Construction and Implementation of Practical Reusable and Robust Fuzzy Extractors for Fingerprint

Lin You[a], Wang Cheng[a], and Gengran Hu[a]

[a]*School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, China*
mryoulin@gmail, chengwang1996@gmail.com, grhu@hdu.edu.cn

***Abstract*— Among the various authentication methods, biometrics provide good user friendliness. However, the non-renewability of biometrics leads to the problem that it might be stolen. The emergence of fuzzy extractors is a promising solution to this problem. The fuzzy extractors can extract uniformly distributed keys from various noise random sources (such as biometrics, physical unclonable functions and quantum bits). However, the research on fuzzy extractors mainly focuses on the theoretical level, and does not consider how the extracted biometrics should be coded and implementated. This paper first introduces a method of feature selection and encoding for fingerprints, together with a secure sketch based on Chebyshev distance in a rectangular coordinate system. Then we present the construction approach of reusable and robust fuzzy extractors(rrFE). Meanwhile, we prove that our secure sketch scheme has sufficient security. Finally, we also present the complete experimental process and a demo program, and test the performance of our proposed fuzzy extractors. Compared with other schemes, our scheme has lower storage overhead.**

***Index Terms*— Biometrics, Bio-cryptography, fingerprint, Secure Sketch, Fuzzy Extractors**

I. Introduction

In modern life, the Internet has become one of the most important and commonly used technologies for sharing or exchanging information. With the development of the Internet, accurate identity verification plays an crucial role in secure communications, such as login, authorization, and transactions. At present, the commonly used authentication method is usually through the user's secret information, such as a password, a possessed physical device (smart card, USB-Shield), or user's biological characteristics (fingerprints, facial images) [1]. However, the first two methods have obvious disadvantages: passwords can be forgotten or deciphered by attackers, physical devices need to be carried around, and there is a certain cost of production. As a result, the unique biometrics of each individual has received widespread attention.

Biometrics is very user-friendly when used, but they also have some security issues. How to securely store the biometric data used in the biometric identification and authentication system on the server is both vital and full of challenge. On the one hand, users will pay attention to whether their private information is safe to store on the server. On the other hand, due to the non-renewability of biometric features, once production is difficult to change, if the biometric data stored on the server is stolen, the attacker may use this information to recover the original biometric data, or even steal the user's data from other servers [2].

According to the Kerckhoff principle of cryptography, the security of a cryptographic system should depend on the security of the key. As a result, the password is very important to the security system because of its good randomness and uniform distribution. Meanwhile, biometrics is unique to each person and difficult to change. So we can combine cryptography with biometrics. If biometrics is to be used as passwords, they must be "uniform" and "precisely regenerative" [3], which is difficult to achieve in reality. Because of the interference caused by environmental factors or signal noise, even if a fixed random source is used as input, such as fingerprints, iris, human faces, there still exist subtle differences in the results of each sampling, which makes it difficult to use biometrics as a key. If there is a way to convert these slightly different inputs into "uniformly distributed" and "precisely reproduced" random strings, biometrics can be used as a secure key.

In the process of determining the solution to the above problem, the possibility of combining cryptography and biometrics are studied by the researchers. The traditional method of biometric protection is the feature transformation method, which manipulates the acquired biometric sets. In this approach, only the modified template is stored in the database and matched in the transform domain later. In the biometric cryptosystem, both of cryptography's and biometrics's advantages are used [4]. The key generating method is a combination of biometric technology and cryptography. The main obstacle to overcome in the biometric cryptosystem is the noise information contained in the biometric information. Because the biometric information changes due to the noise during each sampling, even the same biometric features of the same person, such as the fingerprint of the same finger, may be slightly different each time when they are collected, which means that the biometric information cannot be used directly as a key in cryptography. Therefore, when processing the transformed biometric template, it is necessary to introduce a fault-tolerant mechanism to deal with the noise. Up to now,

common bio-cryptography systems consist of fuzzy vaults, fuzzy commitment, source coding and fuzzy extractors.

The fuzzy extractor was first proposed by Dodis et al. [6] in 2004. It aims to extract strong passwords from other noisy information such as biometrics. The definition of the fuzzy extractor is $FE = (Gen, Rep)$. There are two algorithms, the generating algorithm Gen and the reproducing algorithm Rep. The generating algorithm Gen inputs the string w (one sampling of the noise random sources), outputs a string R and a public helper string P; the reproducing algorithm Rep inputs the input w'(another sampling of the same noise random sources) and public helper String P, output a string R'. If w and w' are close enough, $R' = R$, where R as a secret key can be applied to the encryption algorithm, and the public helper string P will not reveal any information about the original input w. However, they only proposed this concept and did not give a specific implementation method.

The fuzzy extractor proposed by Dodis contains two parts, a Stronger Extractor and a secure sketch. The Stronger Extractor is used to propose a uniformly distributed string as a secret key from the input information, which is commonly realized by hash function. The secure sketch is used for error correction, which is implemented by linear error-correcting codes.

In 2016, Canetti [7] et al. (Eurocrypt 2016) proposed a reusable fuzzy extractor. Their solution relies on a digital locker and adopts the Sample-then-Lock method. For the sampled input source w, a subset of w is randomly selected each time, and the generated key r is used to lock it. In their scheme, the key is randomly generated, so even if the entropy of the input source is very low, a randomly distributed key can be generated. However, the disadvantages are that the solution does not provide an implementation for sampling of input source and requires too much storage space. For 1024-bit input, 1TB of storage space is required to save the helper string.

In 2012, Yang et al. [8] used the unique characteristics of Delaunay triangulation to eliminate the pre-alignment problem. They extracted some reliable, distortion-tolerant, rotation-invariant and translation-invariant local features and use them to achieve alignment-free feature matching with the encryption of the fuzzy extractors. In 2017, Kaur et al. [9] proposed a fuzzy extractor based on multi-modal biometrics, which uses a hash function to extract keys from fingerprints and iris, and uses BCH codes to do error correction. However, this method needs to achieve accurate fingerprint alignment before applying the fuzzy extractor, and the performance of this algorithm is not enough to meet the needs of practical applications.

In 2017, Li et al. [10] assumed that the input source was a coded one-dimensional biometrics feature vector $X = \{x_1, \cdots, x_n\}$, dividing several intervals on the number axis. Each $x_i \in X$ is set to be a point on the number line, then the Chebyshev distance is used to implement the fuzzy extractor, and a biometrics protocol based on the fuzzy extractor is proposed. However, they assumed that biometrics can be encoded into a one-dimensional vector whose elements are all integers, which is not easy to do in reality.

There are some shortcomings in the above methods. On the one hand, they do not provide a simulation case combined with biometrics or only use fictitious biological data for experiments; On the other hand, the uncertainty of biometric data itself is not well studied. The large amount of uncertainty and variability in biometrics may cause the biometric data to be unable to obtain the correct codeword, which makes longer codewords to be needed to store more redundant information and further increase storage space overhead. Therefore, it is more important to reduce the inherent biometric uncertainty of the biometric cryptographic system in the feature extraction stage than to correct the uncertainty of biometric data by relying on error-correcting codes. At the same time, a secure sketch constructed through linear error-correcting codes has relatively low performance in the decoding stage and is difficult to be applied to mobile devices or IoT devices, so a lightweight error-correcting solution is required.

This paper is organized as follows: In Section II, we provide some preliminaries for our work. In Section III, we give our construction of secure sketch and a new fuzzy extractor. The security analysis is introduced in Section IV. In the next section, experiment simulation and comparison of our fuzzy extractors is done and presented. The last section summarizes the article.

II. Preliminaries

In this section, we briefly review some fundamentals about fuzzy extractors and related theories that we may use. We also describe notations used throughout this paper.

A. Min-Entropy and Average Min-Entropy

1) Metric Space: The metric space is a set M with a metric function or distance function. This function defines the distance between all elements in the set, which is called the metric on the set. In the previous work, Hamming distance, set difference and edit distance have been used to construct fuzzy extractors.

2) Min-Entropy: The security of the fuzzy extractor is related to the entropy of the output string. The min-entropy $\mathbf{H}_\infty(A)$ of the random variable A is defined as follows.

$$\mathbf{H}_\infty(A) = -\log\left(\max_a \Pr\left[A = a\right]\right). \tag{1}$$

The min-entropy describes the unpredictability of the result, which is only determined by the probability of the most likely result in a random variable. The higher the unpredictability of a system, the higher the min-entropy. As far as the security of a cryptographic system is concerned, the min-entropy can be related to the attacker's best chance of predicting a person's password in a guess.

3) Average Min-Entropy: For the conditional distribution, we use the concept of average min-entropy. The average min-entropy of random variable A under the condition of random variable B is defined as:

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(A) &= -\log\left(\mathbb{E}_{b\leftarrow B}\left[\max_a \Pr\left[A = a|B = b\right]\right]\right) \\ &= -\log\left(\mathbb{E}_{b\leftarrow B}\left[2^{-\mathbf{H}_\infty(A|B=b)}\right]\right).\end{aligned} \tag{2}$$

B. Fuzzy Extractors

1) Secure Sketch:

Definition 1: Secure sketch [1] is a component of the fuzzy extractors, including two algorithms, SS and Rec, described as follows:

TABLE I: Description of secure sketch

SS	Rec
1. Input $w \in$ M 2. Output $s \in \{0,1\}^*$	1. Input w', s 2. if dis $(w, w') \leq t$ 3. Output w

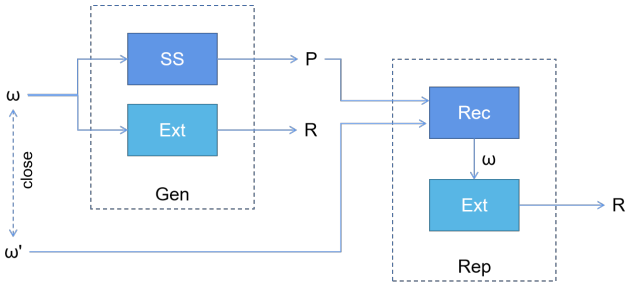


Fig. 1: The construction of fuzzy extractors

2) Fuzzy Extractors:

Definition 2: The fuzzy extractor consists of a pair of algorithms, a generating algorithm and a reproducing algorithm (Gen and Rep), which are described in detail as follows:

TABLE II: Description of Fuzzy Extractors

Gen	Rep
1. Input $w \in$ M 2. R = Ext(w) 3. P = SS(w) 4. Output (R, P)	1. Input w', P 2. if dis $(w, w') \leq t$ 3. w = SS. Rec(w', P) 4. R = Ext(w') 5. Output R

Correctness [11]: The correctness requirement is that if the distance between two samples of w and w' is close enough, then $R' = R$, which means R can be accurately reproduced.

Security [11]: The security requirement is that if the random source has enough entropy, then R is uniformly random.

C. Reusable and Robust Fuzzy Extractor

Robustness: The security of the fuzzy extractor only focuses on passive attacks. That is, the adversary knows the public helper string P but will not modify it. If the adversary tampers with P, then the recovery algorithm Rep of the fuzzy extractor is likely to get a wrong output R'. In order to solve the above problems, Boyen et al. [12] proposed a robust fuzzy extractor in 2005, which is described as follows: Gen$(w) \rightarrow (R, P)$, For all $P' \neq P$, Rep$(w', P') \rightarrow \perp$.

Reusability: Due to the irrevocability of biological characteristics, ordinary fuzzy extractors cannot guarantee that multiple secure keys can be extracted from the same biometrics. Boyen [13] proposed the concept of Reusable Fuzzy Extractor in 2004. The description is as follows: For multiple samples $w, w_1, \cdots, w_\rho$ from a random source, the Gen algorithm generates corresponding$(r, p), (r_1, p_1), \cdots, (r_\rho, p_\rho)$. Its security requirement is that for $i \neq j$, $p_i \neq p_j$, Rep$(w, P_j) \rightarrow r_j$. And there is still pseudo-randomness among all the rest r_i.

D. Symmetric Key Encryption

SKE:= (SKE. Enc, SKE. Dec) denotes the symmetric key encryption. SKE. Enc takes as input secret keys k and plaintext m and outputs ciphertext c. SKE. Dec decrypts the ciphertext c using the same secret key k to recover the plaintext m.

E. Chebyshev Distance

1) Norms: Norm can be used to express the length of a vector, or the distance between a vector and a zero point, or the distance between two points. The norm on the vector space V is represented by the symbol $||\cdot||$. For example, $||\boldsymbol{x}||$ represents the direct distance of each element of vector $\boldsymbol{x}$. L^p Norms are often used in pattern recognition to measure the direct distance between two biological information. Generally, L^p Norms is described as follows:

$$||\boldsymbol{x}|| = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}. \tag{3}$$

If $p = 1$, it is the L_1 norm, and $||\boldsymbol{x}||$ is the sum of the absolute values of the elements of the vector $\boldsymbol{x}$.

If $p = 2$, it is the L_2 norm, and $||\boldsymbol{x}||$ is the $\frac{1}{2}$ power of the sum of the squares of the elements of the vector $\boldsymbol{x}$, also known as the Euclidean norm.

If $p = \infty$, that is, the L_∞ norm $||\boldsymbol{x}||$ is the absolute value of the element with the largest absolute value of each element of the vector $\boldsymbol{x}$. The Chebyshev distance is an example of using the maximum norm.

2) Chebyshev Distance: If there are two vectors $\mathbf{X}$ and $\mathbf{Y}$, where $\mathbf{X} = (x_1, x_2, \cdots, x_n)$ and $\mathbf{Y} = (y_1, y_2, \cdots, y_n)$, then the Chebyshev distance between the two vectors is: dis$(\mathbf{X}, \mathbf{Y}) = \max(|x_i - y_i|), i = 1, \cdots, n$, corresponding to the L_∞ norm.

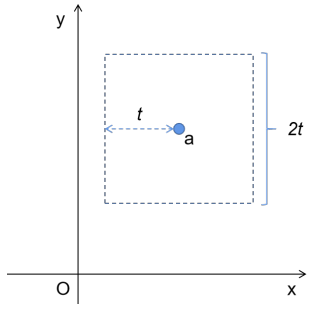


Fig. 2: Maximum Chebyshev distance t of point a

In the rectangular coordinate system, if the threshold t is given, the maximum Chebyshev distance of a point a is a square with the point as the center and $2t$ as the side length. As shown in 2, the Chebyshev distance from the center point

a to any point in the area enclosed by the square will not exceed the threshold t at the maximum.

To describe the construction of proposed schemes, we give some notations in Table III.

TABLE III: Notations

$\mathrm{dis}(x, y)$	a function returns Chebyshev distance between point x and point y
H	a hash function
SS	the secure sketch algorithm
Rec	the recovery algorithm of secure sketch
Gen	the generating algorithm of fuzzy extractors
Rep	the reproducing algorithm of secure sketch
B	the biometrics when register
B'	the biometrics when authorize
R	the key generated by fuzzy extractors
P	the public helper string of fuzzy extractors
v	the authenticator

III. Construction of Reusable and Robust Fuzzy Extractor for Fingerprint

A. Our Proposed Method

1) : We propose a fingerprint feature selection and representation method suitable for our fuzzy extractor scheme. In this method, the fingerprint center point is taken as the origin, and the minimum sampling radius R_{min} is defined. The ending of the fingerprint ridge closest to the origin outside the sampling radius is selected as the reference point, and a ray starting from the center point through the reference point is the polar axis to construct a polar coordinate system. The bifurcations and endings of the fingerprint are recognized as minutiae, and these points are also in the polar coordinate system. We select the first N minutiae with the polar diameter $\rho > R_{min}$, and compose the fingerprint feature vector $B = \{b_1, \cdots, b_n\}$ in the order of the polar angle from small to large.

2) : A secure sketch based on Chebyshev distance in a rectangular coordinate system is proposed, and a fuzzy extractor is constructed based on the secure sketch. Our fuzzy extractors scheme is both reusable and robust. Our solution is different from the previous fuzzy extractors constructed through error-correcting codes, and has the advantages of fast calculation speed and small storage space.

B. Fingerprint Preprocessing and Feature Extraction

In the preprocessing step, we follow the method of [14], [15]. First, the fingerprint area needs to be segmented from the image background. Divide the fingerprint image into $n \times n$ blocks, and then calculate the mean value and variance value of the intensity in each block to segment the fingerprint area from the graphic background. After that, the fingerprint image is enhanced to produce a fingerprint skeleton image. Here, Cross Number method is used to extract the two minutiae of fingerprint, the ending and bifurcation point, as features. The fingerprint processing step and the detected minutiae are shown in Figure 3.

C. Feature Selection

Since the sampled fingerprint image is affected by external factors such as noise, the extracted minutiae contain a large number of pseudo minutiae (such as the edge part of the fingerprint image, the fuzzy part in the image), so the extracted feature points need to be selected. At the same time, in order to match the input of the fuzzy extractor, it is necessary to express the selected feature points in an appropriate form.

1) Selection of Center Point: The center point is defined as the pixel point corresponding to the maximum value in the curvature field of the image, that is, the point in the image where the direction of the ridge changes the most. It is located at the progressive center of the fingerprint ridge, and the surrounding ridges tend to be semicircular. At present, the most commonly used method to obtain the center point is the Poincare index method [16]. Under normal circumstances, because the center point obtained by this method is not accurate enough, we do not use it as a reference point directly for feature extraction. Instead we use the center point as a reference to select several minutiae.

2) Minutiae Selection : At the same time, if the distance between the minutiae and the center point is too close, measurement errors may occur, and the minutiae should be screened. When there are too many selected minutiae, the verification time will increase. On the other hand, too few minutiae will weaken the uniqueness of the fingerprint. Therefore, an appropriate number of minutiae should be obtained so that it can represent the characteristics of the entire fingerprint.

Here, we first define the minimum sampling radius R_{min}. Then, the ending of the fingerprint ridge closest to the center point outside the sampling radius is selected as the reference point.

We take the center point as the pole, start from the center point, and draw a ray to the end point as the polar axis to establish a polar coordinate system. Taking the counterclockwise direction as the positive, then the coordinates of the center point is $(0, 0)$, and the coordinates of any minutia i is (ρ_i, θ_i). For minutia i, if ρ_i satisfies the following conditions, it will be a candidate minutia.

$$\rho_i > R_{min}. \tag{4}$$

At this time, for all candidate minutiae m_d according to the distance from the center point, the nearest N minutiae are selected and added to the selection queue.

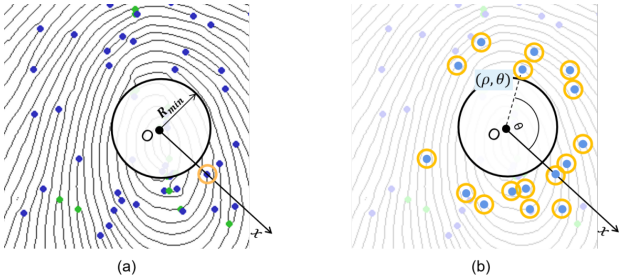
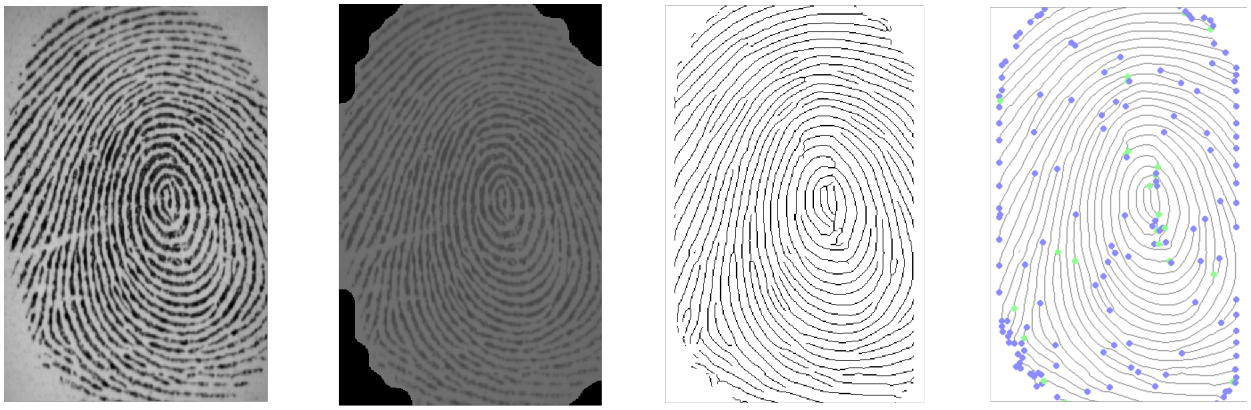


Fig. 4: (a) Select the sampling radius and establish a polar coordinate system based on the closest ridge ending (b) Select the minutiae



(a) (b) (c) (d)

Fig. 3: (a)Original fingerprint image, (b)Fingerprint image after background segmentation, (c)Fingerprint skeleton, (d)Fingerprint minutiae detected.

3) The Order of Minutiae: The minutiae in the selection queue are sorted according to the value from small to large. If there are two or more minutiae with θ value within the given angle threshold δ, the minutiae with smaller ρ are sorted first. Finally, a set of minutiae $m = (m_1, \cdots, m_n)$ is formed, where N is the number of selected effective minutiae. $m_i = (\rho_i, \theta_i)$, ρ_i is the distance from the center point, θ_i is the deflection angle relative to the polar axis.

D. Construction of Secure Sketch

1) Rectangular Coordinate System: In our secure sketch, the error correction algorithm will be implemented based on the Chebyshev distance in a rectangular coordinate system, which is different from the traditional method based on error-correcting codes, such as Reed-solomon codes and Bose–Chaudhuri–Hocquenghem codes. At the same time, we will express biological features in a rectangular coordinate system, and each feature is a point in the coordinate system. First, we define a rectangular coordinate system $C_{a,k,v}$ as follows.

Definition 3: $(\mathbf{C_{a,k,v}})$ The X axis and the Y axis intersect at the origin O, the unit distance $a \in \mathbb{R}^+$, and the point $(\cdots, -4a, -3a, -2a - a, 0, a, 2a, 3a, 4a, \cdots)$ is defined on the X axis and the Y axis respectively. Define $(b, b + ka)$ on the coordinate axis as an interval, where k indicates that it spans several unit distances in an interval, so ka is the width of an interval, and $b = ka \times i, (i \in \mathbb{Z})$ indicates the starting point of the interval. The number of intervals on a coordinate axis is denoted by v.

We show a rectangular coordinate system $C_{a,k,v}$ in Figure 5.

Next, define the representation of an area on the rectangular coordinate system. We set point $(m, n), m = ka \times i + \frac{ka}{2}, n = ka \times j + \frac{ka}{2}, where i, j \in \mathbb{Z}$. The square surrounded by the four points $\left(m - \frac{ka}{2}, n - \frac{ka}{2}\right)$, $\left(m + \frac{ka}{2}, n - \frac{ka}{2}\right)$, $\left(m + \frac{ka}{2}, n + \frac{ka}{2}\right)$, and $\left(m - \frac{ka}{2}, n + \frac{ka}{2}\right)$ is called an area $I_{m,n}$, and an area is defined by the center coordinates of the square forming the area $I_{m,n} = (m, n)$. For example, in Figure 6, $k = 2$, points $(2a, 2a)$, $(4a, 2a)$, $(2a, 4a)$ and $(4a, 4a)$ form an area $I_{3a,3a} = (3a, 3a)$.

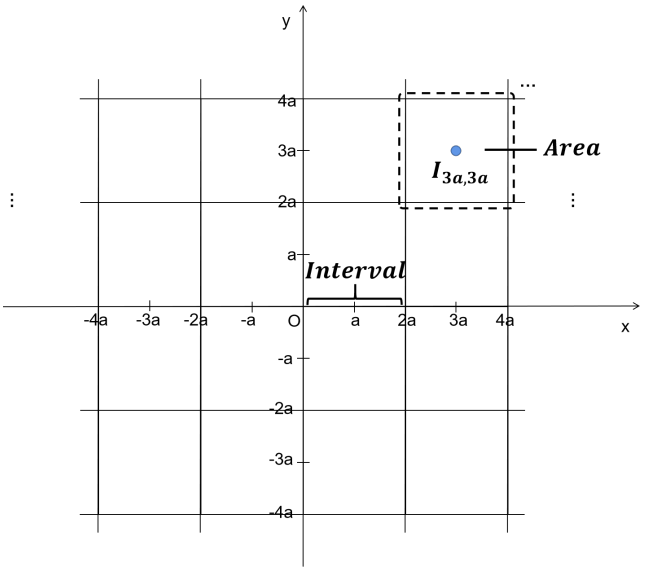


Fig. 5: The rectangular coordinate system constructed by the parameters a, k, v, the figure $[0, 2a]$ is an interval. $I_{3a,3a} = (3a, 3a)$ identifies an area, represented by the coordinates $(3a, 3a)$ of the center point of the area.

2) Secure Sketch Registration Stage: The security outline includes two algorithms: SS and Rec. First we will show the process of system initialization, and then introduce the proposed secure sketch.

Init : The coordinate system $C_{a,k,v}$ is constructed based on Definition 3. There are v intervals on each coordinate axis, thus constituting $v \times v$ areas. Within each area, the maximum acceptable Chebyshev distance is $t = \frac{ka}{2}$, we call t also the biometrics threshold.

In the minutiae set $m = (m_1, \cdots, m_n)$, for each minutiae $m_i = (\rho_i, \theta_i)$, the polar coordinates are converted to rectangular coordinates, and the process is as follows:

$$b_i = (x_i, y_i), \text{where} \begin{cases} x_i = \rho \cos \theta_i \\ y_i = \rho \sin \theta_i \end{cases} . \tag{5}$$

Then we get the biometrics information vector $\mathbf{B} = (b_1, \cdots, b_n)$.

$\mathbf{SS}(\mathbf{B}) \rightarrow \mathbf{s}$: For the user's biometrics information vector $\mathbf{B} = (b_1, \cdots, b_n)$, $b_i = (x_i, y_i)$ is a point in the rectangular coordinate system $C_{a,k,v}$. First, we introduce how to find the area $I_{m,n} = (m, n)$ of a point $x = (p, q)$ in the given coordinate system $C_{a,k,v}$ in Equation 6. According to Definition 3, the width of the interval on each coordinate axis is ka.

$$\begin{aligned} p > 0, m &= \lfloor \frac{p}{ka} \rfloor \times ka + \frac{ka}{2} \\ p \leq 0, m &= \lceil \frac{p}{ka} \rceil \times ka - \frac{ka}{2} \\ q > 0, n &= \lfloor \frac{q}{ka} \rfloor \times ka + \frac{ka}{2} \\ q \leq 0, n &= \lceil \frac{q}{ka} \rceil \times ka - \frac{ka}{2} \end{aligned} \tag{6}$$

This method is also applicable to points on the boundary of an area or the intersection of four areas, for example, $b_i = (ka, ka)$. If b_i is on the boundary perpendicular to the $x-axis$, then b_i is attributed to the left area closest to it; if it is on the boundary parallel to the $x - axis$, then b_i is attributed to the area below; if b_i is located at the intersection of the four areas, the judgment is made according to the above two rules.

For all $b_i \in (b_1, \cdots, b_n)$, according to the area $I_{m,n}$ where each b_i is located, we have

$$s_i = I_{m,n} - b_i. \tag{7}$$

Here s_i is a vector represented by a pair of coordinate points in the coordinate system $C_{a,k,v}$, which means that if the point b_i reaches the center (m, n) of the area $I_{m,n}$, it needs to move along the vector s_i. All of the s_i's and the establishment parameters a, k, v of the rectangular coordinate system constitute the sketch $s = (a, k, v, s_1, \cdots, s_n)$, which can be publicly displayed.

3) Secure Sketch Verification Stage: The same feature extraction method is used for the input fingerprint image, and the biometrics information vector $\mathbf{B}'$ is extracted.

$\mathbf{Rec}(\mathbf{B}', \mathbf{s}) \rightarrow \mathbf{B}$: The input is the sketch s and the encoded user's biological information vector $\mathbf{B}' = (b'_1, \cdots, b'_n)$, where $b'_i = (x, y)$ is the point in the coordinate system $C_{a,k,v}$. The recovery algorithm is shown below.

First reconstruct the rectangular coordinate system $C_{a,k,v}$ through parameter a, k, v.

For all $b'_i \in (b'_1, \cdots, b'_n)$ and $s'_i \in (s'_1, \cdots, s'_n)$, calculate:

$$b''_i = b'_i + s_i. \tag{8}$$

For all $b''_i \in (b''_1, \cdots, b''_n)$, find the area $I_{m,n}$ of each b''_i according to Equation 8. Then we calculate:

$$\widetilde{b}_i = I_{m,n} - s_i. \tag{9}$$

Return the vector $\mathbf{D} = \left(\widetilde{b}_1, \cdots, \widetilde{b}_n\right)$, and the Rec algorithm is done. If $dis(\mathbf{B}, \mathbf{B}') \leq t$, we believe that $\mathbf{D} = \mathbf{B}$, which means that the biometrics collected during the registration phase can be fully recovered. The proof will be given below.

***Theorem** 1:* (Correctness of the secure sketch) The correctness of the proposed secure sketch scheme is: if $SS(\mathbf{B}) \rightarrow s$ and $dis(\mathbf{B}, \mathbf{B}') \leq t$, then $\mathbf{B}$ can be recovered from $\mathbf{B}'$, that is, $Rec(\mathbf{B}', s) \rightarrow \mathbf{B}$.

Proof: Given two vectors $\mathbf{B} = (b_1, \cdots, b_n)$ and $\mathbf{B}' = (b'_1, \cdots, b'_n)$, where b_i and b'_i are points defined in the rectangular coordinate system $C_{a,k,v}$, then $\mathbf{B}$ is the input of the algorithm SS, and $\mathbf{B}'$ is the input of the algorithm Rec. For all $b_i \in \mathbf{B}$, $b'_i \in \mathbf{B}'$, given the premise $dis(b_i, b'_i) \leq t$, we have the following derivation process. ■

$$\begin{aligned} &(a) : I_{m,n} = b_i + s_i \\ &(b) : b''_i = b'_i + s_i \\ &(a) - (b) : I_{m,n} - b''_i = b_i - b'_i \\ &\Rightarrow dis(I_{m,n}, b''_i) = dis(b_i, b'_i) \leq t \end{aligned} \tag{10}$$

We know that $I_{m,n}$ is the area where each original minutiae is located. According to Equation 8, the calculated point b''_i is also located in the area $I_{m,n}$, so we have found the area where each original minutia is located in the registration phase. According to the Rec algorithm, the calculation method of the output recovered minutia $\widetilde{b}_i$ is as follows:

$$\widetilde{b}_i = I_{m,n} - s_i \tag{11}$$

According to Equation 7, we have $\widetilde{b}_i = b_i$. Therefore, the original biological feature vector $\mathbf{B}$ can be recovered by the Rec algorithm. The process of recovring the original minutiae b_i is in Figure 6.

Suppose $dis(\mathbf{B}, \mathbf{B}') > t$, then there is at least a pair of points (b_i, b'_i), which satisfies the following inequality :

$$dis(b_i + s_i, b'_i + s_i) > t \tag{12}$$

It can be seen that $b'_i + s_i = b''_i$, $b_i + s_i = I_{m,n}$, we have $dis(I_{m,n}, b''_i) > t$. Thus the calculated point b''_i does not belong to the area $I_{m,n}$ corresponding to the original minutiae. We assume that the area that b''_i belongs to is $I'_{m,n}$, so $\widetilde{b}_i = I'_{m,n} - s_i \neq b_i$, which means the original biometric information cannot be recovered by the Rec algorithm, because the distance between $\mathbf{B}$ and $\mathbf{B}'$ is greater than the threshold t.

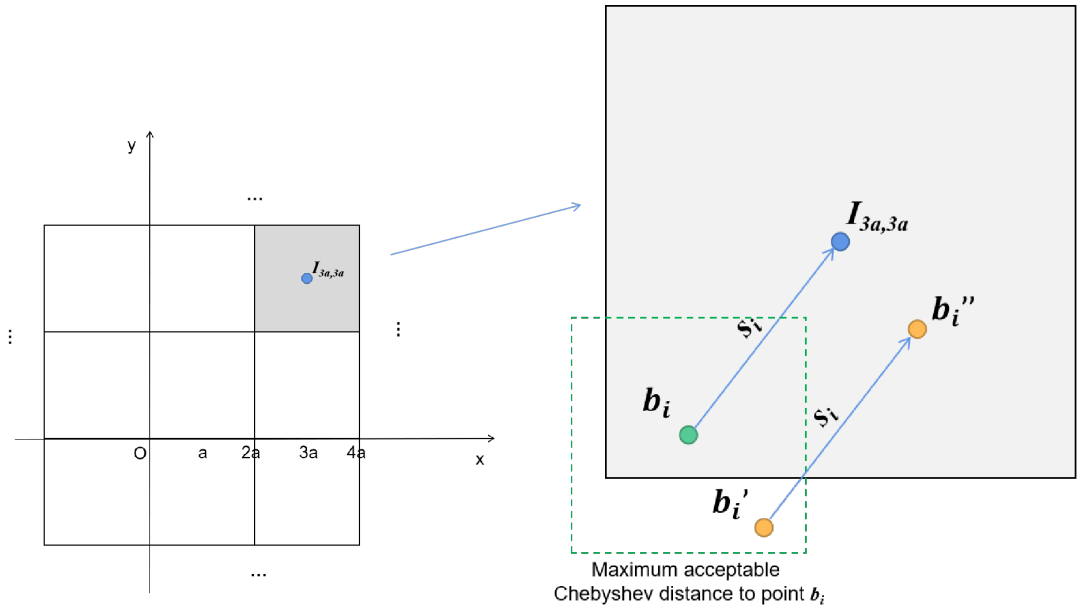


Fig. 6: The distance between b_i' and b_i does not exceed the threshold t, meaning $b_i' + s_i = b_i''$ also belongs to the area $I_{m,n}$ of b_i, and the original minutiae b_i can be recovered by $\widetilde{b}_i = I_{3a,3a} - s_i$

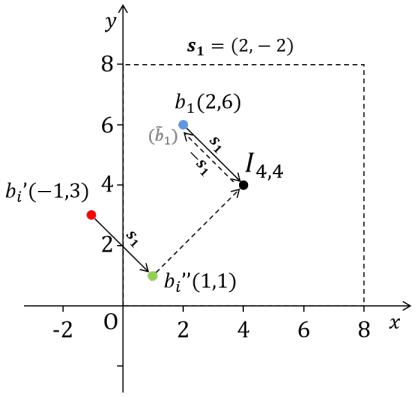


Fig. 7: An example of our secure sketch

4) An Example of Our Secure Sketch: In the coordinate system in Figure 7, it can be seen that the interval width $ka = 8$. Therefore, the threshold $t = \frac{8}{2} = 4$. The point $b_1 = (2, 6)$ is the original biometrics (for example, a fingerprint ending or bifurcation). The area where b_1 is located is $I_{4,4} = (4, 4)$. In the SS algorithm, first calculate the sketch $s_1 = I_{4,4} - b_1 = (4-2, 4-6) = (2, -2)$ according to Equation 7 , then save the public parameter s_1, and the original biometric information b_1 can be destroyed.

In the Rec algorithm, input the biometrics $b_1' = (-1, 3)$ sampled again. Although b_1 and b_1' are in different areas, the Chebyshev distance between them is $dis(b_1, b_1') = 3 < t$. So we can compute $b_1'' = b_1' + s_1 = (1, 1)$, and find the area $I_{4,4}$ of b_1 according to Equation 6. Finally, we calculate $\widetilde{b}_1 = I_{4,4} - s_1 = (4 - 2, 4 - (-2)) = (2, 6)$, and obtain $b_1 = \widetilde{b}_1$, which finishes the recovery.

E. Reusable and Robust Fuzzy Extractor Construction

Based on the secure sketch described in Section 3.3, we propose a fuzzy extractor construction scheme.

In most cases, a user will use the same biological information to register at multiple service providers, or hope to obtain multiple independent keys through a single biological feature. Therefore, our scheme is reusable at the same time, that is, the same biometrics of the same user can generate multiple evenly distributed keys. In order to prevent the public helper string P from being maliciously tampered with by the adversary, we use the authenticator v to make our scheme robust.

Although the proof shows that the secure sketch scheme introduced in section 3.3 can completely recover the minutiae of the original fingerprint image from the fingerprint image collected the second time in the case that the maximum Chebyshev distance between the corresponding minutiae of two fingerprint images does not exceed the threshold t.

However, considering that in practical applications, it is not always possible to find the corresponding minutiae between the two sampled images of the same fingerprint, that is, there may be a small number of minutiae that cannot be detected or false minutiae are detected by mistake, which results in that the above secure sketch algorithm cannot completely recover all the minutiae.

Through our experiments (see Section 5 for details), for the sampling of 15 minutiae in each of the giving 100 fingerprints, based on the above fingerprint feature extraction algorithm and secure sketch algorithm, 14 minutiae in each of 48 figerprints can be recovered. And in the worst case, we found a fingerprint in which only 8 minutiae can be recovered.

The core idea of the fuzzy extractor is to extract the key from the original biological characteristics, with only the public helper string P stored in the database. In order to ensure the uniform distribution of the key, the key should be extracted

from all the original minutiae. At the same time, to ensure that the key can be reproduced, the minutiae that are correctly recovered should be used to reproduce the key. However, it is impossible to know which minutiae can be recovered correctly during the generating and reproducing stages of the fuzzy extractor, because the original biometrics do not need to be saved.

The fuzzy extractor scheme we proposed is described as follows, which can solve the problem above.

1) The Generating Algorithm of Fuzzy Extractor: $\mathbf{Gen}(\mathbf{B}) \rightarrow (\mathbf{R}, \mathbf{P})$: There are a total of n minutiae in one fingerprint, let $m(m \leq n)$ be the worst case where the number of minutiae can be recovered ($m = 8$ in our experiment). First, input all n minutiae into the secure sketch scheme to obtain the Sketch s, which will be used in the subsequent Rep algorithm to try to recover the original minutiae. The next step is key extraction. In order to ensure the randomness of the key, we combine the biometrics $\mathbf{B}$ and the current timestamp T. After connecting $\mathbf{B}$ and T, input them into the $Hash$ function, and divide the obtained hash value into two parts on average. The first part is used as the key R, and the second part is used as the authenticator v.

TABLE IV: The Gen Algorithm of Our Fuzzy Extractors

$Gen(\mathbf{B}) \rightarrow (R, P)$
1. Input $\mathbf{B} = (b_1, \cdots, b_n)$
2. $T \leftarrow timestamp$
3. $(R, v) = H(\mathbf{B} \| T)$
4. Pick m unordered outcomes from n possibilities. $(\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_Z), Z = \binom{n}{m}$.
5. for $i = 1$ to Z $c_i = SKE.Enc(\mathbf{B}_i, (R, v))$
6. $C = (c_1, c_2, \cdots, c_Z), Z = \binom{n}{m}$
7. return $P = (s, v, C)$

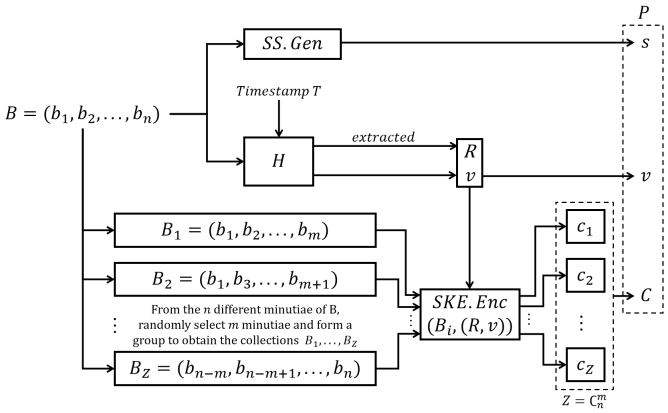


Fig. 8: Gen algorithm

Followed by the construction of public parameters for the key reporoduct of the Rep algorithm. Each time from n different minutiae, randomly select m minutiae and form a group, so as to obtain $\binom{n}{m}$ subsets $(\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_Z), Z = \binom{n}{m}$. $SKE.Enc$ is the encryption algorithm of the symmetric key encryption method. We use (R, v) as the plaintext message and $(\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_Z)$ as the encryption key to encrypt (R, v) to obtain the corresponding ciphertext $c_1, c_2, \cdots, c_Z, Z = \binom{n}{m}$. Let $C = (c_1, c_2, \cdots, c_Z)$. Finally, we save $P = (s, v, C)$ as a public helper string, and the remaining parameters (such as T) and biometrics $\mathbf{B}$ do not need to be saved.

2) The Reproducing Algorithm of Fuzzy Extractor: $\mathbf{Rep}(\mathbf{B}', \mathbf{P}) \rightarrow \mathbf{R}$: Similarly, $\mathbf{B}' = (b'_1, b'_2, , \cdots, b'_n)$ is the fingerprint minutiae of the same finger collected for the second time. First, we need to use the sketch s generated in the previous step, try to recover the original minutiae, and input $\mathbf{B}'$ and s into the recovery algorithm Rec of the secure sketch to get the recovery minutiae's sequence $\widetilde{\mathbf{B}} = (\widetilde{b}_1, \widetilde{b}_2, \cdots, \widetilde{b}_n)$.

TABLE V: The Rep Algorithm of Our Fuzzy Extractors

$Gen(\mathbf{B}', P) \rightarrow (R)$
1. Input $\mathbf{B}' = (b'_1, \cdots, b'_n), P = (s, v, C)$
2. $\widetilde{\mathbf{B}} = Rec(\mathbf{B}', s)$
3. Pick m unordered outcomes from n possibilities. $\widetilde{\mathbf{B}}_1, \widetilde{\mathbf{B}}_2, \cdots, \widetilde{\mathbf{B}}_Z, Z = \binom{n}{m}$.
4. for $i = 1$ to Z for $j = 1$ to Z $(\widetilde{R}_i, \widetilde{v}_i) = SKE.Dec(\widetilde{\mathbf{B}}_j, c_i)$
5. if $\widetilde{v}_i = v$ return $R = \widetilde{R}_i$
6. else return $\perp$

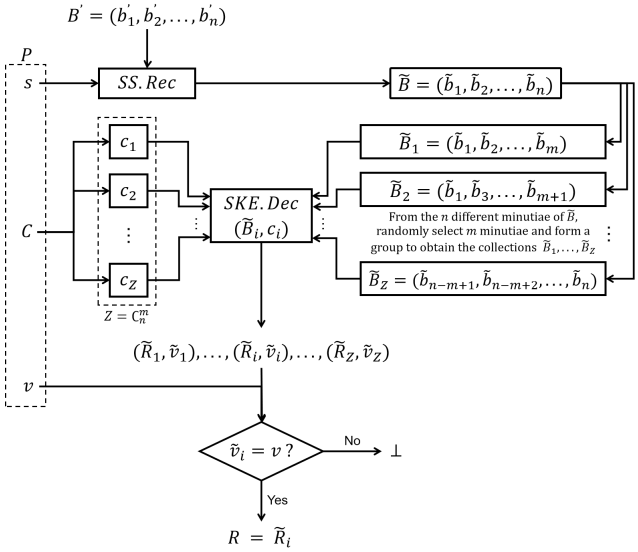


Fig. 9: Rep algorithm

There may be some minutiae $\widetilde{b}_i$ in $\widetilde{\mathbf{B}}$ that are not recovered correctly, so it is also necessary to use the solution of combinations. From all the n recovered minutiae, m minutiae are randomly selected and combined into a group, and we obtain

the subset $\widetilde{\mathbf{B}}_1, \widetilde{\mathbf{B}}_2, \cdots, \widetilde{\mathbf{B}}_Z$, where$Z = \binom{n}{m}$. If there is any $\widetilde{\mathbf{B}}_j = \mathbf{B}$, the decryption of c_i can be realized, which means $SKE.Dec(\widetilde{\mathbf{B}}_j, c_i) \rightarrow (\widetilde{R}_i, \widetilde{v}_i)$. If $\widetilde{v}_i = v$, then $R = \widetilde{R}_i$, which successfully recovered the key R.

3) The Reusability of the Proposed Scheme: Reusability requires that R is pseudo-random for any adversary. For a given biological feature $\mathbf{B}$, the generated n key and help string pairs $\{(R_1, P_1), \cdots, (R_i, P_i), \cdots, (R_n, P_n)\}$. It should be required that even if $\{(R_1, P_1), \cdots, (R_{i-1}, P_{i-1})\}$ and $\{(R_{i+1}, P_{i+1}), \cdots, (R_n, P_n)\}$ are all known by the adversary, R_i should also be secure.

The following gives a proof that the security of key R_i is not affected when any key R_j and the corresponding helper string $P_j = (s_j, v_j, C_j)$ are leaked.

Proof: for $j \neq i$, since $(R_i, v_i) = H(\mathbf{B} \| T_i)$ and $(R_j, v_j) = H(\mathbf{B} \| T_j)$, and T_i and T_j are different timestamps, we know that $T_i \neq T_j, v_i \neq v_j$, which means $R_i \neq R_j$. ■

Because each T is derived from the timestamp when the user registered, and there is no case of registering twice with the same biometrics at the same time. So each T is different for the same biometrics. Therefore, our scheme is reusable. At the same time, if it satisfies the random oracle model, our scheme has strong reusability.

4) The Robustness of the Proposed Scheme: Robustness requires that it is difficult for an adversary to forge a $\widetilde{P}_i \neq P_i$ that is legal for R_i, which means it is impossible to trick the user into generating a false key with a tampered helper string. We will discuss robustness by a game between adversary and user. Suppose that the adversary A tampered with $P_i \xrightarrow{A} \widetilde{P}_i$ in the the $Game$.1.1. At the same time, the adversary does not have the user's real biometrics, and $\mathbf{B}^*$ is the fictitious biometrics of the adversary. We define that A wins the game if user uses his forged key $\widetilde{R}_i$.

$Game$.1.1 Attacker's actions
Input: P_i, $\mathbf{B}^*$
for $P_i = (s_i, v_i, C_i)$
$T \leftarrow timestamp$
$(\widetilde{R}_i, \widetilde{v}_i) = H(\mathbf{B}^* \| T)$
$\widetilde{C}_i = (\widetilde{c}_1, \cdots, \widetilde{c}_Z), \widetilde{c}_j = SKE.Enc(\mathbf{B}_j^*, (\widetilde{R}_i, \widetilde{v}_i))$
return $\widetilde{P}_i = (s_i, \widetilde{v}_i, \widetilde{C}_i)$

Fig. 10: $Game$.1.1 Attacker A forges the key and tampered with the public helper string P

The forged $\widetilde{P}_i$ is then sent to the user in an attempt to induce the user to generate the wrong key $\widetilde{R}_i$. The user's judgment process is as the following game:

$Game$.1.2 User's actions
Input: $\mathbf{B}', \widetilde{P}_i$
$\widetilde{\mathbf{B}} = Rec(\mathbf{B}', s)$
Pick m unordered outcomes from n possibilities.
for $i = 1$ to Z
for $j = 1$ to Z
$\widetilde{\mathbf{B}}_j \neq \mathbf{B}_i^*$(since $\widetilde{\mathbf{B}} \neq \mathbf{B}^*$)
$\perp \leftarrow SKE.Dec(\widetilde{\mathbf{B}}_j, \widetilde{c}_i)$
return $\perp$

Fig. 11: $Game$.1.2 Authentication on the user side will fail.

The biometrics input at this time is $\mathbf{B}'$. Although the Sketch s has not been tampered with, $\widetilde{\mathbf{B}}$ can be recovered by the Rec algorithm of the secure sketch. But for the adversary's forged $\widetilde{C}$ by $\mathbf{B}^*$, the $SKE.Enc$ algorithm uses a subset of $\mathbf{B}^*$ as the encryption key. In the reproduction phase, the user uses a subset of $\widetilde{\mathbf{B}}$ as the decryption key for $SKE.Dec$. Because the user's biometrics will not be stored in any database, the adversary does not have any information about user's biometrics, so $\mathbf{B}^* \neq \widetilde{\mathbf{B}}$, so the decryption algorithm will output an error. Finally, the fuzzy extractor outputs $\perp$ in the reproduction stage. So it is impossible for an adversary to win the $Game$.1, which means that our solution is robust.

IV. Security Analysis of Our Scheme

The proposed secure sketch is used as a main block of our fuzzy extractors. To show the security level of secure sketch scheme, we consider an adversary who aims to recover the original biometrics from a given Sketch s. That is, we assume that an adversary can obtain and manipulate Sketches in some manners. If there is too much information about the original input in s, then this information may be obtained by the adversary to recovery the original biological characteristics. Formally, the advantage of this adversary is captured by using information entropy.

***Theorem** 2:* The proposed algorithm (SS, Rec) is an $(C_{a,k,v}, 2n \log akv, 2n \log v, t)$-secure sketch, where v is the number of intervals on each axis. Both the sketch generating SS and the recovery algorithm Rec run in polynomial time of n, k, a and v.

Proof: Since s is a public parameter, if an adversary knows the area $I_{m,n}$ to which each s_i belongs, then he can recover the user's biometrics $\mathbf{B}$. The rectangular coordinate system defined by us is determined by the parameters (a, k, v), in which there are a total of $v \times v$ areas, and the area of each area is $ka \times ka$, so the total area is $(akv)^2$. We know that in the biological feature $\mathbf{B} = (b_1, \cdots, b_n)$, each b_i represents the coordinate of the minutiae in the fingerprint image, so each minutia is an integer. Therefore, when analyzing security, we only need to consider the integer points in the coordinate system. In the defined coordinate system $C_{a,k,v}$, including the point on the boundary, there are a total of $(akv + 1)^2$ integer points. ■

Randomly select a point x in the coordinate system $C_{a,k,v}$, the probability that the point is exactly b_i is:

$$\Pr[x = b_i] = \frac{1}{(akv+1)^2} \tag{13}$$

Then if n points are selected to form a set $\mathbf{X} = (x_1, \cdots, x_n)$, the probability that $\mathbf{X}$ is equal to the original biological feature $\mathbf{B}$ is:

$$\Pr[\mathbf{X} = \mathbf{B}] = \frac{1}{(akv+1)^{2n}} \tag{14}$$

Therefore, the minimum entropy m of input $\mathbf{B}$ is calculated as follows:

$$\begin{aligned} m &= \mathbf{H}_\infty(\mathbf{B}) = -\log \frac{1}{(akv+1)^{2n}} \\ m &= 2n \log (akv+1) \end{aligned} \tag{15}$$

We define that when the Sketch s is given, the probability that the attacker can successfully guess the input biometrics $\mathbf{B}$ is:

$$\mathrm{P}_1 = \Pr[\mathbf{B} = b_i | s = s_i] \tag{16}$$

At the same time, when the biometrics b_i are known, the Sketch s_i is uniquely determined by Equation 7, so P_1 is defined as follows:

$$\begin{aligned} \mathrm{P}_2 &= \Pr[s = s_i | \mathbf{B} = b_i] \\ \mathrm{P}_2 &= 1 \end{aligned} \tag{17}$$

After that, we discuss the security of the secure sketch by calculating the average min-entropy $\widetilde{\mathbf{H}}_\infty(\mathbf{B}|s)$.

$$\begin{aligned} &\max_{x_i} \Pr[\mathbf{B} = b_i | s = s_i] \\ &= \frac{\Pr[s = s_i | \mathbf{B} = b_i] \Pr[\mathbf{B} = b_i]}{\sum_{j=1}^{n} \Pr[s = s_j | \mathbf{B} = b_j] \Pr[\mathbf{B} = b_i]} \\ &= \frac{1 \times \frac{1}{(akv+1)^n}}{\frac{1}{(akv+1)^n}(\Pr[s = s_i | \mathbf{B} = b_1] + \cdots + \Pr[s = s_i | \mathbf{B} = b_n])} \\ &= \frac{1}{\underbrace{1 + \cdots + 1}_{v^2 \text{ in total}}} \\ &= \frac{1}{v^2} \end{aligned} \tag{18}$$

Based on the above calculations, we get the maximum probability of guessing b_i from s_i. Because each element b_i in the input biological feature $\mathbf{B} = (b_1, \cdots, b_n)$ is independent of each other, the average min-entropy of the proposed secure sketch is:

$$\begin{aligned} &\widetilde{\mathbf{H}}_\infty(\mathbf{B}|s) \\ &= -\log(E_{s_i \leftarrow s}[\max_{b_i} \Pr[\mathbf{B} = b_i | s = s_i]]) \\ &= -\log(\frac{1}{v^2})^n \\ &= 2n \log v \end{aligned} \tag{19}$$

The entropy loss of our secure sketch is:

$$\begin{aligned} &m - \widetilde{\mathbf{H}}_\infty(\mathbf{B}|s) \\ &= 2n \log (akv+1) - 2n \log v \\ &= 2n \log \frac{akv+1}{v} \end{aligned} \tag{20}$$

According to the analysis, the security of our secure sketch is related to the number of intervals v of the coordinate axis and the number of sampling points n. The security can be improved by increasing v and n.

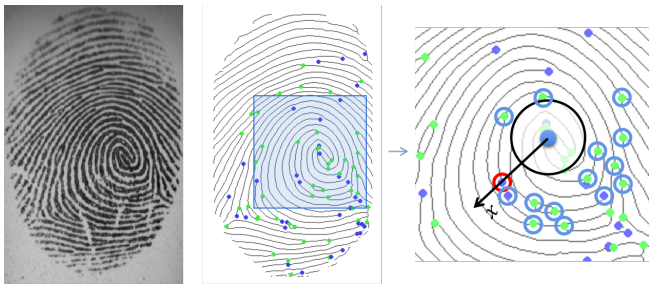
V. Experiment

We use FVC2002 DB1 SetA to evaluate the effect of our fuzzy extractor. The database includes 100 fingers, and each finger has 8 different live scan fingerprint images. The image size is 296×560. For each finger, we choose image 1 as the fingerprint image in the registration process, and choose image 2 as the fingerprint image in the test process, so that a total of 100 sets of experiments will be performed. The experimental environment is as follows: computer system Windows 10 (CPU:Intel i5-8265U; Memory: 16GB; Disk: 512GB). First we use two fingerprint sample images to fully demonstrate the process of our proposed fuzzy extractor, and then show the detailed data of our experiment.

A. Experimental process display

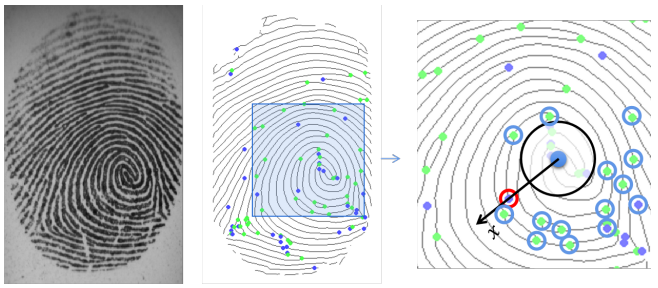
The hash function used in the experiment is SHA-256, and the length of the output hash value is 256 bits. The symmetric key encryption scheme used is AES(Advanced Encryption Standard).

In this experiment, we selected the number of feature points $N = 15$, and the sampling radius $R_{min} = 40px$ (px represents picture pixels). Figure 13 is the fingerprint image (referred to as the registration image for short) used in the fuzzy extractor Gen algorithm, and Figure 14 is the fingerprint image (referred to as the verification image for short) used in the Rep algorithm. The blue point in the figure is the ending of the fingerprint ridge, and the green one is the bifurcation point. The black circle is the sampling radius. The points circled by circles represent the minutiae we finally choose, and the red circles represent the fingerprint ending closest to the center point outside the sampling radius.



(a) (b) (c)

Fig. 12: (a) The fingerprint sampled image used in the fuzzy extractor Gen algorithm. (b) Fingerprint skeleton image. (c) Selected minutiae, sampling radius and established polar coordinate system.



(a) (b) (c)

Fig. 13: (a) The fingerprint sampled image used in the fuzzy extractor Rep algorithm. (b) Fingerprint skeleton image. (c) Selected minutiae, sampling radius and established polar coordinate system.

$\mathbf{Gen}(\mathbf{B}) \rightarrow (\mathbf{R}, \mathbf{P})$:

1) Step 1: Extract the minutiae $\mathbf{B} = (b_1, \cdots, b_{15})$ of the registered fingerprint image and convert the polar coordinates to rectangular coordinates. Select the parameters $a = 10, k = 2, v = 8$, and establish a rectangular coordinate system. The coordinates of the minutiae b_{1-15} are shown in the Table VI.

2) Step 2: Select the timestamp T when the user registered, connect the biometrics, and generate the key R and the authenticator v.

$T = 1622521128.$

$H(\mathbf{B}\|T) = 46408bc21d8ffbcbd1dd2248d56600b0160f13b30b7d0daa561742270f2ac208.$

$R = 46408bc21d8ffbcbd1dd2248d56600b0.$

$v = 160f13b30b7d0daa561742270f2ac208.$

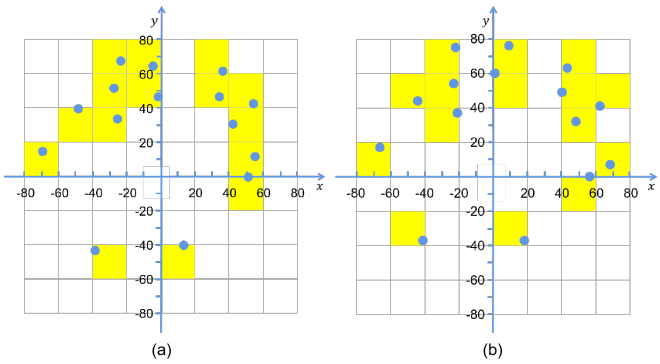
3) Step 3: According to Equation 6, the area of each minutiae is determined. The area $I_{m,n}$ of the minutiae is shown in Table VI. These minutiae and corresponding areas are also shown in Figure 11. According to Equation 7, the calculation sketch $s = (a, k, v, s_1, \cdots, s_{15})$, where $a = 10, k = 2, v = 8$. The values of s_{1-15} are shown in Table VI.

4) Step 4: The number of feature points in $\mathbf{B}$ is $n = 15$, and we choose $m = 10$ minutiae from $\mathbf{B}$ to form a group, so there are a total of $\binom{15}{10} = 3003$ kinds of results, namely $\mathbf{B}_1, \cdots, \mathbf{B}_{3003}$. We use the hash value (R, v) in step 2 as the encrypted content, and use $\mathbf{B}_1, \cdots, \mathbf{B}_{3003}$ as the key to perform the encryption operation to get the ciphertext $c_1, \cdots, c_{3003}$, let $C = (c_1, \cdots, c_{3003})$. Finally we save the public helper string $P = (s, v, C)$.

$\mathbf{Rep}(\mathbf{B}', \mathbf{P}) \rightarrow \mathbf{R}$:

TABLE VI: Parameters generated during the Gen algorithm

Minutiae b_{1-15} coordinate of register fingerprint image				
b_1 $(51, 0)$	b_2 $(55, 12)$	b_3 $(42, 31)$	b_4 $(54, 43)$	b_5 $(34, 47)$
b_6 $(36, 62)$	b_7 $(-2, 47)$	b_8 $(-5, 65)$	b_9 $(-24, 68)$	b_{10} $(-28, 52)$
b_{11} $(-26, 34)$	b_{12} $(-49, 40)$	b_{13} $(-70, 15)$	b_{14} $(-39, -43)$	b_{15} $(13, -40)$
The corresponding area of each minutiae				
$I_{50,-10}$ $(50, -10)$	$I_{50,10}$ $(50, 10)$	$I_{50,30}$ $(50, 30)$	$I_{50,50}$ $(50, 50)$	$I_{30,50}$ $(30, 50)$
$I_{30.70}$ $(30, 70)$	$I_{-10,50}$ $(-10, 50)$	$I_{-10,70}$ $(-10, 70)$	$I_{-30,70}$ $(-30, 70)$	$I_{-30,50}$ $(-30, 50)$
$I_{30,30}$ $(-30, 30)$	$I_{-50,30}$ $(-50, 30)$	$I_{-70,10}$ $(-70, 10)$	$I_{-30,-50}$ $(-30, -50)$	$I_{10,-50}$ $(10, -50)$
Coordinates of vector s_{1-15}				
s_1 $(-1, -10)$	s_2 $(-5, -2)$	s_3 $(8, -1)$	s_4 $(-4, 7)$	s_5 $(-4, 3)$
s_6 $(-6, 8)$	s_7 $(-8, 3)$	s_8 $(-5, 5)$	s_9 $(-6, 2)$	s_{10} $(-2, -2)$
s_{11} $(-4, -4)$	s_{12} $(-1, -10)$	s_{13} $(0, -5)$	s_{14} $(9, -7)$	s_{15} $(-3, -10)$



(a) (b)

Fig. 14: (a)The minutiae of original fingerprint and their area. (b)The minutiae of verification fingerprint minutiae and their area

5) Step 1: Extract the minutiae $\mathbf{B}' = (b'_1, \cdots, b'_{15})$ of the verification fingerprint image and convert the polar coordinates to rectangular coordinates. According the public parameters $a = 10, k = 2, v = 8$, and establish a same rectangular coordinate system. The coordinates of the minutiae b'_{1-15} are shown in the Table VII. Figure 15(b) shows these minutiae and their area.

6) Step 2: Step 2 According to Equation 8, calculate b''_i. The coordinates of b''_i are shown in the Table VII.

7) Step 3: Calculate the recovered minutiae coordinate $\widetilde{b}_i$ according to Equation 9 and the area where b''_i is located. The $\widetilde{b}_i$ coordinates are shown in the Table VII.

8) Step 4: Let $\widetilde{\mathbf{B}} = (\widetilde{b}_1, \cdots, \widetilde{b}_{15})$, and the number of minutiae in $\widetilde{\mathbf{B}}$ is 15. Now also choose $m = 10$ minutiae from $\widetilde{\mathbf{B}}$ to form a group, so there are a total of $\binom{15}{10}$ kinds of results, namely $\widetilde{\mathbf{B}}_1, \cdots, \widetilde{\mathbf{B}}_{3003}$. There are several encryption results stored in $C = (c_1, \cdots, c_{3003})$ in the public helper string saved in the grberation phase, then we can use $\widetilde{\mathbf{B}}_1, \cdots, \widetilde{\mathbf{B}}_{3003}$ to decrypt $C = (c_1, \cdots, c_{3003})$. Because as long as the number

TABLE VII: Parameters generated during the Rep algorithm

Minutiae b'_{1-15} coordinate of verification fingerprint image				
b'_1 $(56, 0)$	b'_2 $(68, 7)$	b'_3 $(48, 32)$	b'_4 $(62, 41)$	b'_5 $(40, 49)$
b'_6 $(43, 63)$	b'_7 $(9, 76)$	b'_8 $(1, 60)$	b'_9 $(-22, 75)$	$b_{10'}$ $(-23, 54)$
b'_{11} $(-21, 37)$	b'_{12} $(-44, 44)$	b'_{13} $(-66, 17)$	b'_{14} $(-41, -37)$	b'_{15} $(18, -37)$
The coordinates of b''_i				
b''_1 $(55, -10)$	b''_2 $(63, 5)$	b''_3 $(56, 31)$	b''_4 $(58, 48)$	b''_5 $(36, 52)$
b''_6 $(37, 71)$	b''_7 $(1, 79)$	b''_8 $(-4, 65)$	b''_9 $(-28, 77)$	b''_{10} $(-25, 52)$
b''_{11} $(-25, 33)$	b''_{12} $(-45, 34)$	b''_{13} $(-66, 12)$	b''_{14} $(-32, -44)$	b''_{15} $(15, -47)$
Coordinates of recovered minutiae $\widetilde{b}_{1-15}$				
$\widetilde{b}_1$ $(51, 0)$	$\widetilde{b}_2$ $(75, 12)$	$\widetilde{b}_3$ $(42, 31)$	$\widetilde{b}_4$ $(54, 43)$	$\widetilde{b}_5$ $(34, 47)$
$\widetilde{b}_6$ $(36, 62)$	$\widetilde{b}_7$ $(18, 67)$	$\widetilde{b}_8$ $(-5, 65)$	$\widetilde{b}_9$ $(-24, 68)$	$\widetilde{b}_{10}$ $(-28, 52)$
$\widetilde{b}_{11}$ $(-26, 34)$	$\widetilde{b}_{12}$ $(-49, 40)$	$\widetilde{b}_{13}$ $(-70, 15)$	$\widetilde{b}_{14}$ $(-39, -43)$	$\widetilde{b}_{15}$ $(13, -40)$

of recovered minutiae are greater than m (here $m = 10$, we actually successfully recovered 13 minutiae), there is $\widetilde{\mathbf{B}}_j = \mathbf{B}_i$, and c_i can be decrypted.

9) Step 5: Use the authenticator v stored in the public helper string P to compare whether the decrypted v_i is equal to v. If they are equal, the reproduction process is completed and the key is successfully recovered. Finally get the reproduced key R.

$$R = 46408bc21d8ffbcbd1dd2248d56600b0.$$

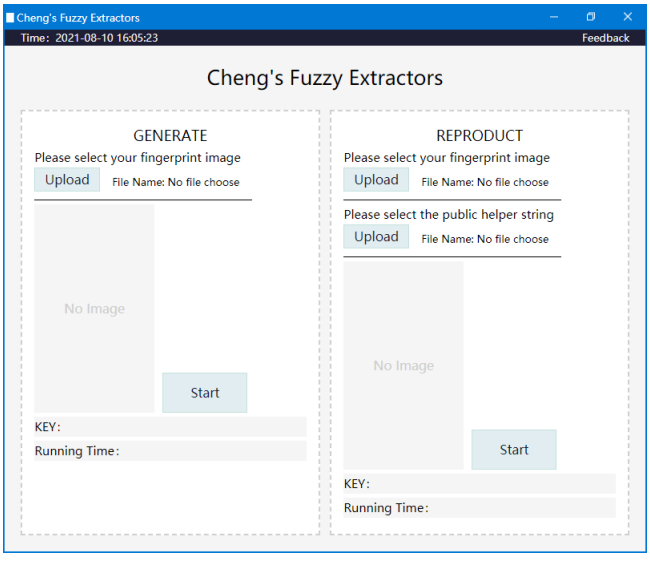


Fig. 15: The screenshot of the program we designed for our proposed scheme

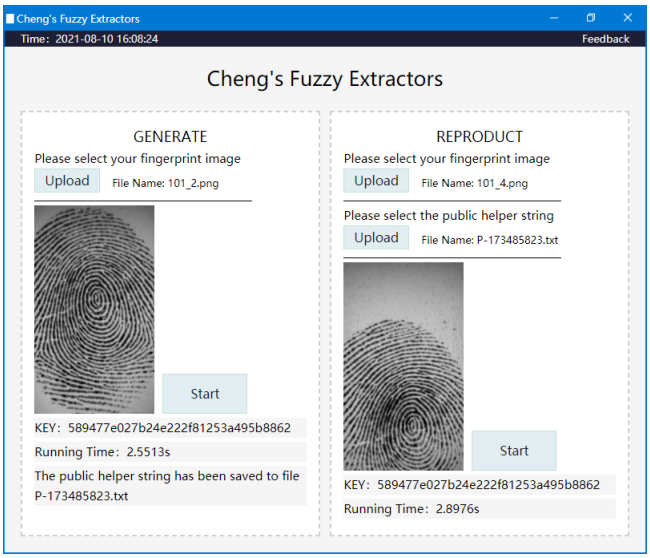


Fig. 16: The screenshot of the program we designed for our proposed scheme

B. The performance of fuzzy extractor

According to the above test procedure, if R is successfully reproduced, it is recorded as a successful experiment, and the output $\perp$ means a failed experiment. We tested the two cases of selected minutiae $N = 15$ and $N = 20$ respectively. The false reject rate(FRR) of the experiment are recorded in Table VIII.

TABLE VIII: The performance of fuzzy extractor

Security	Biometrics				FRR
R(bit)	Number of minutiae (n)	Verification success threshold (m)	Number of samples	Not Passed sample	%
128	15	14	100	52	52. 0%
128	15	13	100	34	34. 0%
128	15	12	100	17	17. 0%
128	15	11	100	10	10. 0%
128	15	10	100	6	6. 0%
128	15	9	100	3	3. 0%
128	15	8	100	2	2. 0%
128	20	18	100	61	61. 0%
128	20	16	100	47	47. 0%
128	20	15	100	46	46. 0%
128	20	14	100	29	29. 0%
128	20	13	100	17	17. 0%
128	20	12	100	13	13. 0%
128	20	11	100	5	5. 0%
128	20	10	100	4	4. 0%

As we can see in the table, when the number of sampling minutiae is $n = 15$ and the number of recovered minutiae is $m = 90$, FRR is equal to 3%. This is a balance between security and efficiency. Too few recovered minutiae will affect the safety of the system. However, if this standard is raised, it will also lead to an increase in FRR.

Next, we analyze the storage space overhead of our schema. The length of the biological feature is calculated as follows: the each minutiae of fingerprint are represented by about 8

bytes, so the length(bytes) of each biological feature is $L = N \times 8 \times 8(bit)$.

For comparison, some experimental data of Canetti et al. are as follows. These analyses are from JH Cheon et al [17]. It can be found that our solution has obvious advantages in storage space overhead when it has longer keys and biometric data.

TABLE IX: Key length, biometrics length and storage space in Canetti's experiment

Security K	Error tolerance T	Biometrics Length (bit)	Storage space (byte)
80	0. 20	512	6. 00G
80	0. 25	1024	932G

TABLE X: The length of the key R, the length of the biometrics $\mathbf{B}$, and the storage cost of the public helper string P in our scheme

Key	Biometrics			Storage space (byte)				FRR
R (bit)	n	Length (bit)	m	s	v	C	Total	%
128	15	960	10	120	16	0. 096G	0. 096G	6. 0
128	15	960	8	120	16	0. 205G	0. 206G	2. 0
128	20	1280	14	160	16	1. 240G	1. 240G	5. 0
128	20	1280	10	160	16	5. 912G	5. 912G	2. 0

VI. Conclusion

The fuzzy extractor effectively protects the security of biometrics and keys. Here, we successfully combined the fuzzy extractor with biological features. In our scheme, taking fingerprints as an example, a complete scheme is proposed from feature extraction and coding to the construction of secure sketche and reusable robust fuzzy extractors. We also analyzed its security theoretically. Finally, a simulation implementation was made, and our solution has relatively lower storage overhead.

References

[1] Koptyra, K., Ogiela, M.R. Multiply information coding and hiding using fuzzy vault. Soft Comput 23, 4357–4366 (2019).

[2] Javier Galbally et al. "Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms". Computer Vision and Image Understanding 117.10(2013): 1512-1525.

[3] Wen Y., Liu S. (2018) Robustly Reusable Fuzzy Extractor from Standard Assumptions. In: Peyrin T., Galbraith S. (eds) Advances in Cryptology – ASIACRYPT 2018. ASIACRYPT 2018. Lecture Notes in Computer Science, vol 11274. Springer, Cham.

[4] Wu, L. , et al. "A Fuzzy Vault Scheme for Ordered Biometrics." Journal of Communications 6.9(2011):682-690.

[5] Pussewalage, Hsg , J. Hu , and J. Pieprzyk . "A survey: Error control methods used in bio-cryptography." International Conference on Fuzzy Systems & Knowledge Discovery IEEE, 2014.

[6] Dodis Y., Reyzin L., Smith A. (2004) Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In: Cachin C., Camenisch J.L. (eds) Advances in Cryptology - EUROCRYPT 2004. EUROCRYPT 2004. Lecture Notes in Computer Science, vol 3027. Springer, Berlin, Heidelberg.

[7] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. D. Smith. Reusable fuzzy extractors for low-entropy distributions. In M. Fischlin and J. Coron, editors, EUROCRYPT 2016, volume 9665 of LNCS, pages 117–146. Springer, Heidelberg, 2016.

[8] Yang, W. , J. Hu , and W. Song . "A Delaunay Triangle-Based Fuzzy Extractor for Fingerprint Authentication." IEEE International Conference on Trust IEEE, 2012.

[9] aur, T. , and M. Kaur . "Cryptographic key generating from multimodal template using fuzzy extractor." 2017 Tenth International Conference on Contemporary Computing (IC3) IEEE, 2018.

[10] Nan, L. , et al. "Fuzzy Extractors for Biometric Identification." 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) IEEE, 2017.

[11] Benjamin Fuller and Xianrui Meng and Leonid Reyzin. "Computational fuzzy extractors". Information and Computation 275.prepublish(2020)

[12] Boyen, X. , et al. "Secure Remote Authentication Using Biometric Data." Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings DBLP, 2005.

[13] Xavier Boyen. "Reusable cryptographic fuzzy extractors". Computer and communications security. pp. 82-91.

[14] Singh, S. D. , and S. P. Majhi . "Fingerprint recognition: A study on image enhancement and minutiae extraction." iosrjournals org (2010).

[15] Jin, B. , H. P. Tang , and M. L. Xu . "Fingerprint singular point detection algorithm by Poincaré Index." WSEAS Transactions on Systems 7.12(2008):1453-1462.

[16] Kawagoe Masahiro and Tojo Akio. "Fingerprint pattern classification". 17.3(1984): 295-303.

[17] Cheon, J. H. , et al. "A Reusable Fuzzy Extractor withPractical Storage Size: Modifying Canetti et al.'s Construction." Australasian Conference on Information Security & Privacy Springer, Cham, 2018.