188497

Academic Editors: Larbi Boubchir, Elhadj Benkhelifa and Boubaker Daachi

Received: 15 August 2021
Accepted: 10 September 2021
Published: 13 September 2021

1. Introduction

With the rapid development of biometrics-based recognition technology, biometric images (e.g., face, iris, fingerprint, iris, retina) can be adopted to generate a biometric key (bio-key), which is used as a user's physical identity in the fields of IoT, blockchain, and cloud computing [1–3]. In recent years, people have paid more attention to the privacy and security of biometric data. Once the bio-key generation system exposes biometric data, attackers can utilize this data to access the server for stealing the user's private information, which leads to sensitive information leakage and financial loss. In addition, biometric data is permanently associated with the user's natural identity, thus revocation of the user's biometric trait is impossible [4]. Therefore, for a secure and reliable bio-key generation approach, there are three main issues to be solved. As shown in Figure 1, these issues are as follows:

1. Accuracy issue. Generated bio-key is affected by some variations of the biometric image such as illumination, blur, and pose.

2. Security issue. Since the stored helper data or auxiliary information has the risk of information leakage, an attacker can reconstruct biometric data from the helper data in a database.
3. Privacy issue. Once the bio-key is leaked, an attacker can use the leaked key to achieve authentication in other applications. Moreover, a new bio-key cannot be regenerated to deploy the application system.

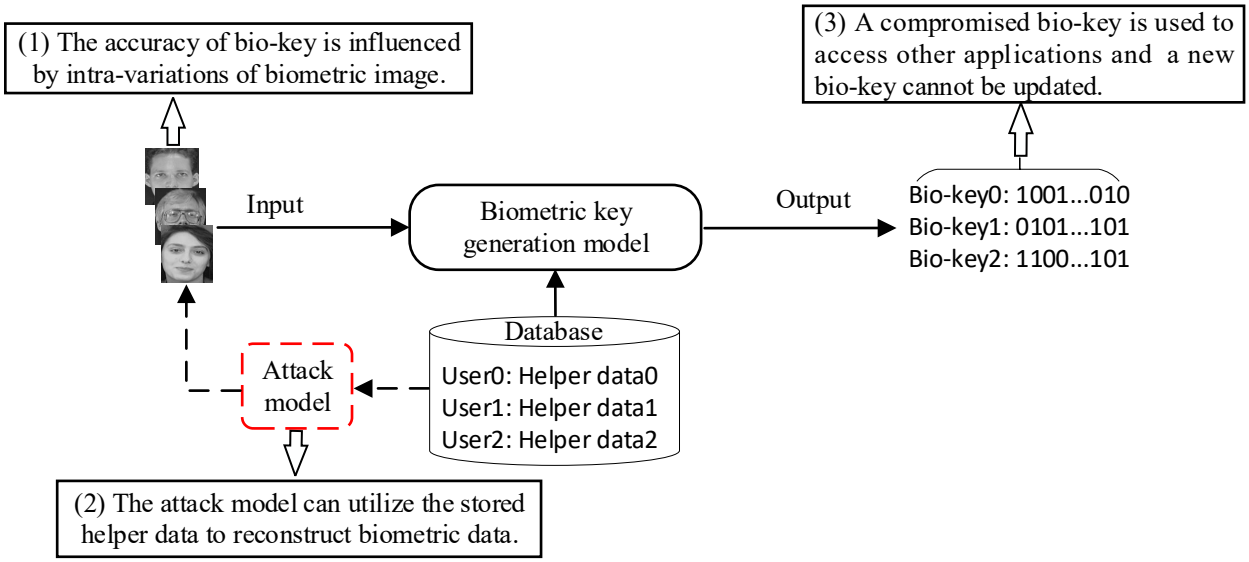

Figure 1. Illustration of security, privacy, and accuracy issues in the biometric key generation methods.

Many researchers have explored different approaches to solve these issues. The traditional bio-key generation scheme is divided into three categories [5,6]: key binding, key generation, and secure sketch and fuzzy extractor, which face the following challenges:

1. For the key binding scheme, biometric data and cryptographic key are bound to generate the helper data for hiding the biometric information. There are two typical instances of this scheme: fuzzy commitment and fuzzy vault. On the one hand, Ignatenko et al. [7] demonstrate the fuzzy commitment approach leaks the biometric information. On the other hand, Kholmatov et al. [8] show that multiple helper data of the fuzzy vault can be filtered chaff points to retrieve bio-key via the correlation attack. Thus, they both face the information leakage challenge.
2. For the key generation scheme, biometric data is used to directly generate bio-keys without the external auxiliary information. However, the accuracy of the generated bio-key is sensitive to intra-user variations. In addition, since the input biometric data is continuous, generating a high-entropy bio-key is difficult [9]. Therefore, there is still room for improvement in accuracy and security.
3. For the secure sketch and fuzzy extractor schemes, they both use auxiliary information to restore the bio-key. Nevertheless, Smith et al. [10] and Dodis et al. [11] demonstrate that these two schemes have information leakage risk. Furthermore, multiple uses of helper data cause privacy risk [12].

With the rapid development of deep learning in the field of biometric recognition [13,14], Pandey et al. [15] use a deep neural network (DNN) to learn maximum entropy binary (MEB) codes from biometric images. Roh et al. [16] design a bio-key generation method based on a convolutional neural network (CNN) and a recurrent neural network (RNN). Roy et al. [17] propose a DNN framework to learn robust biometric features for improving authentication accuracy. However, these methods based on the DNN or CNN scheme did not consider the mentioned challenges of security and privacy.

To overcome the above challenges, we propose a secure bio-key generation method based on deep learning. The proposed approach is used to improve security and privacy

while maintaining accuracy in the biometric authentication system. Specifically, it consists of three parts: (1) a biometrics mapping network; (2) a random permutation module; and (3) a fuzzy commitment module. Firstly, the generated binary code by the random number generator (*RNG*) can represent the biometric data for each user. Subsequently, we adopt the biometrics mapping network to learn the mapping relationship between the biometric data and the binary code during enrollment, which can preserve the recognition accuracy and prevent the information leakage of biometric data. Then, a random permutation module is designed to shuffle the elements of the binary code for generating the distinctive bio-keys without retraining the biometrics mapping network, which keeps the generated bio-key revocable. Next, we construct the fuzzy commitment module to encode the random binary code for generating the auxiliary data without revealing any biometric information. The bio-key is decoded from query biometric data with the help of the auxiliary data, which enhances its stability and security. Finally, the proposed scheme is applied to the AES encryption scenario for verifying its availability and practicality on our local computer. In this work, we use face image as the biometric trait to demonstrate our proposed approach. In summary, the contributions of our paper are summarized as follows:

1. We design a biometrics mapping network based on the DNN framework to obtain the random binary code from biometric data, which prevents information leakage and maintains the accuracy performance under intra-user variations.
2. We propose a revocable bio-key protection approach by utilizing a random permutation module, which can powerfully guarantee the revocability and protect the privacy of bio-key.
3. We construct a fuzzy commitment architecture through an error-correcting technique, which can generate stable bio-keys with the help of auxiliary data, and avoid the exposure of bio-key and biometric data during enrollment.
4. We conduct extended experiments on three benchmark datasets, and the results show that our model not only effectively improves the accuracy performance but also enhances the security and privacy of the biometric authentication system.
5. Furthermore, we validate our bio-key generation model in the AES encryption application, which can reliably generate the bio-keys with different lengths to meet practical encryption requirements on our local computer.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed approach of bio-key generation in detail. Section 4 discusses our experimental results. Finally, we conclude in Section 5.

2. Related Work

Bio-key generation schemes can be classified into key binding, key generation, secure sketch and fuzzy extractor, and machine learning. Therefore, we briefly review these schemes in this section.

2.1. Key Binding Scheme Based on Biometrics

This scheme is used to generate a bio-key by binding biometric data with the secret key. Specifically, the biometric data and the key are bound to generate helper data during the enrollment stage. If the query biometric data is different from the registered biometrics with a limited error, the bio-key can be retrieved by the helper data. This scheme has two typic instances: fuzzy commitment [18] and fuzzy vault [19]. Hao et al. [20] proposed a fuzzy commitment approach based on a coding scheme that used Hadamard code and Reed-Solomon codes. Veen et al. [21] presented a renewable fuzzy commitment method that integrated helper data in a biometric recognition system. Chauhan S et al. [22] proposed a fuzzy commitment approach based on the Reed-Solomon code that removed the error of the biometric template. However, the above methods based on fuzzy commitment do not guarantee that input biometric data is high entropy. Ignatenko et al. [7] and Zhou et al. [23] demonstrated the fuzzy commitment scheme existed information leakage when input biometric data is low entropy. Moreover, Rathgeb et al. [24,25] proposed a statistical attack

that could attack different fuzzy commitment schemes. Clancy et al. [26] improved the fuzzy vault scheme that provided an optimized algorithm by exploiting the best vault parameters. Uludag et al. [27] combined the fuzzy vault with helper data to protect biometric data. Nandakumar et al. [28] utilized the helper data to align the biometrics and query biometrics for improving the authentication accuracy. Li C et al. [29] designed a fuzzy vault scheme by utilizing a pair-polar structure to increase the reliability of the cryptosystem. Nevertheless, the attacker can compare multiple vaults to obtain a candidate set of real points mixed by using attack via record multiplicity (ARM) in the fuzzy vault scheme [30–32]. Therefore, the above methods cannot ensure security and privacy in the key binding scheme. In this paper, we propose a deep learning framework to generate random binary code, and utilize random binary codes to represent biometric data, which can effectively prevent information leakage.

2.2. Key Generation Scheme Based on Biometrics

The task of the key generation scheme is to directly generate a bio-key from biometric traits. Zhang et al. [33] proposed a generalized thresholding approach for improving the authentication accuracy and the security of the bio-key. Hoque et al. [34] presented a key generation method based on several feature partitioning schemes. Rathgeb et al. [35] designed an interval-mapping approach that mapped the features into intervals for generating the bio-key. Lalithamani et al. [36] described a non-invertible bio-key generation approach from biometric templates. The main idea of this approach is to divide the templates into two vectors, and then shuffle the divided vectors and convert them into a matrix to ensure irreversibility. Wu et al. [37] proposed a key generation approach based on face images that combined binary quantization and Reed-Solomon techniques. Ranjan et al. [38] introduced a key generation approach based on the distance to reduce some complex operations for generating the bio-key. Sarkar et al. [39] gave a cancelable key generation approach for asymmetric cryptography. Specifically, they adopted a transformation method based on shuffling to generate the revocable bio-key. Anees et al. [40] presented a bio-key generation method based on binary feature extraction and quantization. However, these methods do not consider the intra-user variations, which makes it difficult to generate stable bio-keys. Moreover, maintaining a high entropy of the key is the main challenge when the bio-key is derived directly from the biometric data.

2.3. Secure Sketch and Fuzzy Extractor Scheme Based on Biometrics

Dodis et al. [41] first proposed secure sketch and fuzzy extractor notions. On the one hand, the secure sketch could generate helper data that did not reveal biometric data and yet recovered the bio-key when query data was close to biometric data. Therefore, this scheme has error correction capability and can correct error-prone biometric data. On the other hand, the fuzzy extractor could obtain biometrics to produce a uniform bio-key for applying various cryptographic applications. Chang et al. [42] designed a hiding secret points approach based on the secure sketch scheme. Sutcu et al. [43] presented a secure sketch by fusing face and fingerprint features for enhancing security. Li et al. [44] proposed two levels of quantization approach for constructing a robust and effective secure sketch. Specifically, they used the first quantizer to calculate the difference between the codeword and noise data, and further utilized the second quantizer to quantize the difference for correcting the noise. Lee et al. [45] added some random noise into the minutiae measurements to construct a fuzzy extractor. Yang et al. [46] improved the fuzzy extractor scheme through registration-free and Delaunay triangulation for improving authentication performance. Chi et al. [47] proposed a multi-biometric cryptosystem that combined secret share and fuzzy extractor approaches. Alexandr et al. [48] designed a new fuzzy extractor without the non-secret helper data for improving its security. Nevertheless, these methods did not take information leakage into consideration. Smith et al. [10] and Dodis et al. [11] demonstrated that the secure sketch and fuzzy extractor schemes would leak information about input biometric data. Morever, Linnartz et al. [12] showed they

suffered from privacy risks in the case of multiple uses. Hence, the above methods still have weaknesses in security and privacy.

2.4. Machine Learning Scheme

With the rapid development of machine learning and deep learning in biometric recognition, there are many meaningful works on these topics [49,50]. Wu et al. [51] studied a novel bio-key generation algorithm based on machine learning, which was used to directly generate stable bio-keys for improving accuracy. Panchal et al. [52] proposed a support vector machine (SVM)-based ranking scheme without threshold selection to increase the accuracy. Pandey et al. [15] presented a DNN model to generate bio-keys with randomness. Roh et al. [16] combined a CNN framework and an RNN framework to produce bio-keys without helper data. Wang et al. [53] used a DNN architecture to learn biometric features for enhancing the stability of bio-keys. Roy et al. [17] used a CNN model to extract robust features for improving the accuracy. However, the above methods only focus on accuracy and ignore the security and privacy issues of the bio-key generation. Iurii et al. [54] designed an efficient approach for securing identification documents using deep learning, which can demonstrate high-accuracy performance while resisting biometric impostor attacks.

3. Methodology

In this section, we illustrate the proposed bio-key generation scheme. First, we give an overview of the proposed bio-key generation mechanism in Section 3.1. Then, we introduce two components of our biometrics mapping network: feature vector extraction and binary code mapping networks in Section 3.2. Next, we present the implementation of random permutation and fuzzy commitment in Section 3.3. Finally, we describe the enrollment and reconstruction processes of whole bio-key generation in Section 3.4.

3.1. Overview

The overall framework of the proposed bio-key generation mechanism via deep learning is shown in Figure 2. It mainly consists of the enrollment stage and reconstruction stage. (1) In the enrollment stage, we use a random binary code generator comprised of *RNG* to produce the binary code K, and then train a biometrics mapping network to learn the mapping between the original biometric data and random binary code. Specifically, this network includes two components: feature extraction and binary code mapping. Next, the elements of the binary code are shuffled by using a random permutation module to yield a permuted code K_R as the bio-key, meanwhile, the generated permutation vector (PV) is stored in the database. Finally, K and K_R are encoded to generate auxiliary data (AD) via a fuzzy commitment encoder. Therefore, the PV and AD are only stored in the database during the enrollment process. (2) In the reconstruction stage, a query image is input to the trained network model to generate the corresponding binary code K^*. Subsequently, we obtain the stored PV and AD from the database. Next, the query permuted code $K_R{}^*$ is generated from the predicted binary code by utilizing the random permutation module with PV. Finally, the bio-key K_R is decoded with the help of AD when the query image is close to the registered biometric image. Otherwise, the bio-key cannot be restored. In the next section, we describe the biometrics mapping network in detail.

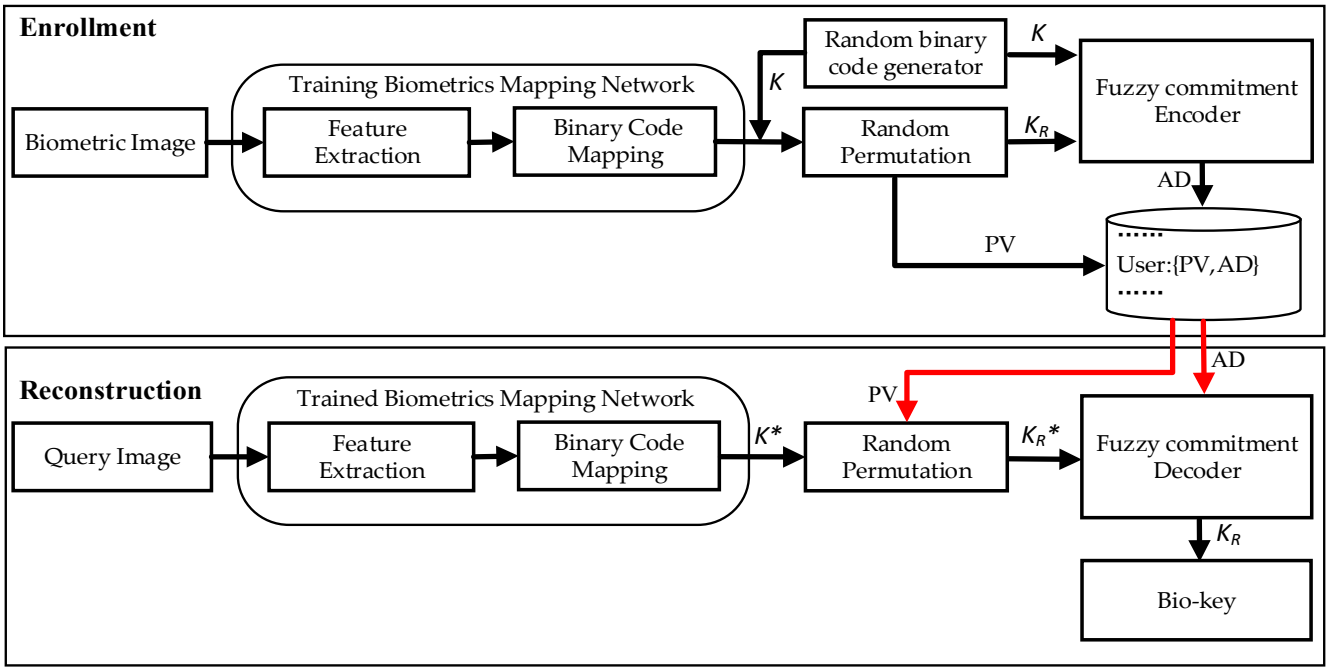

Figure 2. An overview of our proposed bio-key generation mechanism. (1) Enrollment stage: a pair of generated PV and AD are stored in database for each user; (2) Reconstruction stage: the final bio-key is recovered with the helper data of the PV and AD.

3.2. Biometrics Mapping Network Based on DNN Architecture

As DNNs [13,50] have made great progress in the field of image recognition, Taigman et al. [55] and Deng et al. [56] proposed a biometric recognition model based on a DNN framework which can effectively learn intermediate feature representation from the biometric image. Although these methods have satisfactory performance, there are still two challenges while applying them in the real-word. The first challenge is that these models with large weight parameters require greater computing power of the device. The second challenge is that directly learning random binary code from biometric images needs a robust feature extractor. To overcome these challenges, we propose a biometrics mapping network based on DNN architecture which contains two components: feature extraction network and binary code mapping network. The former is used to extract an intermediate feature vector, and the latter is to map the extracted feature vector into binary code. This architecture is shown in Figure 3. In the next section, we introduce two components of the biometrics mapping network.

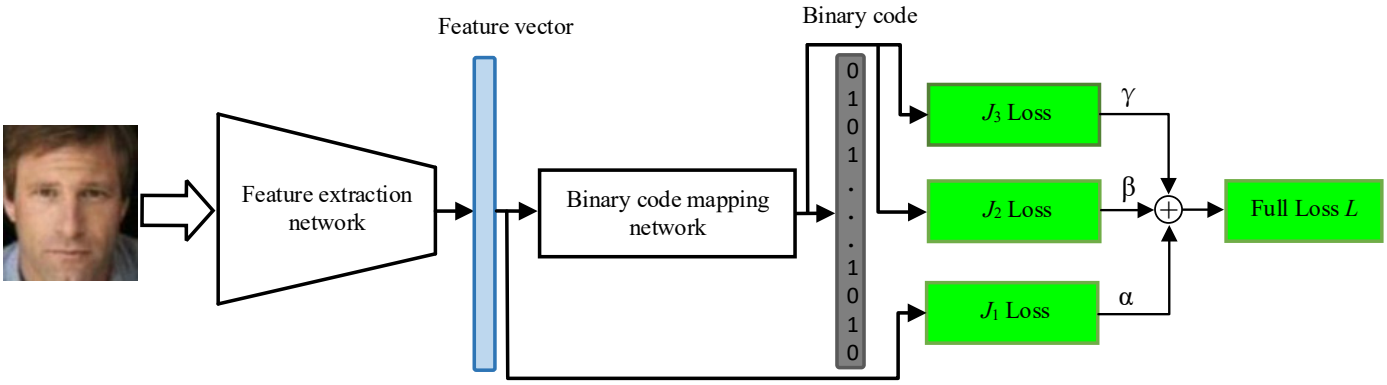

Figure 3. The framework of our proposed biometrics mapping network based on a DNN for generating binary code. This architecture consists of a feature extraction network and a binary code mapping network.

3.2.1. Feature Extraction Network

To solve the first challenge, we adopt pointwise (PW) and depthwise (DW) convolutions instead of standard convolution to build a lightweight feature extraction network which can reduce the amount of memory storage and computational power while preserving accuracy [57]. On this basis, we improve the bottleneck architecture for a better intermediate feature representation. The architecture of the network is shown in Figure 4. Specifically, on the one hand, we first use PW to expand input features into a higher-dimensional feature space for extracting rich feature maps, and then utilize DW to reduce the computation redundancy. On the other hand, we add an attention module named a squeeze-and-excitation network (SENet) [58] between two nodes of the bottleneck, which can selectively strengthen useful features and suppress useless features or less useful ones for improving the ability of feature representation. Therefore, these key components can complement each other, resulting in an effective and robust biometric feature vector.

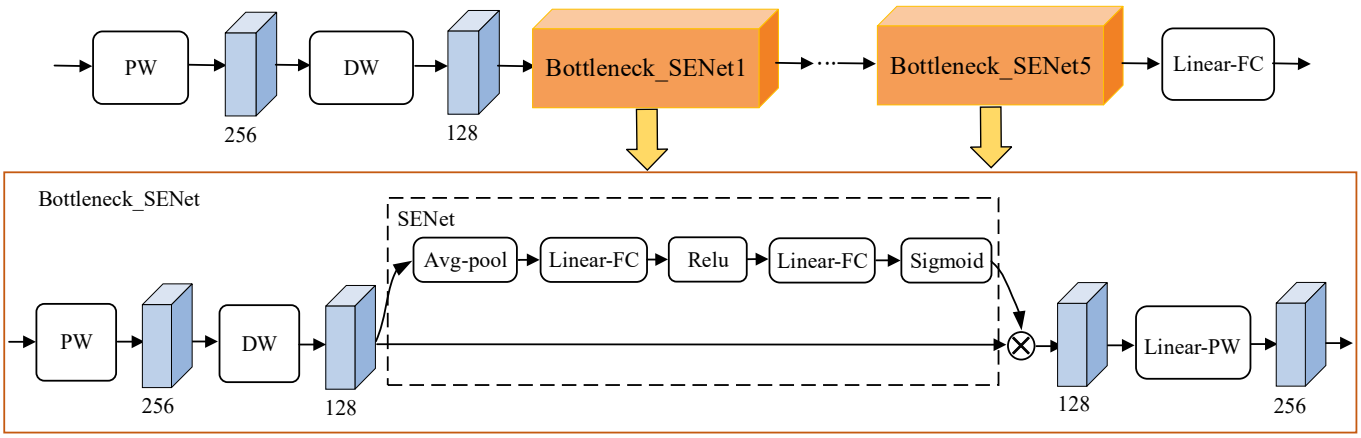

Figure 4. The architecture of feature extraction network.

3.2.2. Binary Code Mapping Network

To effectively learn the mapping between face image and random binary code, we design a robust binary mapping network. In fact, the mapping network is to learn unique binary code, which follows a uniform distribution. In other words, each bit of this binary code has a 50% chance of being 0 or 1. Since the extracted feature vector can represent the uniqueness of each face image, our proposed method only requires a nonlinear project matrix to map the feature vector into the binary code. Assuming that the extracted feature vector can be defined as V and the nonlinear project matrix can be defined as M, the mapped binary code K can thus be denoted as:

$$K = M^T V \tag{1}$$

Therefore, we can combine a sequence of fully connected (FC) layers with a nonlinear activate function to establish nonlinear mapping, such as Equation (1). The mapping network contains three FC layers (namely FC_1 with 512 dimensions, FC_2 with 2048 dimensions, FC_3 with 512 dimensions) and one tanh layer. For different bio-key lengths, we slightly modify the dimension of the FC_3 layer. Moreover, a dropout strategy [59] is applied to these FC layers with a 0.35 probability to avoid overfitting. The tanh layer is used as the last activation function for generating approximately uniform binary code. This is because the tanh layer is differentiable in the backpropagation learning and close to the signum function.

It is noted that each element of the mapped real-value Y through the network may be close to 0 or 1 where $Y \in \mathbb{R}^l$. In this case, we adopt binary quantization to generate binary code from Y. To obtain the uniform distribution of the binary code, we set a dynamic

threshold $\overline{Y} = \frac{1}{l}\sum_{i=1}^{l} Y_i$ where Y_i denotes i-th element of Y, and l represents the length of Y. Therefore, the final mapping element K_r of binary code K can be defined as:

$$K = [K_1, \ldots, K_r \ldots, K_l] = [q(Y_1), \ldots, q(Y_r) \ldots, q(Y_l)] \tag{2}$$

Here, quantization function $q(Y_r)$ is defined as:

$$q(Y_r) = \begin{cases} 1, & if\ Y_r \geq \overline{Y} \\ 0, & \text{otherwise} \end{cases} \quad \text{where } l \geq r \geq 1 \tag{3}$$

3.2.3. Training Network

As mentioned above, our proposed DNN model includes two components: feature extraction and binary code mapping. So as to effectively learn the mapping between biometric image and random binary code, we combine three objective functions to implement an end-to-end training network.

First, for the feature extraction component, we use ArcFace loss [55] as classification loss to train this component, which is used to generate a discriminative feature vector for the user's face image. Hence, the first objective function J_1 is expressed by the ArcFace loss. Second, for the binary code mapping component, the output of this network is an l-dimensional binary code; this is actually a regression task. To reduce the quantization loss between mapping real-value Y and binary codes K, the second loss is defined as:

$$J_2 = \frac{1}{N}\sum_{i=0}^{N} \|Y_i - K_i\|^2 \tag{4}$$

where N denotes the batch size, and i represents i-th sample. Moreover, the binary code is high entropy, that is, the distributions of 0 and 1 are equal probability. To maximize the entropy of the binary code, the third objective function is selected as:

$$J_3 = \frac{1}{N}\sum_{i=0}^{N} \|\text{mean}(Y_i) - 0.5\|^2 \tag{5}$$

where mean (Y_i) donates average of Y_i. Therefore, the final loss L can be defined as:

$$L = \alpha\, J_1 + \beta J_2 + \gamma J_3 \tag{6}$$

where α, β, and γ are the scale factor of each term, respectively.

We give the implementation process in Algorithm 1. In our training network process, binary codes K are firstly assigned by the random binary code generator module according to different users. Then, we can set up the mapping relationship between biometric images X and binary codes K. Next, we initialize the weight parameter W and bias parameters b, and the full objective function as the mentioned Equation (6) is adopted to train our network. Subsequently, multiple pairs of X and K are fed into the DNN model to update parameters W and b by using a stochastic gradient descent (SGD) method. Finally, the parameters are computed to obtain the trained model parameters. All steps are presented in Equation (1). To improve the security for preventing information leakage, a random binary code is assigned to each user and used as the label data to train the biometrics mapping model based on the DNN framework. Therefore, during every new enrolment, we should assign a new random binary code to a new subject, and then retrain the network to learn the mapping between the new biometric image and binary code, which can provide better accuracy and security.

Algorithm 1 Process of training network in our DNN model

Parameters: learning rate η, epoch size N, weight parameter W and bias paramters b
Input: biometric images X as input data, the assigned binary code K as label data
Output: the trained DNN model with W and b

1. Generate binary codes K by random binary code generator according to different users. Then, establish mapping a relationship between input X and output K.
2. Initialize W and b.
3. Compute loss function according to Equation (6).
4. Update W and b by SGD:

$$W^{(i+1)} = W^{(i)} + \eta \nabla W$$
$$b^{(i+1)} = b^{(i)} + \eta \nabla b$$

$i = i$ +1
Until $i < N$
5. Output W and b.

3.3. Random Permutation and Fuzzy Commitment

In this section, we introduce two functional modules: random permutation and fuzzy commitment. For the revocability, we utilize the random permutation module to shuffle the elements of binary code for generating distinctive bio-keys. Furthermore, a fuzzy commitment is constructed to generate stable bio-key while avoiding information leakage of bio-key and biometric data.

3.3.1. Random Permutation

Random permutation can shuffle the elements of binary code, which can obtain different bio-keys. On the one hand, once the generated bio-key is compromised, a new bio-key can be easily regenerated by only modifying the random permutation seed. On the other hand, it can reduce the cost-time of bio-key generation instead of retraining the DNN model. PV is firstly produced by *RNG* with random permutation seed for each user, let the produced PV $= \left\{P^1, P^2, P^3 \cdots P^r, \cdots P^{l-1}, P^l\right\}$, which satisfies $\forall t = \{1 \cdots l\}$, $\exists t = P^r$. Given a binary code K with length l where $K = \{K_1, K_2, K_3 \cdots K_r, \cdots K_{l-1}, K_l\}$, a permuted binary code $K_R = \{K_{P1}, K_{P2}, K_{P3} \cdots K_{Pr}, \cdots K_{Pl-1}, K_{Pl}\}$ is obtained by randomly shuffling the elements of K according to PV. In addition, PV is stored in the database during the enrollment stage. It can be observed from random permutation that PV does not reveal any information of the binary code or bio-key. This is because that PV is only determined by random permutation seed and has no relationship with binary code. In the bio-key reconstruction stage, the permuted binary code is generated by only shuffling the elements of query binary code through the stored PV in the database. As shown in Figure 5, we give an example of random permutation when l = 256.

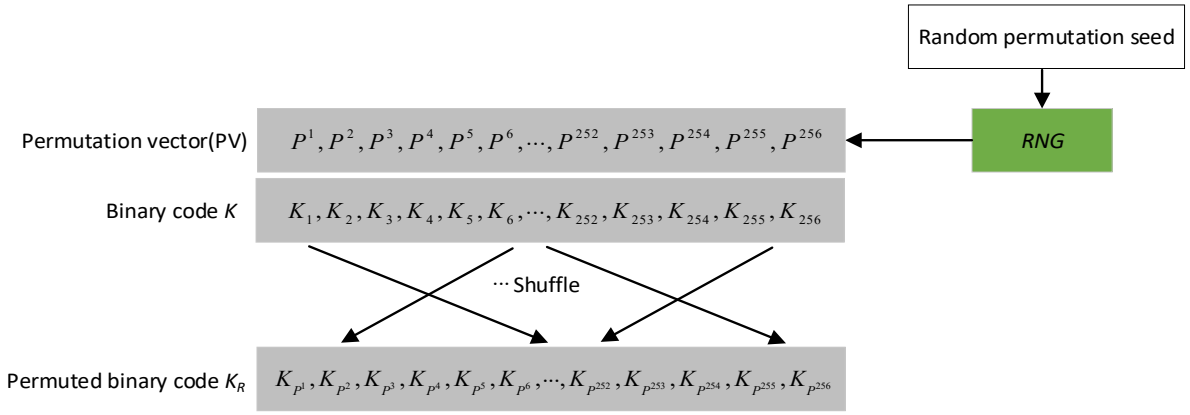

Figure 5. Illustration of the random permutation where the length of binary code is 256.

3.3.2. Fuzzy Commitment

Considering the distance error between the predicted binary code and the corresponding random binary code during the reconstruction stage, we can correct the predicted binary code using an error correction code (ECC) technique for obtaining a stable bio-key. Inspired by the fuzzy commitment, we construct a new fuzzy commitment to extract a robust bio-key from the binary code and adopt BCH code as the encoder and decoder of ECC in Figure 6. There are two advantages in our fuzzy commitment. The first is that the error correction code is utilized to reconstruct a stable bio-key from the predicted binary code during the decoder stage. The second is that generated helper data does not reveal the biometric data or bio-key during the encoder stage.

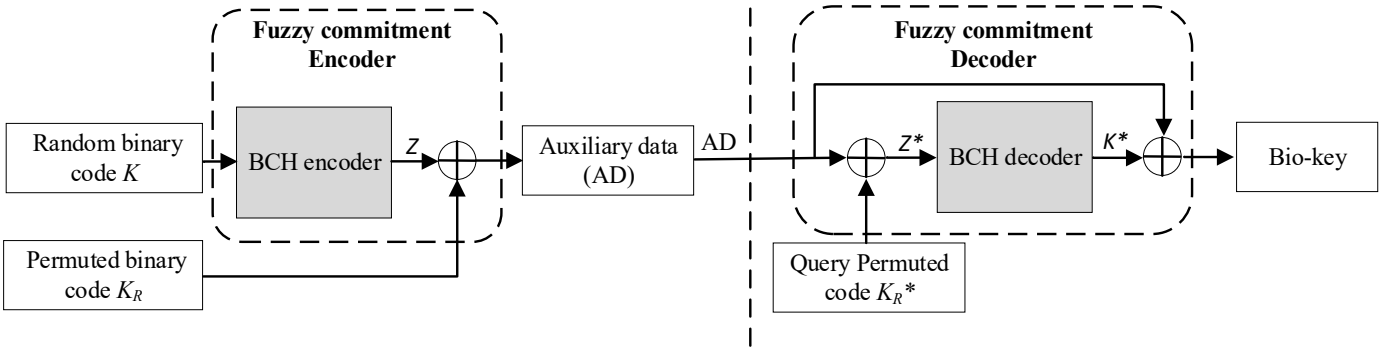

Figure 6. The implementation process of the fuzzy commitment. During the encoder stage, binary codes K_R and K are encoded to auxiliary data AD through the fuzzy commitment encoder. During the decoder stage, the bio-key is decoded with the help of the AD and query permuted code K_R^* through the fuzzy commitment decoder.

In the next part, we describe how to construct the encoder of the fuzzy commitment from random binary code K and the permuted binary code K_R during the encoder stage. Then, we introduce how the decoder of the fuzzy commitment is designed to recover the bio-key from query permuted code $K_R{}^*$ during the decoder stage.

Encoder stage: Random binary code K is encoded by BCH to generate a binary codeword Z:

$$Z = ENC(K) \tag{7}$$

where $ENC()$ means the BCH encoder of the fuzzy commitment. Then, the generated Z is XORed with the permuted binary code K_R to generate AD:

$$\mathrm{AD} = Z \oplus K_R = ENC(K) \oplus K_R \tag{8}$$

where $\oplus$ denotes XOR operation and K_R is generated from K through random permutation module. Since K_R and K are random, the AD does not expose any biometric data and bio-key although it is public.

Decoder stage: Query permuted code K_R^* is XORed with AD to yield Z^*. Then, Z^* is decoded to generate the corresponding binary code K^* through the BCH decoder:

$$K^* = DEC(Z^*) = DEC(\mathrm{AD} \oplus K_R^*) \tag{9}$$

where $DEC()$ denotes the BCH decoder of fuzzy commitment and K_R^* is obtained from query biometric image through DNN and random permutation modules. Finally, the bio-key $\hat{K_R}$ is reconstructed by XOR operation between K^* and AD:

$$\hat{K_R} = K^* \oplus \mathrm{AD} = DEC(\mathrm{AD} \oplus K_R^*) \oplus \mathrm{AD} \tag{10}$$

If the distance error $|\varepsilon|$ between permuted binary codes K_R and K_R^* is less than the error tolerance threshold τ of the BCH code, the reconstructed $\hat{K_R}$ is equal to K_R. It can be deduced as the following formula:

$$\begin{aligned} \hat{K_R} &= DEC(\text{AD} \oplus K_R^*) \oplus \text{AD} \\ &= DEC(ENC(K) \oplus K_R \oplus K_R^*) \oplus \text{AD} \\ &= DEC(ENC(K) \oplus \varepsilon) \oplus \\ &= K_R \end{aligned} \tag{11}$$

The constructed fuzzy commitment has two characteristics: reliability and security. In terms of reliability, the bio-key can be correctly recovered when $|\varepsilon|$ of a genuine query is less than τ. On the contrary, the bio-key cannot be correctly reconstructed when $|\varepsilon|$ of imposter query is larger than τ, which has the ECC ability and increases the stability of the bio-key. In terms of security, the generated AD does not reveal biometric data or the bio-key because input binary codes K_R and K are uniformly distributed during the encoder stage, which can enhance the security of bio-key generation.

3.4. Enrollment and Reconstruction Procedure

In this section, we present the process of enrollment and reconstruction in Figure 2. In the enrollment stage, the generated random binary code K is firstly assigned to each user. At the same time, the biometric images X as input data and random binary code K as label data are used to train the DNN model. Then, PV is obtained by *RNG*, and the binary code K is shuffled to produce the permuted binary code K_R according to PV. Next, the binary codes K_R and K are encoded into AD by using the fuzzy commitment encoder module. Finally, the generated PV and AD are stored in the database. In our method, only AD and PV are registered to the database instead of biometric data, which can effectively prevent information leakage about biometrics from AD and PV.

In the bio-key reconstruction stage, we adopt the trained DNN model to generate binary code K^* from the query image. Then, the permuted binary code K_R^* is obtained from K^* by random permutation according to PV. Next, the bio-key can be correctly decoded from K_R^* and AD by utilizing the fuzzy commitment decoder module when K_R^* is close to K_R for genuine queries. Therefore, the final bio-key K_R is recovered.

4. Experimental Results

In this section, we introduce datasets and experimental setup in Sections 4.1 and 4.2, respectively. Then, we conduct our experiments to evaluate the accuracy performance in Section 4.3. Subsequently, we analyze revocability and randomness properties in Section 4.4. Next, we discuss the security of our proposed scheme in Section 4.5. Furthermore, we compare our approach with related works in Section 4.6. Finally, our proposed method is applied to the data encryption scenario for validating its effectiveness and practicality in Section 4.7.

4.1. Dataset

We adopt multi-shot enrolment including more than one image to evaluate our method on the following three benchmark datasets.

(1) ORL dataset [60]: this dataset comes from Olivetti Research Laboratory formerly named American Telephone and Telegraph Company. This dataset is composed of 10 different face images of each 40 face subjects, which includes different illuminations, expressions, and poses. In addition, we randomly select five face images of each subject for enrollment, and other face images are applied to test the performance of bio-key generation during the reconstruction stage.

(2) Extended YaleB dataset [61]: this dataset includes 2332 face images of 38 subjects, and it is captured under 64 different lighting conditions. Hence, the face image of

each user has 64 different illuminations. We randomly choose 10 face images of each subject in the enrollment stage, and the rest images are used for testing.

(3) CMU-PIE dataset [62]: the CMU-PIE dataset contains 41,368 face images of 68 subjects, including larger variations in illuminations, poses, and expressions. In this experiment, we utilize five different poses (p05, p07, p09, p27, p29) and illuminations to validate our scheme. We follow the same partition strategy with the Extended YaleB dataset in training and testing images.

4.2. Experiment Setup

In this experiment, we train the DNN model during the enrollment stage on ORL, Extended YaleB, and CMU-PIE datasets, respectively. Before training our DNN model, we adopt the MTCNN [63] model to implement image alignment operation, and then take the center crop operation to generate the final face image of 112×96 from the aligned image so that the input size to the network is consistent. After the alignment and crop operations, we train our DNN model to generate bio-keys from the pre-processed face images. In the training process, α, β, and γ are set to 0.25, 0.25, and 0.5. The batch size is 64 and the learning rate is 0.0001. We use a SGD optimizer to train our network with 80,000 epochs. It is noted that five different trained DNN models are generated by only modifying the output dimension of the FC_3 layer. Then, these cropped images are fed into the trained DNN models to generate 128-, 256-, 512-, 1024-, and 2048-bit binary codes. Next, these binary codes produce PV and AD for registering into the database. In the reconstruction stage, test images are submitted as the inputs of the bio-key generation model for testing. Our experiment environment is: Window10, 64 bits, CPU: and Intel(R) Core(TM) i7-9750H. In addition, all verifications of our proposed scheme are implemented in Pycharm IDE.

4.3. Accuracy Performance

In this section, we discuss the accuracy performance on ORL, Extended YaleB, and CMU-PIE datasets. The accuracy is evaluated by GAR (Genuine Accept Rate) and EER (Equal Error Rate) at a given 1% FAR (False Acceptance Rate). The results on these three datasets are listed in Table 1. It can be seen that all GARs are larger than 97% at a fixed 1% FAR, and EERs are less than 2% where the key length of bio-key is varied from 128, 256, 1024, and 2048. Generally, since the generated binary code contains more noise when the key length is longer, it may reduce the authentication accuracy of the bio-key. However, as the key length increases, our proposed scheme also achieves better accuracy. For example, we obtain a better GAR of 99.62% and EER of 0.51% at fixed 1% FAR for a length size of 1024 bits on the ORL dataset. This is because that our method based on the DNN model can effectively learn the mapping relationship between binary code and biometric image. In general, accuracy performance can be well preserved under the different key lengths and the intra-user variations.

Table 1. Accuracy performance under different key lengths on three benchmark datasets.

Length	ORL		Extended YaleB		CMU-PIE	
	GAR	EER	GAR	EER	GAR	EER
128	99.12%	0.75%	99.40%	0.83%	97.97%	1.49%
256	99.40%	0.70%	99.28%	0.86%	98.34%	1.35%
512	99.59%	0.52%	99.29%	0.85%	98.06%	1.47%
1024	99.62%	0.51%	99.31%	0.86%	98.47%	1.09%
2048	99.22%	0.72%	99.30%	0.85%	98.43%	1.29%

To further illustrate the accuracy performance of our methods, we compare our approach with other state-of-art methods [15,64–66] on Extended YaleB and CMU-PIE dataset. Tables 2 and 3 show the comparison results on the Extended YaleB and CMU-PIE datasets, respectively. On the one hand, our method can earn a GAR of 99.40% for 128 bits binary code, which outperforms the Genetic-ECOC method [64] at GAR at a

fixed 1% FAR on the Extended YaleB dataset. On the other hand, references [64,66] may perform extremely well; they are close to our performance. However, their lengths of the codeword are both less than 90, and as such they offer lower security to brute force attacks. In Table 1, we can find that the GAR and EER of our method are 99.97% and 1.49%, respectively, for a length size of 128 on the CMU-PIE dataset. Thus, our algorithm outperforms reference [65] even if our length of the codeword is less than [65]. In summary, our method can achieve a higher GAR and a lower EER than Hybrid [65], BDA [66], MEB coding [15], and Genetic-ECOC [64] on the CMU-PUE dataset. The reason is that our proposed scheme adopts the DNN model based on feature extraction network and binary code mapping network to generate robust binary code, which can enhance the compactness of intra-class and discrepancy of inter-class. Our proposed method can effectively perform better accuracy than the state-of-the-art methods under the intra-user variations such as illumination, pose, and expressions. Therefore, our approach can enhance stability under the multi-shot enrolment.

Table 2. Accuracy comparison on Extended YaleB dataset.

Method	Length	GAR@1%FAR	EER
Genetic-ECOC [64]	72	93.42%	-
Our method	128	99.40%	0.83%

Table 3. Accuracy comparison on CMU-PIE dataset.

Method	Length	GAR@1%FAR	EER
Hybrid [65]	210	90.61%	6.81%
BDA [66]	76	96.38%	–
MEB coding [15]	1024	97.59%	1.14%
Genetic-ECOC [64]	88	97.01%	–
Our method	1024	98.47%	1.09%

4.4. Basic Property Analysis

In this section, we discuss two basic attributes, including randomness and revocability, which are the basic security requirement of the bio-key generation method. In the next part, we will analyze these security properties in detail.

4.4.1. Randomness Analysis

For a key to be secure, it needs to guarantee randomness, which can increase the computational complexity of brute force attacks. To validate this point, we test the randomness of the generated bio-keys by the standard NIST statistical test suite. Firstly, we randomly select 1000 face images to generate the corresponding bio-keys, and then these generated bio-keys can be sequentially formed into 1000×512 binary bits as a source file, which is larger than 10,000 bits. Next, the source file is fed into NIST statistical test suite for evaluating randomness. Finally, according to the test benchmark, 16 tests of randomness are generated. According to NIST recommendations, if the p-value of 16 tests is larger than 0.01, the binary bit of the bio-key is random. The computed p-values of 16 tests are all greater than 0.01 in our scheme. This is because that generated bio-key is derived from random binary code. Overall, our generated bio-keys pass the randomness test and meet the randomness property.

4.4.2. Revocability Analysis

The revocability denotes a new bio-key can be regenerated when the original bio-key is leaked. Meanwhile, it should ensure that the new bio-key has no relation with the leaked bio-key, which can prevent the attacker from successfully accessing other authentication systems through the leaked bio-key or biometric data. Therefore, we will analyze the revocability of our proposed approach. As is illustrated in Section 3.3.1, we adopt the

random permutation module to generate different PV by modifying a random permutation seed for each user, which can shuffle the assigned binary code to generate the new bio-key. Therefore, an old bio-key can be replaced by a new bio-key by setting a different random permutation seed. In addition, there is no relationship between the new bio-key and the old bio-key because the generated bio-key is only determined by random PV.

To further demonstrate the effectiveness of the revocability in our scheme, we conduct some experiments on ORL, Extended YaleB, and CMU-PIE datasets. The mated distances are computed by Hamming Distance (HD), which are from two generated bio-keys of intra-user with different random permutation seeds. On the contrary, the genuine distances are calculated by HD matching from intra-user with the same permutation seed. If the mated HDs are both close to $l/2$, where l denotes the key length, it indicates that the new bio-key is fully dissimilar to the old bio-key. Therefore, we can collect genuine and mated HDs from ORL, Extended YaleB, and CMU-PIE datasets, respectively. As shown in Figure 7, it can be observed that the distributions of mated and genuine HD are distinguishable on three datasets at l = 1024, 2048. In addition, the mated HDs are mainly distributed around half of the key length. Specifically, HD between the new bio-key and the old bio-key from the same user is larger than the error-correcting range of the BCH decoder. Through the above phenomenon, we can conclude two points. The first point is that the new key and the old bio-key of intra-user are fully independent and dissimilar from each other. The second point is that our scheme can approximatively produce 100% revocability of the bio-key. For example, if l = 2048, there are 2^{2048-1} different revocations of the bio-key. Hence, our proposed scheme satisfies the revocability property.

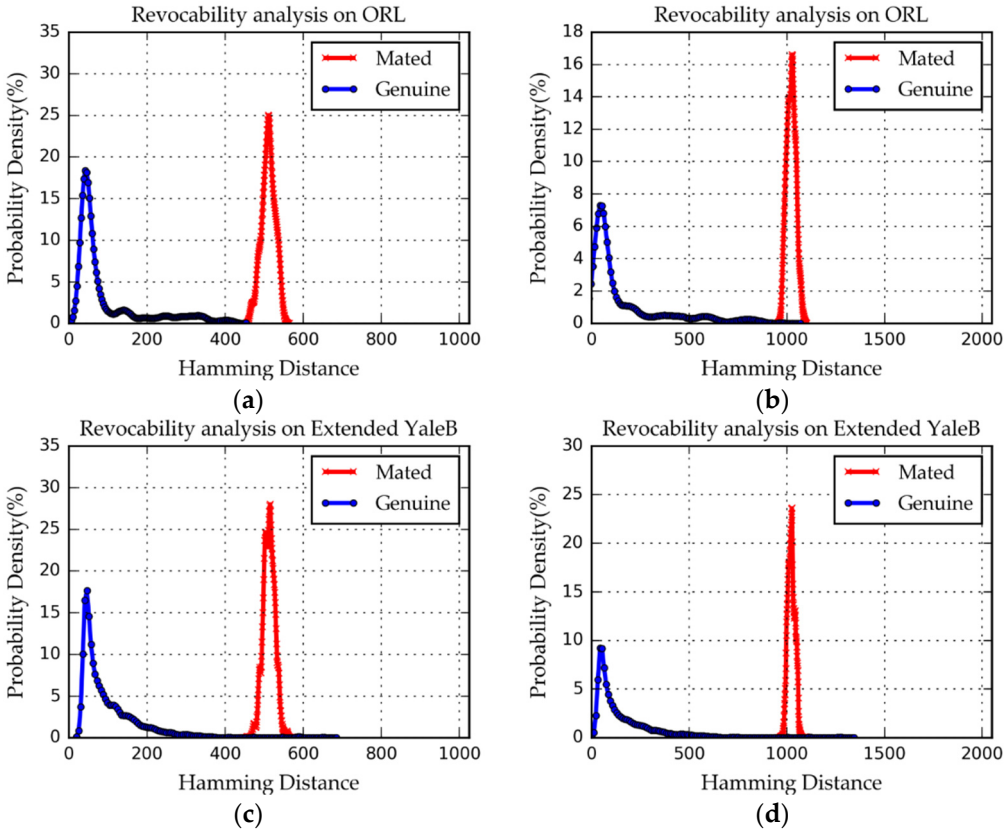

Figure 7. *Cont*.

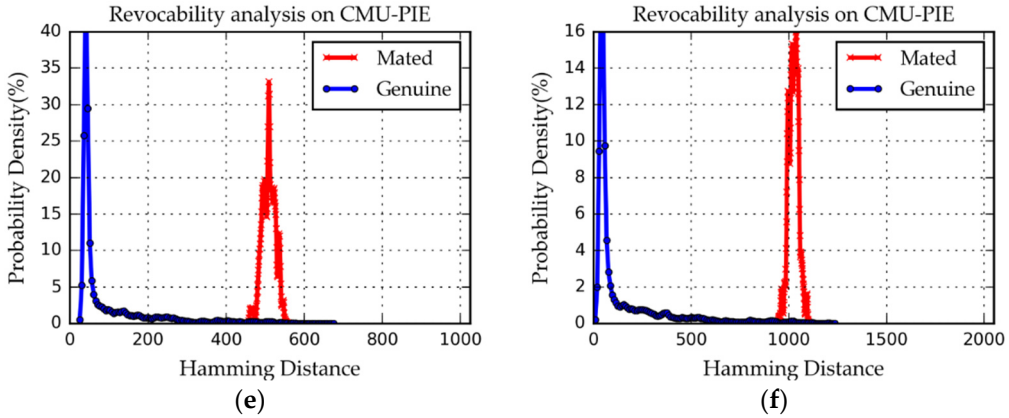

Figure 7. The distributions of Hamming Distance between the new bio-keys and the old bio-keys for intra-user, applied on (**a**) ORL at $l = 1024$; (**b**) ORL at $l = 2048$; (**c**) Extended YaleB at $l = 1024$; (**d**) Extended YaleB at $l = 2048$; (**e**) CMU-PIE at $l = 1024$; (**f**) CMU-PIE at $l = 2048$.

4.5. Security Analysis

In practical application, we should consider the security of generated bio-key in different attack scenarios. In this part, we analyze attacks at information leakage about biometric data and bio-keys for our system. In addition, we discuss some other attacks including cross-matching attack, correlation attack, and guessing mapped binary code attack in detail.

4.5.1. Resisting Information Leakage Attacks

In the worst-case scenario, we assume that attackers can obtain intermediate information in our proposed system. Moreover, our algorithm is public to attackers. There are two points where information is leaked as follows: (1) trained network parameters, (2) PV and AD stored in the database. We will analyze the security according to these two points.

(1) Trained network parameters: In the trained DNN model, there are a large number of weight and bias parameters, which are used to achieve the mapping of the biometric image to binary code. Since network parameters are only combined with the input biometric image to forward predict binary code, the information of biometric data and bio-key is not revealed from the network parameters. In the case of the known algorithm with network parameters, the attackers can use a large number of imposter samples as input to yield a false acceptance in brute forcing. Actually, this attack exploits the vulnerability of a biometric system in false acceptance. If the system has low distinguishability between genuine and imposter samples, the attacker can easily access the system under a false acceptance. Thus, the FAR of the system under this attack scenario is a satisfactory evaluation metric.

To verify this point, we utilize the trained DNN model with parameters to generate binary code under the aforementioned attack. The distributions between genuine and imposter matching distance for all other user samples other than the genuine is considered as the imposter. As shown in Figure 8, it can be seen that the HD distribution of inter-subjects is close to half of the key length. Meanwhile, the HD distribution of intra-subjects is about 15% of the key length. Thus, our model can recognize individuals well when the error-correcting range is set to approximately 15% of the key length. According to the generation rule of BCH codewords, we can make the error-correcting capacity slightly larger than 15% of the key length, which can also maintain the recognition accuracy at FAR = 0. In summary, the error-correcting range of the BCH can recognize individuals in our scheme. In other words, it can be observed that the distributions between the imposter and genuine matching distance are fully distinguishable on ORL, Extended YaleB, and CMU-PIE datasets, specifically, which indicates the FAR is close to zero under this attack. Hence, it proves the false acceptance attack is difficult and biometric information is not leaked in this situation.

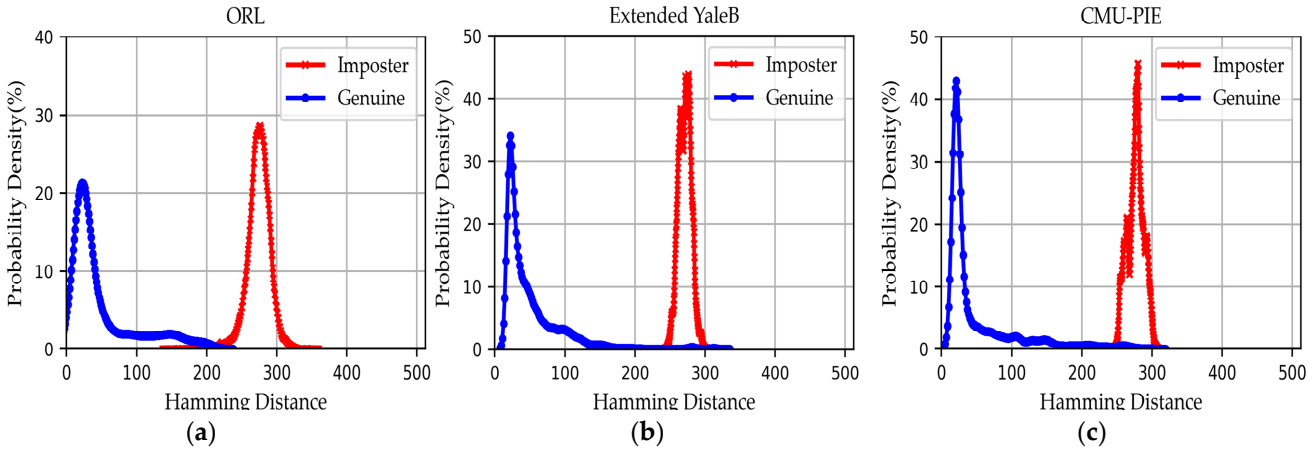

Figure 8. The distributions of Hamming Distance between the imposter and the genuine, applied on (**a**) ORL dataset at $l = 512$; (**b**) Extended YaleB at $l = 512$; (**c**) CMU-PIE at $l = 512$.

(2) PV and AD stored in the database: In this case, it is assumed that the PV and AD for each user in the database are obtained by attackers. Firstly, the PV is generated by *RNG* in the random permutation module, and the binary code K is randomly shuffled to generate bio-key by using PV. It can be concluded that PV is independent of the binary code and bio-key. Even if the PV is compromised, sensitive biometric information is not leaked. Secondly, the AD is produced from binary codes K_R and K by using the fuzzy commitment encoder. It is noted that the elements of the K_R and K follow a uniform distribution. Hence, the attacker cannot infer the element of random binary code through the element correlation of AD. References [24,25] proposed the statistical attacks to retrieve the biometric data and bio-key by analyzing the correlation of multiple helper data for the same user. However, the extracted binary codes from the same user are independent of each other in our proposed approach. Although multiple ADs are public to the attacker, it is also difficult to retrieve the binary code by using statistical attacks. This is because the binary codes are randomly assigned by *RNG* with different random seeds for the same user, which leads to no correlation between multiple ADs. Therefore, it is practically impossible to retrieve biometric data and bio-keys when PV and AD are compromised.

4.5.2. Resisting Other Attacks

In this section, we will discuss brute force attack, cross-matching attack and guessing mapped binary code attack.

(1) Brute force attack: In this attack scenario, an attacker directly tries to guess a bio-key. It is assumed that no a priori information about the bio-key is public to the attacker. To obtain the corrected bio-key, the attacker has no choice but to guess the bio-key in brute forcing. If the length l of bio-keys is known to the attacker, it will take 2^l trials to obtain the corrected bio-key when the bio-keys are random and dissimilar. For instance, the attacker conducts 2^{1024} attempts when the length $l = 1024$. Therefore, it is difficult to guess the bio-key by using an electronic computer in this situation. It can be noted that the randomness of bio-keys is important against the brute force attack. In our proposed approach, the bio-key for each user is derived from random binary code. Further, the randomness of bio-keys has been passed in Section 4.4.1. Therefore, bio-keys of inter-users are random and independent of each other, and it is impractical to guess the bio-key.

(2) Cross-matching attack: In this attack scenario, we assume that the attacker can obtain a bio-key of a user, which can be used to perform cross-matching across multiple biometric databases. In our proposed scheme, to generate multiple discriminative bio-keys for the same user, we only modify permutation seed to produce random PVs in the enrollment stage, which can be deployed to multiple databases. Hence, the generated bio-keys of intra-users are also random and dissimilar for each other. The compromised bio-key cannot be linked to other biometric systems. In addition, we can easily regenerate

the PV to obtain a new bio-key by setting different random permutation seeds in the compromised system. In other words, our bio-key generation method has revocability that ensures the distinguishability and renewability of the bio-keys for intra-user, which has been analyzed in Section 4.4.2. Hence, our proposed approach can effectively resist cross-matching attack.

(3) Guessing mapped binary code attack: In our bio-key reconstruction process, the bio-key is generated from mapped binary code. It is assumed that PV and AD stored in the database are known to an attacker. Meanwhile, it is tested if an attacker can correctly guess the mapped binary code. Based on the above assumption, the bio-key can be restored by utilizing the guessed binary code, PV, and AD of the user through our scheme. To analyze the effectiveness of our proposed approach against the attack, we measure the degree of freedom (N) (i.e., uncertainty) of the binary code from one sample of a user to an inter-sample of other users. Firstly, the binary code is generated from the sample of a user, and other binary codes are generated from inter-samples of other users. The normalized Hamming distances are calculated from the generated binary codes on ORL, Extended YaleB, and CMU-PIE datasets. Then, we compute the mean (μ) and the standard deviation (σ) from the normalized Hamming distances, respectively. Next, the degrees of freedom N of the mapped binary code for the user in the three datasets are obtained as ($N = \frac{\mu(1-\mu)}{\sigma^2}$). Moreover, our ECC coding can correct approximately 10 percent of bit error. If $N = 512$ and $C = 0.1 \times N \approx 51$, the attacker actually has $2^{N-C} = 2^{461}$ brute force (BF) trials. Finally, we can compute a conservative theoretical bound BF by using the bound formula, it can be defined as:

$$BF = 2^{N-C} \tag{12}$$

where N is the degree of freedom, C is the error-correction bits, and BF is the brute force trials. Therefore, we can compute N of the mapping binary code for the user in Table 4. There are 2^{417}, 2^{387}, and 2^{366} trials for three datasets, respectively. Thus, it is difficult to guess the correctly mapped binary to restore the bio-key for a user.

Table 4. Degree of freedom computation for the mapped binary code at the length of 1024 bits.

Dataset	μ	σ	Degree of Freedom (N)	BF Trials
ORL	0.4996	0.0232	464	2^{417}
Extended YaleB	0.4934	0.0241	431	2^{387}
CMU-PIE	0.4980	0.0247	407	2^{366}

4.6. Comparison with Related Works

In this section, we compare our approach with related works in the aspect of security. From Table 5, we compare with the related works [17,29,65,66]. Now, we will analyze the mentioned methods in detail.

Table 5. Comparisons with other related works.

Method	Biometrics	Storage Data	Technique Scheme	Resist Information Leakage
Li et al. [29]	Fingerprint	chaff points	fuzzy vault	NO
Chauhan et al. [67]	Iris	Helper data	fuzzy commitment	NO
Roy et al. [17]	Retinal	Biometric template	DNN	NO
Asthana et al. [68]	Fingerprint	Helper data	Key binding	NO
Ours	Face	AD and PV	DNN	Yes

Li et al. [29] proposed alignment-free fingerprint cryptosystem based on fuzzy vault. Since the stored data contains the biometric feature, an attacker can retrieve biometric data. Chauhan et al. [67] improved the fuzzy commitment scheme. If parameters of S and P are compromised, the biometric template and bio-key can be retrieved by using stored data in a database. Roy et al. [17] put forward a DNN model to generate the bio-key from the

retinal image. However, if the extract biometric template is compromised, the new template cannot be regenerated. Asthana et al. [68] designed a new key binding with biometric data via objective function. Nevertheless, the security actually depends on the predefined threshold value. If this threshold value is obtained, the stored helper data may reveal biometric data. Therefore, these methods are vulnerable to information leakage attack.

To avoid the information leakage of biometric data and bio-key, we combine DNN architecture and fuzzy commitment to generate the bio-key. During enrollment, we assign random binary code to each user, which is shuffled via PV to generate permuted code. Next, the binary codes are encoded by the fuzzy commitment module to yield AD. Finally, PV and AD are stored in the database. During bio-key reconstruction, we utilize the DNN to generate binary code, and the final bio-key is restored through our scheme. It can be observed that input binary codes of the fuzzy commitment encoder are uniformly distributed and are not related to biometric data during the enrollment stage. Hence, it does not expose biometric data and bio-key when PV and AD are public. Additionally, we can modify the permutation seed to update a new pair of PV and AD for the same user, which can also resist cross-matching attack.

Overall, our proposed approach is more robust against information leakage attack and cross-matching attack. Moreover, to meet the different sizes of the key in the encryption application, our model can be easily modified by only setting the output dimension in the DNN architecture, which means that our accuracy can be well preserved in different key lengths.

4.7. Application

4.7.1. Experiment Platform

To validate the flexibility and practicality, we apply our model to a real-world data encryption scenario. In this application, we use the computer system to test the performance of cost time and accuracy in data encryption application. This test system contains a 64-bit CPU with Intel(R) Core(TM) i7-9750H, and a USB camera with a resolution of 640 $\times$ 480. a face image is captured by the USB camera then the captured face data and plaintext are entered in the test model to verify its performance. We can utilize the PC display to observe the test result.

4.7.2. Experiment Dataset

In this application scenario, we use face image as the input biometric image and alice29.txt from Canterbury corpus as plaintext to test the performance on the user computer. For biometric images, we adopt the USB camera to collect face images of different users in a real environment, which are composed of 10 different face images with 640 $\times$ 480 resolution of each of the 40 face subjects. These datasets, including different illuminations, expressions, and poses, are aligned to obtain the cropped face images with the resolution of 112 $\times$ 96 by using MTCNN [63]. In addition, we randomly select five face images of each subject for training our key generation model, and other face images are used for testing. For alice29.txt of plaintext file, it consists of various characters, and the size of the file is 152,089 bytes.

4.7.3. Experiment Process

To meet the different key lengths of standard AES based on cipher block changing (CBC), our key generation model is trained to generate 128-, 192-, and 256-bit bio-keys from the collected dataset during the training stage. For different sizes of bio-keys, we only make a slight modification on the output dimension in the DNN architecture. Subsequently, our key generation model with trained DNN parameters is deployed into the user computer.

As is shown in Figure 9, we give the encryption and decryption process. Our test model consists of face detection, key generation model, AES encryption, and decryption modules. In the encryption process, we adopt the USB camera to obtain a face image of the user; then, this face image is detected through the face detection module by MTCNN.

Next, a bio-key is generated by our trained key generation model. At the same time, the plaintext is submitted as an input of the AES encryption module. Finally, we perform data encryption with the inputs of the bio-key and plaintext to yield a ciphertext. In the decryption process, we also obtain different face images of the same user to regenerate the bio-keys, which are used to decrypt the ciphertext for generating plaintext. Therefore, we compare the data consistency.

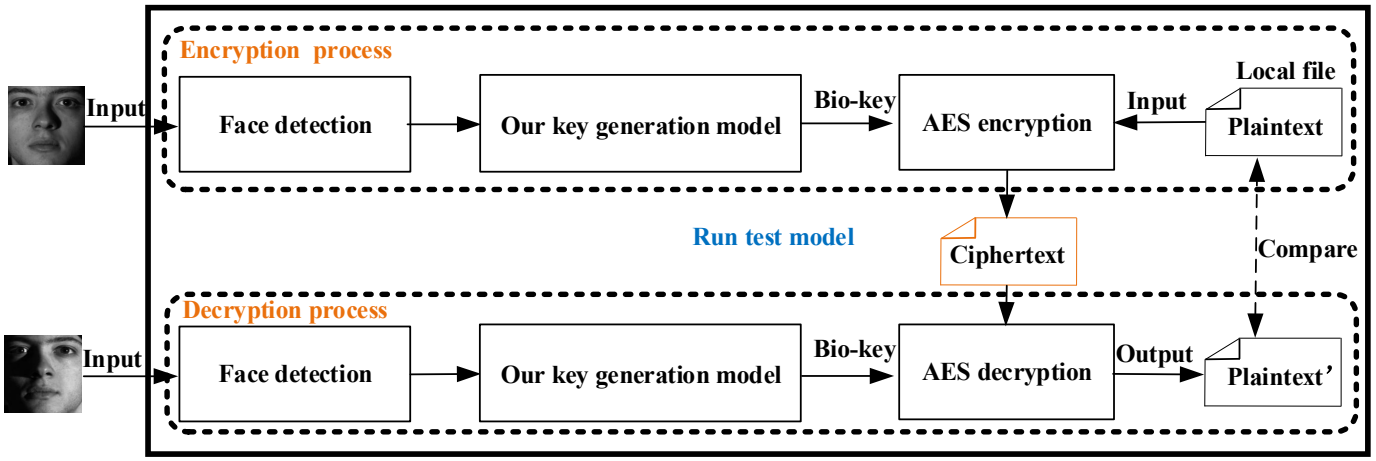

Figure 9. Illustration of encryption and decryption process on the personal computer platform.

4.7.4. Results Analysis

In this section, we analyze accuracy and time consumption in the data encryption application. In the aspect of accuracy, there are 200 × 199 = 39,800 comparisons at different key lengths in data encryption and decryption processes. Among them, genuine comparisons have 3800 trials and imposter comparisons have 36,000 trials. The verification results are listed in Table 6. Our GAR@1%FAR is close to 100% under different key lengths. However, the several obtained face images include large pose changes and partial occlusion, which slightly reduce the accuracy. In practical application, face images are generally obtained when a user maintains a normal posture. Therefore, this accuracy is acceptable.

Table 6. Verification results from different users in data encryption and decryption.

Length	GAR@1%FAR	EER
128	99.92%	0.05%
192	99.96%	0.03%
256	99.95%	0.04%

In the aspect of time consumption, the cost time of data encryption and decryption is shown in Figure 10. The face detection and key generation modules account for nearly half of the total time consumption for the key lengths of 128, 192, and 256 bits. This is because these two modules both adopt DNN architecture, thus they are more time-consuming than the AES encryption module. In general, the time for once encryption and decryption is within 250 ms on this application. Specially, the time of key generation is less than 130 ms of 128, 192, and 256 key lengths. For the data encryption application, this cost consumption is worthwhile, and our scheme is effective and practical for real-world bio-key applications.

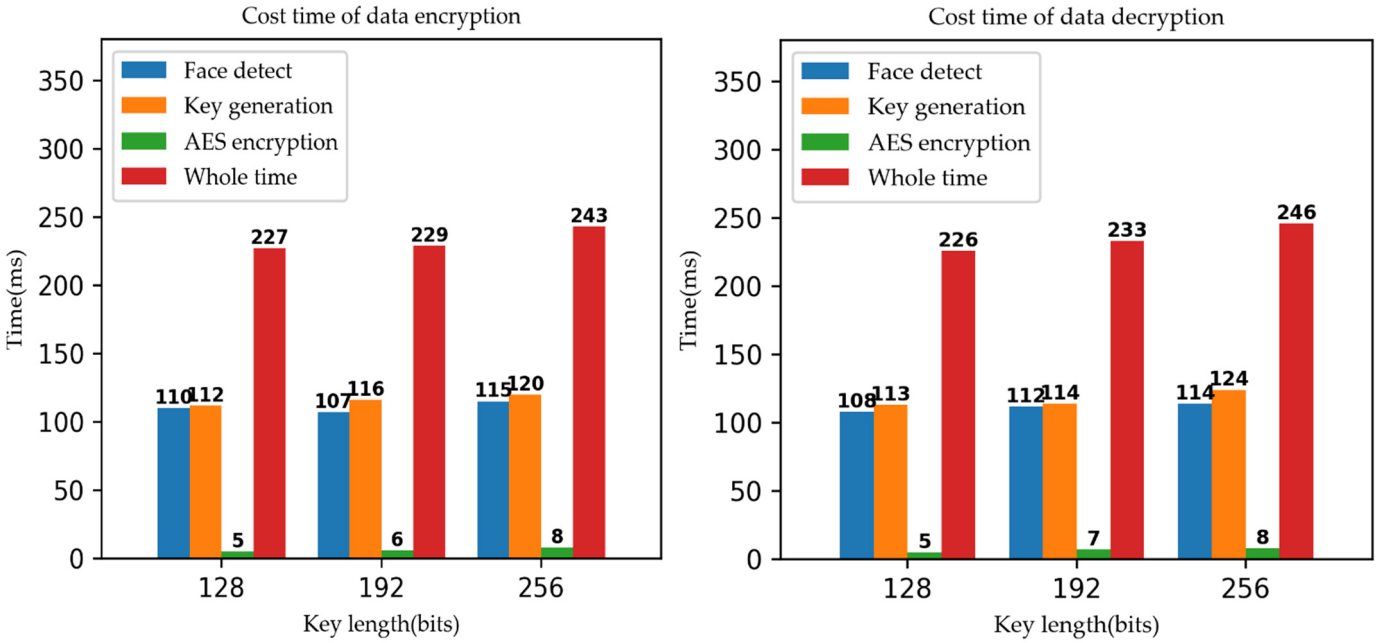

Figure 10. Time cost of data encryption and decryption: (**a**) Time cost of several modules in data encryption process at different key lengths; (**b**) Time cost of several modules in data decryption process at different key lengths.

5. Conclusions

In this paper, we propose a secure bio-key generation scheme based on deep learning. Firstly, to improve the security for preventing information leakage, a random binary code is assigned to each user. Moreover, the biometrics mapping model based on the DNN framework is designed to map the biometric images into diverse binary codes for different users. Secondly, the random permutation is adopted to shuffle the random binary code by modifying the permutation seed for protecting privacy and guaranteeing revocability. Thirdly, to generate a stable and secure bio-key, we construct a new fuzzy commitment module. Furthermore, our scheme was applied to the data encryption scenario for testing its practicality and effectiveness. Through the analysis of the experimental results, on the one hand, our scheme can effectively enhance security and privacy while maintaining accuracy performance. On the other hand, the security analysis illustrates our scheme not only satisfies the properties of revocability and randomness of bio-keys, but resists various attacks such as information leakage attack, brute force attack, cross-matching attack, and guessing mapped binary code attack. However, our method has a limitation without retraining the network. In other words, it is not suitable for zero-shot enrollment. Since the generated bio-key needs to be unique, reliable, and random, it is difficult to ensure that the trained DNN model meets the above three properties without retraining. We will focus on how to improve stability and security under zero-shot enrollment in the future work.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W. and B.L.; software, Y.W.; validation, Y.W.; formal analysis, Y.W.; resources, Y.W.; writing—original draft, Y.W.; writing—review & editing, B.L., Y.Z., Q.M. and J.W.; supervision, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the ShenZhen Science Technology and Innovation Commission (SZSTI): JCYJ20170817115500476.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html] (accessed on 9 September 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hassen, O.A.; Abdulhussein, A.A.; Darwish, S.M.; Othman, Z.A.; Tiun, S.; Lotfy, Y.A. Towards a Secure Signature Scheme Based on Multimodal Biometric Technology: Application for IOT Blockchain Network. *Symmetry* **2020**, *12*, 1699. [CrossRef]
2. Karimian, N. *Cardiovascular PPG Biometric Key Generation for IoT in Healthcare Domain*; Mobile Multimedia, Image Processing, Security, and Applications; SPIE: Baltimore, MD, USA, 2019; pp. 1–7.
3. Wazid, M.; Das, A.K.; Kumari, S.; Li, X.; Wu, F. Provably secure biometric-based user authentication and key agreement scheme in cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 4103–4119. [CrossRef]
4. Sheng, W.; Chen, S.; Xiao, G. A Biometric Key Generation Method Based on Semisupervised Data Clustering. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1205–1217. [CrossRef]
5. Karimovich, G.S.; Turakulovich, K.Z. Biometric cryptosystems: Open issues and challenges. In Proceedings of the International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2–4 November 2016; pp. 1–3.
6. Jain, A.K.; Nandakumar, K. Biometric template security. *EURASIP J. Adv. Signal Process.* **2008**, *1*, 1–20. [CrossRef]
7. Ignatenko, T.; Willems, F. Information leakage in fuzzy commitment schemes. *IEEE Trans. Inform. Forens. Secur.* **2010**, *5*, 337–348. [CrossRef]
8. Kholmatov, A.; Yanikoglu, B. Realization of correlation attack against the fuzzy vault scheme. In Proceedings of the Security, Forensics, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, 27 January 2008; pp. 28–30.
9. Sarkar, A.; Singh, B.K. A review on performance, security and various biometric template protection schemes for biometric authentication systems. *Multimed. Tools Appl.* **2020**, *79*, 27721–27776. [CrossRef]
10. Smith, A. Maintaining Secrecy when Information Leakage Is Unavoidable. Ph.D. Thesis, Massachusetts Institute of Technology, McGill University, Montreal, QC, Canada, 2004.
11. Dodis, Y.; Smith, A. Correcting errors without leaking partial information. In Proceedings of the 37th annual ACM symposium on Theory of computing, Baltimore, MD, USA, 22–24 May 2016; pp. 654–663.
12. Linnartz, J.P.; Tuyls, P. *New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates*; Springer: Guildford, UK, 2003; Volume 2688, pp. 393–402.
13. Qi, W.; Jinxiang, L.; Luc, C.; Zhengguo, Y.; Liang, L.; Wenyin, L. A Novel Feature Representation: Aggregating Convolution Kernels for Image Retrieval. *Neural Netw.* **2020**, *130*, 1–10.
14. Labati, R.D.; Munoz, E. Deep-ECG: Convolutional neural networks for ecg biometric recognition. *Pattern Recognit. Lett.* **2019**, *126*, 78–85. [CrossRef]
15. Pandey, R.K.; Zhou, Y.; Kota, B.U. Deep Secure Encoding for Face Template Protection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June 2016; pp. 77–83.
16. Roh, J.H.; Cho, S. Learning based biometric key generation method using CNN and RNN. In Proceedings of the 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia, 24–26 July 2018; pp. 1–4.
17. Roy, N.D.; Biswas, A. Fast and robust retinal biometric key generation using deep neural nets. *Multimed. Tools Appl.* **2020**, *79*, 6823–6843. [CrossRef]
18. Juels, A.; Wattenberg, M. A fuzzy commitment scheme. In Proceedings of the 6th ACM Conference on Computer and Communication Security, Kent Ridge Digital Labs, Singapore, 1–4 November 1999; pp. 28–36.
19. Juels, A.; Sudan, M. A fuzzy vault scheme. In Proceedings of the IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 June 2002; pp. 1–13.
20. Hao, F.; Anderson, R. Combining crypto with biometrics effectively. *IEEE Trans. Comput.* **2006**, *55*, 1081–1088.
21. Veen, M.; Kevenaar, T.; Schrijen, G.J. Face biometrics with renewable templates. *SPIE* **2006**, *6072*, 1–13.
22. Chauhan, S.; Sharma, A. Fuzzy Commitment Scheme based on Reed Solomon Codes. In Proceedings of the 9th International Conference on Security of Information and Networks, Newark, NJ, USA, 20–22 July 2016; pp. 96–99.
23. Zhou, X.; Kuijper, A.; Veldhuis, R. Quantifying Privacy and Security of Biometric Fuzzy Commitment. In Proceedings of the 2011 International Joint Conference on Biometrics, Washington, DC, USA, 11–13 October 2011; pp. 1–8.
24. Rathgeb, C.; Uhl, A. Statistical attack against iris-biometric fuzzy commitment schemes. In Proceedings of the Computer Vision and Pattern Recognition Workshops (CVPRW), Colorado Springs, CO, USA, 20–25 June 2011; pp. 25–32.
25. Rathgeb, C.; Uhl, A. Statistical attack against fuzzy commitment scheme. *Biometrics Iet.* **2012**, *1*, 94–104. [CrossRef]
26. Clancy, T.C.; Kiyavash, N. Secure smartcardbased fingerprint authentication. In Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications, Berkley, CA, USA, 8 November 2003; pp. 45–52.
27. Uludag, U.; Jain, A. Securing fingerprint template: Fuzzy vault with helper data. In Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, New York, NY, USA, 17–22 June 2006; pp. 1–8.

28. Nandakumar, K.; Jain, A.K. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Trans. Inform. Forens. Secur.* **2007**, *2*, 744–757. [CrossRef]
29. Li, C.; Hu, J. A security-enhanced alignment-free fuzzy vault-based fingerprint cryptosystem using pair-polar minutiae structures. *IEEE Trans. Inform. Forens. Secur.* **2015**, *11*, 543–555. [CrossRef]
30. Scheirer, W.J.; Boult, T.E. Cracking fuzzy vaults and biometric encryption. In Proceedings of the Biometrics Symposium, Baltimore, MD, USA, 11–13 September 2007; pp. 1–6.
31. Hartloff, J.; Bileschi, M. Security analysis for fingerprint fuzzy vaults. In Proceedings of the Conference on biometric and surveillance technology for human and activity identification X, Baltimore, MD, USA, 31 May 2013; pp. 1–12.
32. Tams, B.; Mihăilescu, P.; Munk, A. Security considerations in minutiae-based fuzzy vaults. *IEEE Trans. Inform. Forens. Secur.* **2015**, *10*, 985–998. [CrossRef]
33. Zhang, W.; Chen, T. Generalized optimal thresholding for biometric key generation using face images. In Proceedings of the IEEE International Conference on Image Processing, Genova, Italy, 14 September 2005; pp. 1–4.
34. Hoque, S.; Fairhurst, M.C.; Howells, W. Evaluating Biometric Encryption Key Generation Using Handwritten Signatures. In Proceedings of the 2008 Bio-inspired, Learning and Intelligent Systems for Security, Edinburgh, UK, 4–6 August 2008; pp. 17–22.
35. Rathgeb, C.; Uhl, A. An iris-based Interval-Mapping scheme for Biometric Key generation. In Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, Salzburg, Austria, 16–18 September 2009; pp. 511–516.
36. Lalithamani, N.; Soman, K.P. An Efficient Approach For Non-Invertible Cryptographic Key Generation From Cancelable Fingerprint Biometrics. In Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala, India, 27–28 October 2009; pp. 47–52.
37. Wu, L.; Liu, X.; Yuan, S. A novel key generation cryptosystem based on face features. In Proceedings of the IEEE International Conference on Signal Processing, Beijing, China, 24–28 October 2010; pp. 1675–1678.
38. Ranjan, R.; Singh, S.K. Improved and innovative key generation algorithms for biometric cryptosystems. In Proceedings of the IEEE International Advance Computing Conference, Ghaziabad, India, 22–23 February 2013; pp. 943–946.
39. Sarkar, A.; Singh, B.K.; Bhaumik, U. RSA Key Generation from Cancelable Fingerprint Biometrics. In Proceedings of the 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 7–18 August 2017; pp. 1–6.
40. Anees, A.; Chen, Y.P.P. Discriminative binary feature learning and quantization in biometric key generation. *Pattern Recognit.* **2018**, *77*, 289–305. [CrossRef]
41. Dodis, Y.; Ostrovsky, R. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.
42. Chang, E.-C.; Qiming, L. Hiding Secret Points Amidst Chaff. In Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Petersburg, Russia, 28 May–1 June 2006; pp. 59–72.
43. Sutcu, Y.; Li, Q.; Memon, N.D. Secure Biometric Templates from Fingerprint-Face Features. In Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–6.
44. Li, Q.; Chang, E.C. Robust, short and sensitive authentication tags using secure sketch. In Proceedings of the 8th Workshop on Multimedia and Security, Geneva, Switzerland, 26–27 September 2006; pp. 56–61.
45. Lee, S.W.; Li, S.Z. Fuzzy Extractors for Minutiae-Based Fingerprint Authentication. *Adv. Biom.* **2007**, *4642*, 760–769.
46. Yang, W.; Hu, J.; Song, W. A Delaunay Triangle-Based Fuzzy Extractor for Fingerprint Authentication. In Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, UK, 25–27 June 2012; pp. 66–70.
47. Chi, C.; Wang, C.; Yang, T. Optional multi-biometric cryptosystem based on fuzzy extractor. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, Xiamen, China, 19–21 August 2014; pp. 1–6.
48. Alexandr, K.; Anastasia, K.; Anna, U. New Code Based Fuzzy Extractor for Biometric Cryptography. In Proceedings of the International Scientific-Practical Conference Problems of Infocommunications. Science and Technology, Kharkiv, Ukraine, 9–12 October 2018; pp. 119–124.
49. Wang, Q.; Liu, X.; Liu, W.; Liu, A.-A.; Liu, W.; Mei, T. MetaSearch: Incremental Product Search via Deep Meta-learning. *IEEE Trans. Image Process.* **2020**, *29*, 7549–7564. [CrossRef]
50. Schroff, F.; Kalenichenko, D. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–10.
51. Wu, Z.; Tian, L. Generating stable biometric keys for flexible cloud computing authentication using finger vein. *Inform. Sci.* **2018**, *433*, 431–447. [CrossRef]
52. Panchal, G.; Samanta, D. A novel approach to fingerprint biometric-based cryptographic key generation and its applications to storage security. *Comput. Electr. Eng.* **2018**, *69*, 461–479. [CrossRef]
53. Wang, Y.; Li, B. A biometric key generation mechanism for authentication based on face image. In Proceedings of the 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), NanJing, China, 23–25 October 2020; pp. 1–5.
54. Iurii, M.; Farhad, S.; Leandro, C.; Nuno, G. Towards Facial Biometrics for ID Document Validation in Mobile Devices. *Appl. Sci.* **2021**, *11*, 1–15.

55. Taigman, Y.; Ming, Y.; Ranzato, M. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
56. Deng, J.; Guo, J.; Xu, E.N. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1–11.
57. Chen, S.; Liu, Y.; Gao, X.; Han, Z. MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. *arXiv* **2018**, arXiv:1804.07573v4.
58. Jie, H.; Li, S.; Gang, S. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023.
59. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. *Comput. Sci.* **2012**, *3*, 212–223.
60. Samaria, F.; Harter, A. Parameterisation of a stochastic model for human face identification . In Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision, Sarasota, FL, USA, 5–7 December 1994; pp. 138–142.
61. Phillips, P.J.; Moon, H.; Rizvi, S.A.; Rauss, P.J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660.
62. Sim, T.; Baker, S.; Bsat, M. The cmu pose, illumination, and expression (pie) database. In Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 21–21 May 2002; pp. 53–58.
63. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
64. Nazari, S.; Moin, M.S. A discriminant binarization transform using genetic algorithm and error-correcting output code for face template protection. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 433–449. [CrossRef]
65. Feng, Y.C.; Yuen, P.C. A hybrid approach for generating secure and discriminating face template. *IEEE Trans. Inform. Forens. Secur.* **2010**, *5*, 103–117. [CrossRef]
66. Feng, Y.C.; Yuen, P.C. Binary discriminant analysis for generating binary face template. *IEEE Trans. Inform. Forens. Secur.* **2012**, *7*, 613–624. [CrossRef]
67. Chauhan, S.; Sharma, A. Improved fuzzy commitment scheme. *Int. J. Inf. Technol.* **2019**, *1*, 1–11. [CrossRef]
68. Asthana, R.; Walia, G.S.; Gupta, A. A novel biometric crypto system based on cryptographic key binding with user biometrics. *Multimed. Syst.* **2021**, *3*, 1–15.