Journal of Computers Vol. 31 No. 4, 2020, pp. 126-140
doi:10.3966/199115992020083104010

A Retrieval Algorithm of Encrypted Speech Based on Biological Hashing

Qiu-Yu Zhang*, Gai-Li Li, Si-Bin Qiao

School of Computer and Communication, Lanzhou University of Technology, Gansu, Lanzhou 730050, China
{zhangqylz, ligaili01, qiaosibin}@163.com

Received 13 January 2019; Revised 10 May 2019; Accepted 31 May 2019

Abstract. In order to improve the robustness and discrimination of the speech perceptual hashing function over the content based encrypted speech retrieval, increase the retrieval efficiency and accuracy, and realize the privacy and security of speech data. Biological hash has a good robustness, discrimination, and good protection for biometric template in the aspect of multimedia authentication as a feature extraction method. We propose a retrieval algorithm of encrypted speech based on biological hashing. Firstly, a two-dimensional Henon mapping encrypted algorithm is used to encrypt the speech and upload it to encrypted speech library of the cloud. Secondly, the speech signal is conducted with discrete wavelet transform (DWT) after pre-processing conducts to produce low frequency wavelet coefficients which are regarded as the wavelet Merlin matrix. Finally, the pseudo random matrix is generated by the Logistic mapping, and the pseudo random matrix is conducted with fast Fourier transformation (FFT) transform to produce pseudo random Fourier matrix. The wavelet Merlin matrix and pseudo random matrix are conducted with iterations and thresholds to produce the binary perceptual hashing sequence, and binary perceptual hashing sequence stored in the system hash index table of the cloud. When retrieving data, the Hamming distance is used to match and retrieval. Experimental results show that the hash function of the proposed algorithm has excellent robustness, discrimination and security, as well as the efficiency and accuracy of retrieval model is obviously improved.

Keywords: biological hashing, discrete wavelet transform, encrypted speech retrieval, Henon map, logistic pseudo random matrix

1 Introduction

Facing massive multimedia data, it has been a hot issue in the field of multimedia retrieval research that how to guarantee user's data security and how to retrieve required content accurately and quickly from mass data. Especially before speech is uploaded to cloud, speech data is encrypted to ensure security of the sensitive speech data. However, encrypted data loses inherent characteristics of speech and has a certain impact on retrieval. Therefore, quickly and accurately retrieving data in encrypted speech data will be a problem to be solved [1].

Currently, speech retrieval methods mainly include text retrieval [2], content retrieval [3], and semantic retrieval [4]. In which content-based speech retrieval is a hot topic of research. Its methods mainly include: multiple keyword retrieval [5], and fuzzy retrieval [6] and sorting retrieval [7]. Feature extraction methods mainly include perceptual hashing [8], audio fingerprint [9], biological hashing [10], etc. Based on these retrieval and feature extraction methods, Domestic and foreign scholars have achieved very good research results in the content-based encrypted speech retrieval. For example, Zhao, et al. [11] proposed a speech perceptual hashing algorithm that piecewise aggregate approximation (PAA) was used for compressing data size and multifractal extract perceptual hash to improve retrieval speed,

* Corresponding Author

but it was not good at robustness and discrimination of speech perceptual hashing function. Wang et al. [12] proposed that zero-crossing rate is used as retrieval digests, and perceptual digest was embedded into speech as digital watermark, but robustness and discrimination were poor, besides, there was no better balance between robustness and compactness. Hao et al. [13] proposed a speech perceptual hashing algorithm based on time and frequency domain, short-term energy was used as perceptual feature extraction in time domain, and change characteristic of Bark domain energy was used as perceptual feature extraction in frequency domain, perceptual hashing of time and frequency domain is combined in a cross way to generate final perceptual hash sequence. But robustness and compactness are poor. Jin et al. [14] proposed a resilience mask method for robust speech hashing, hash sequences are generated under spectral spreading and distortion of adding noise, but security is not under consideration. Aljawarneh et al. [15] proposed a multimedia retrieval algorithm under large-scale data, using Feistel encryption standard (FES) to generate keys, advanced encryption standard (AES) encrypting plaintext, and using genetic algorithm (GA) to combine, the algorithm has good security, but robustness and discrimination were poor. He et al. [16] proposed an encrypted speech retrieval algorithm based on syllable-level perceptual hash, and embedded binary perceptual hash sequence into speech as digital watermark, but robustness and security were weak. Sun et al. [17] proposed a speech endpoint detection algorithm based on sub band energy ratio, and improved retrieval efficiency by using Hamming window function and Mel-frequency cepstrum coefficient (MFCC) sub band division method, but security is unsatisfied. Chen et al. [18] proposed an audio fingerprints algorithm based on wavelet transform to improve retrieval efficiency, but it did not consider security. Liu et al. [19] proposed a random projection algorithm, which feature sets were projected onto random subspaces according to randomly generated pseudo matrices, binary quantization after projection prevents recovery of original biometric data from distorted templates, and algorithm has good security and recognition performance. Lacharme et al. [20] proposed a biological template protection algorithm. By reconstructing fingerprints, the original and counterfeit fingerprint can be accurately identified by biological hash algorithm. It is proved that biological hashing has a good protection and recognition performance for biometric templates.

By analyzing above literature, the existing methods do not better balance between robustness and compactness in extracting retrieval digests from speech signals, discrimination and robustness are poor, and retrieval efficiency and retrieval accuracy are relatively low. Although the existing algorithms take security of speech data in cloud into account, it does not consider security of retrieving hash digests. Therefore, the existing content-based encrypted speech retrieval algorithms still have many defects and deficiencies. In addition, biological hash has a good robustness, discrimination, and good protection for biometric template in the aspect of multimedia authentication as a feature extraction method. The biological hash algorithm is mainly applied to the feature extraction of images, in order to take advantage of the biological hash algorithm. It can be used in deep study of encrypted speech retrieval algorithm.

From what has been discussed above, we present a retrieval algorithm of encrypted speech based on biological hashing in this study. This method uses biological hash technique, which combines speech feature vector conducted by speech feature with the pseudo random number generated by Logistic pseudo random matrix to produce binary hash sequence and completes the content-based matching retrieval as retrieval digest. The experimental results show that the proposed algorithm can handle five different speech formats of speech clips in the hash function construction, such as WAV, MP3, FLAC, OGG and M4A. Through analyzing feature extraction and retrieval results of 2,000 speech clips of from five different speech formats, the proposed algorithm has good robustness and discrimination, and good retrieval efficiency and accuracy of retrieval model as well, as well as security also has a good performance.

In this paper, we present a retrieval algorithm of encrypted speech based on biological hashing. The main contributions of our method are summarized as follows:

(1) We propose the features of the biological hashing function, which can have good protection for biometric templates as a feature extraction method, while traditional hashing algorithm lack of security considerations for hashing sequences.

(2) We apply DWT and Logistic algorithm to construct biological hashing. The experimental results show that the proposed algorithm has very good robustness, discrimination and security, as well as the retrieval efficiency and accuracy.

(3) When the generated hashing sequence is destroyed, a new biological hashing sequence can be generated by the new key, and thus has good revocability, while traditional hashing algorithm lose the

revocability.

(4) We evaluate the proposed method on 2,000 databases, and experimental results demonstrate the effectiveness and robustness of the proposed method.

The rest of this paper is organized as follows: Section 2 describes related theory. Section 3 introduces our proposed algorithm. Section 4 gives experimental results and performance analysis as compared with other related methods. Finally, we conclude our paper in Section 5.

2 Related Theory

2.1 Logistic Generated Pseudo Random Matrices

Pseudo random matrix generated by Logical map [21] and feature vectors formed by biometrics are conducted with iterations to produce retrieval digest. Logistic map is a simple one-dimensional mixed model, but it can generate complex trajectories. Its mathematical formula is defined as Eq. (1):

$$x_n = u * x_{n-1} * (1 - x_{n-1}), \tag{1}$$

where, u is branch parameter, ranges from 0 to 4, and the system is in a chaotic state when $3.5699<u<4$, the value of x_n is between 0 and 1. Under the condition that u is between 3.5699 and 4, the sequence (x_n, n=0, 1, 2, 3, …, n) resulting from initial value x_0 is a pseudo random sequence with strong randomness and initial sensitivity.

The pseudo random matrix generation step is as follows:

Step 1: Using the Eq. (1) to iterate, In order to get random numbers with good randomness, taking parameter u=3.9769, initial value x_0=0.2785, x_0 is key that generates random matrix, x_0 is between 0 and 1. The length of the generated pseudo random matrix is 4,000.

Step 2: The x_n obtained by each iteration is conducted to binaryzed, if $x_n>=0.5$, then let x_n=1; otherwise, x_n=0.

Step 3: The pseudo random matrix generated by binary is transformed into $\boldsymbol{G}(1, n)$ and then converted to $\boldsymbol{G}(m, n)$.

2.2 Discrete Wavelet Transform

A wavelet transform is used to construct biometric vector to generate retrieval digest. DWT aims to discretize scale and translation. In the application, it is necessary to discretize scale factor a and displacement factor b, as shown in Eq. (2):

$$a = a_0^m, b = nb_0a_0^m, \tag{2}$$

where, m and n are integers, a_0 is a constant greater than 1, b_0 is a constant greater than 0, the choice of a and b is related to specific form of wavelet.

The DWT function is expressed as follows:

$$\varphi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}}\varphi(\frac{t - nb_0a_0^m}{a_0^m}) = \frac{1}{a_0^m}\varphi(a_0^{-m}t - nb_0). \tag{3}$$

The corresponding DWT is represented as follows:

$$w_f(m,n) =< f, \varphi_{m,n}(t) >= \int_{-\infty}^{+\infty} f(t)\varphi_{m,n}(t)dt. \tag{4}$$

When a_0=2 and b_0=1, DWT is called as binary DWT.

The significance of wavelet decomposition lies in ability to decompose signals at different scales, different scales' choice can be determined according to different goals. Fig. 1 shows how a discrete signal can be transformed by discrete wavelet using a hierarchical architecture.

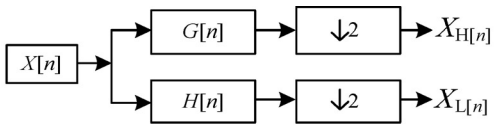


Fig. 1. Discrete signal of DWT schematic

In Fig. 1, $X[n]$ is discrete input signal of length N. $G[n]$ is low-pass filter that filters high-frequency part of input signal and outputs low-frequency part. $H[n]$ is high-pass filter, which filters out low-frequency components and outputs high-frequency components. ↓2 is a down sampling filter. If $X[n]$ is taken as input, then $X_{L[n]}$ or $X_{H[n]}$ is output, where $X_{H[n]}$ is high frequency part of $X[n]$ which obtained speech signal through low-pass filter and down-sampling filter; $X_{L[n]}$ is low-frequency part of speech signal obtained by inputting low-pass filter and down-sampling filter as input, and low-frequency part can be used to construct the wavelet Merlin matrix.

2.3 Henon Mapping

Since speech data is relatively sensitive and widely used, it is more necessary to ensure security if it is uploaded to cloud. Therefore, this paper use two-dimensional (2D) Henon [22] chaotic map encryption algorithm to construct encrypted speech. The encryption algorithm uses combination of XOR operation and modular subtraction. The principle of the algorithm is shown in Eq. (5).

$$\begin{cases} x_{k-1} = 1 - ax_k^2 + y_k \\ y_{k+1} = bx_k \end{cases}, \tag{5}$$

where, a and b have influence on Henon map, when $0.54<a<2$, $0<|b|<1$, the system is in a chaotic state, x and y are two variables to determine the iterative equation.

Construction of 2D Henon map function, take the 2D Henon map as follows:

$$x_{k+1} = 1 - ax_k^2 + 0.3x_{k-1}, \tag{6}$$

where, x determines iteration process, when a increases from 0, Henon map goes to chaos via period-doubling bifurcation. When $0<a<0.35$, Henon map has stable fixed points.

$$1 - ax_k^2 + 0.3x_{k-1} - x_i^2 a_k x = \frac{-0.7 \pm \sqrt{0.49 + 4a}}{2a}. \tag{7}$$

According to the Euclid formula and the 2D Henon map function in the above mathematical knowledge, the following function will be constructed:

$$c_i = \begin{cases} 0 & 0.5 < x_k \le 1.5 \\ 1 & -0.5 < x_k \le 0.5 \\ 2 & -1.5 \le x_k \le 0.5 \end{cases} \quad (i = 0,1\cdots, k = 0,1\cdots), \tag{8}$$

where x_k is in the range of [-0.5, 1.5], x_k will be binarized to 0 and 1 according to the first two functions of c_i, the speech data is encrypted by the XOR. When x_k is in the range [-1.5, 0.5], the proposed algorithm converts x_k to 2 according to third function of c_i, and then speech data is encrypted by subtractive computing. This completes encryption of speech data.

3 The Proposed Algorithm

The processing flow of retrieval algorithm of encrypted speech based on biological hashing is shown in Fig. 2. The proposed algorithm consists of three processes: constructing encrypted speech library, constructing system hash index table, and user speech retrieval.

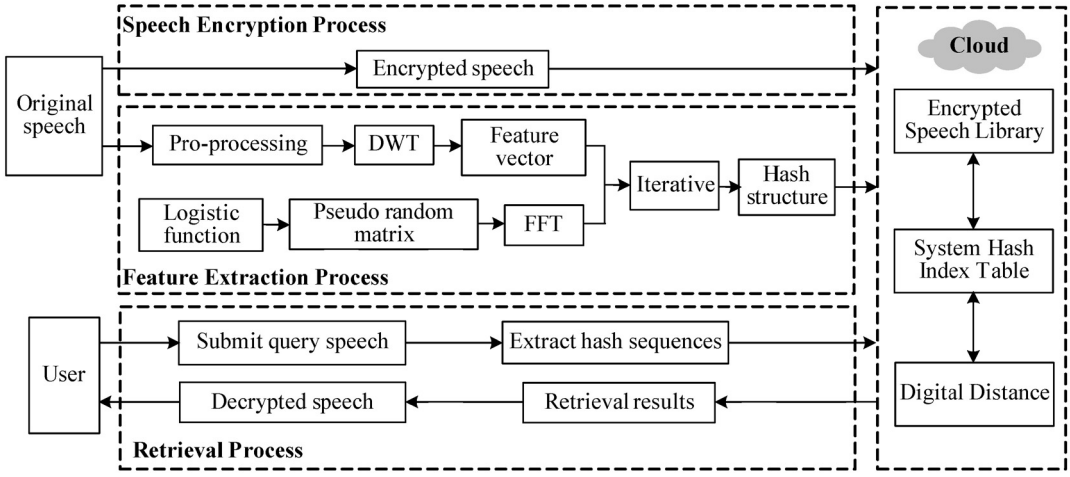


Fig. 2. The flow chart of proposed algorithm

In the process of constructing encrypted speech library, in order to ensure security of speech data, firstly, the speech use 2D Henon chaotic map encryption algorithm to construct encrypted speech library, and then encrypted speech library is to be uploaded and stored in the cloud.

In the process of constructing system hash index table, firstly, the speech signal is pre-processed, speech feature vector is obtained through DWT transformation and pseudo random matrix generated by Logistic function and FFT are conducted with iterations. Secondly, the result of iteration is threshold to generate binary hash sequence which is used to complete process of hash construction. Finally, the generated retrieval digest is stored in the system hash index table of the cloud.

In the retrieval process, firstly, using the construction process of the system hash index table extract the hash retrieval digest of the speech to be retrieved. And then the hash sequence of the speech to be retrieved and the system hash index table of the cloud system are compared through the Hamming distance matching algorithm. When the hash sequence and the hash sequence in the system hash index table match successfully, the encrypted speech corresponding to the hash sequence in the system hash index table is returned for the user, and it is successfully retrieved. Finally, the encrypted speech retrieved is constructed to decrypt to complete the content-based encrypted speech retrieval process.

3.1 Speech Encryption Process

Cloud storage is not a trusted third party. Therefore, if sensitive speech data stored in cloud without protected, it may result to personal privacy or even cause great danger to national security. Therefore, speech data need to be encrypted before uploaded to cloud. With the reason that we use 2D Henon chaotic map encrypted algorithm to encrypt the speech. The flow diagram of encryption algorithm is shown in Fig. 3.

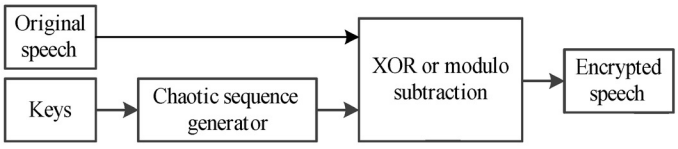


Fig. 3. The flow diagram of encryption algorithm

The specific encryption processes are as follows:

Step 1. Matrix generation. Let the original speech be $s(t)$ and convert it to matrix $\boldsymbol{A}$.

Step 2. Key generation and speech encrypted. Set an initial value x_0, x_1 as the key. Where x_0=1, x_1=1.2, a chaotic sequence $\boldsymbol{C}$ is generated according to the above Eq. (10). Finally, the matrix $\boldsymbol{A}$ generated by the speech and the chaotic sequence $\boldsymbol{C}$ are subjected to XOR operation or modulo subtraction to obtain the encrypted speech $s'(t)$.

This method is used to encrypt the speech for all the original speeches of the user, and the encrypted speech is uploaded to encrypted speech library of the cloud.

3.2 Biological Hashing Feature Extraction Process

The process of using the biological hash function to extract speech perceptual features and constructing a biological hash digest are as follows:

Step 1. Pre-processing. The input speech signal $s(t)$ needs to conduct a pre-weighting process to enhance the part of high frequency. The speech signal processed by pre-weighting is denoted as $s'(t)$.

Step 2. DWT transform. 3-level wavelet decomposition is performed on speech signal $s'(t)$, and low-frequency coefficients are obtained, which are denoted as $L = \{L_i \mid i = 1, 2, \ldots, N\}$, and N is the length of low-frequency coefficients. The wavelet transform will be used to extract the low frequency coefficient $\boldsymbol{A}$, and construct as a matrix $\boldsymbol{X}(n, m)$.

Step 3. Logistic pseudo random transformation. According to the equations $x_n=u\times x_{n-1}\times(1-x_{n-1})$, an initial value x_1 is set as the key, u is a constant; and it iterate over 4,000 thresholds to generate a random matrix $\boldsymbol{B}(n, m)$.

Step 4. FFT transformation. A fast Fourier transform is performed on the random matrix $\boldsymbol{B}$ generated in Step 3 to obtain a matrix $\boldsymbol{Y}(n, m)$.

Step 5. Hash structure. Using the matrix $\boldsymbol{X}$ obtained in Step 2 and the matrix $\boldsymbol{Y}$ generated in Step 4 is carried out iteratively to generate the matrix $\boldsymbol{P}(n, n)$, and then the N-dimensional matrix $\boldsymbol{P}$ produce the matrix $\boldsymbol{H}(1, N)$ and calculate the average value of the matrix $\boldsymbol{H}$ as avg.

Using the vector $\boldsymbol{H}$ is conducted with hash structure to generate a hash sequence $h=\{h(i) \mid i=1, 2, \ldots, M\}$. The binary hash structure method is as follows:

The average value avg of the matrix $\boldsymbol{H}$ is subtracted from all the data of the parameter matrix $\boldsymbol{H}$. If it is greater than 0, the data of the row becomes 1; otherwise it is 0.

$$h(i)=\begin{cases}0 & if \quad H(i)-avg>0\\ 1 & else\end{cases} \quad i=1,2\cdots M\ , \tag{9}$$

where, M is the length of the perceptual hashing value.

Step 6. System hash index table build. The original data is used to obtain the hash sequence $\boldsymbol{h}(1,400)$ through the above-mentioned biological hashing algorithm. In this process, different data have the same subscript, which is a conflict phenomenon. To solve this problem, you can use the first data stored in this subscript as the node of the linked list, and then build a linked list of all the data that obtain the subscript. The specific process is as follows:

Through the hash function, all the speech data U is mapped to the slot of the system hash index table $\boldsymbol{T}[0, 1, ..., m\text{-}1]$:

$$h:U\to\{0,1,2,\cdots,m-1\}\ . \tag{10}$$

According to this method, all speech in the speech library are operated. After traversing all the speeches, the system hash index table is built, and the speech are effectively organized in the form of a system hash index table, which lays the foundation for the next retrieval.

The biological hashing feature extraction process is shown as follows.

Input: the speech signal
Output: hash sequence

```
1. [c, l] = wavedec();                    /* Using 3-level wavelet to decompose data */
2. A = appcoef();                         /* The wavelet extract the low frequency coefficient A,
                                             and construct X(n, m) */
3. x(n) = u × x(n-1) × (1-x(n-1));        * iterating over 4000 thresholds to generate B(n, m) */
4. Z = fft2(Y);                           /* A fast Fourier transform obtain a matrix Y(n, m). */
5. P = X × Z;
6. avg = mean(P);                         /* Hash structure */
```

3.3 Retrieval and Decryption Process

Retrieval process. Retrieval phase is the most important part of the system. In order to prevent data from causing unnecessary losses, the speech data should be encrypted before uploaded it to the cloud. However, in the case of ensuring data security, how to find the necessary speech clips in the encrypted data is the problem to solve. First, it uses a speech clip to extract, match, and then output the result. The specific steps are as follows:

Step 1. Submit speech. When user submits the speech S_1 to be queried, according to feature extraction method mentioned above, the hash sequence h_1 is extracted from the submitted query speech clip.

Step 2. Speech retrieval. Compare hash sequences h_1 with hash sequences $h=\{h(i) \mid i=1, 2, \ldots, M\}$ of the system hash index table constructed by all speech $S=\{S(i)|i=1, 2, \ldots, M\}$ through hamming distance. When Hamming distance between h_1 and the hash sequence $h=\{h(i)|i=1, 2, \ldots, M\}$ of the system hash index table is less than or equal to the threshold value, the retrieval is successful, returns the encrypted speech corresponding to the hash sequence in the system hash index table for user. The specific Hamming distance matching process is as follows: h_1 is recorded as the hash value of query speech signal S_1, and h_2 is recorded as the hash value of any speech signal S_2 in the speech library, hd is denoted as the normalized Hamming distance $\boldsymbol{DH}$ (:,:) of h_1 and h_2, bit error rate (BER) is defined as the ratio of error bit number and the total number of the perceptual hashing value. The BER is defined as follows:

$$DH(S_1, S_2) = hd(h_1, h_2) = \frac{1}{M}\sum_{i=1}^{M}\left(\left|h_1 - h_2(i)\right|\right) = \frac{1}{\mathrm{M}}\sum_{\mathrm{i}=1}^{\mathrm{M}}\left(h_1 \oplus h_2(\mathrm{i})\right), \tag{11}$$

where, $\boldsymbol{DH}$ (S_1, S_2) is the BER of speech signal S_1, S_2, h_1 and h_2 are the hash values generated by speech S_1, S_2 respectively, M is the length of perceptual hashing value.

We use hypothesis test of hashing digital distance $\boldsymbol{DH}$**(:,:)** to describe the hashing matching.

P_0: if the query speech signal S_1 and any speech signal S_2 in the speech library are the same:

$$DH(S_1, S_2) \le \tau, \tag{12}$$

P_1: if the query speech signal S_1 and any speech signal S_2 in the speech library are different:

$$DH(S_1, S_2) > \tau, \tag{13}$$

where, τ represents the perceptual retrieval threshold, by setting the size of matching threshold, calculate mathematical distance between perceptual hashing sequences of the speech clips S_1 and S_2. If mathematical distance of the speech clips S_1 and any speech signal S_2 which the speech library is inquired $\boldsymbol{DH} \le \tau$, then their perceptual content are treated as the same, the retrieval is successful, and otherwise it is not successful.

Generally, under the condition that the speech is not subjected to noise processing, since the content of query speech S_1 and the speech signal S_2 in speech library are the same, the threshold τ=0 indicates that the retrieval is successful. Retrieval process of the proposed algorithm is shown as follows.

Input: submit speech
Output: the encrypted speech corresponding to the hash sequence

```
Time to submit a speech clips
[y, Fs] = audioread ('liang1641.MP3')      /* Submit speech */
y1 = hash_generation (y);                  /* Exact hashing */
mm= hash_generation_logistic(audio);       /* Exact all speech hashing sequence*/
j = 1;
for i = 1 to length(mm)-1
    y2 = mm{i};                            /* Hash index table structure */
    K(j) = h_distance (y1, y2);            /* Hamming distance matching process */
    if (K(j) == 0) then break;             /* Match success */
j = j + 1;
end for
The final time for submitting speech to extracting hashing sequence and matching.
```

Decryption process. When retrieval is successful, the encrypted speech is returned for user. Therefore, it needs to be decrypted. Fig. 4 shows the flow diagram of decryption algorithm.

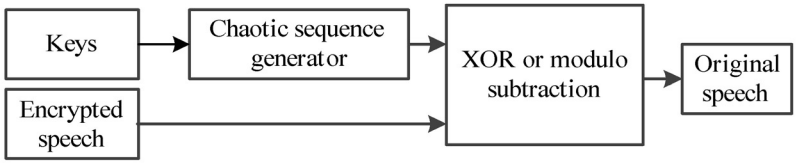


Fig. 4. The flow diagram of decryption algorithm

The specific decryption process is as follows:

Step 1. Obtain the initial values x_0, x_1 and generate a chaotic sequence according to Eq. (5).

Step 2. Chaotic sequence $\boldsymbol{C}$ is constructed to decrypt the speech to obtain the speech $s'(t)$.

4 Experimental Results and Analysis

The text speech data is constituted with the Texas Instruments and Massachusetts Institute of Technology (TIMIT) and Text to Speech (TTS). It is composed of different content recorded by men and women in Chinese and English. The frequency of the speech clips is 16 kHz, the sampling precision is 16 bit, and the length is 4 s. The test speech library with five different speech formats is used in this experiment data, and there are 2,000 speech clips. The five speech formats are: WAV, MP3, FLAC, M4A and OGG five different speech clips, a total of 2,000 speech clips. Experimental hardware platform: Intel (T) Core(TM) i5-2450M CPU, 2.50 GHz, computer memory 8 GB. Software environment: Windows 7, MATLAB R2016a.

4.1 Discrimination Analysis

According to Eq. (11), by estimating BERs, it can be determined whether the submitted query speech hash sequence and the speech hash sequence in the speech library have the same speech content. If the BER value is less than the threshold, the same content is assumed; otherwise, it is considered as different speech contents. Discrimination and robustness are the main characteristics of the retrieval algorithm.

In order to clearly determine these two characteristics, this paper usually uses false accept rate (FAR) and false reject rate (FRR). The FAR and FRR are defined as follows:

$$\mathrm{FAR}(\tau)=\int_{-\infty}^{\tau} f(x|\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}}dx, \tag{14}$$

$$\mathrm{FRR}(\tau)=1-\int_{-\infty}^{\tau} f(x|\mu,\sigma)=1-\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}}dx, \tag{15}$$

where, τ is perceptual threshold, μ represents the BER mean, σ is called BER variance, x is false acceptance rate.

The BER of the hash value of different speech content basically obeys the normal distribution. The hash values is extracted from the above 2,000 speech clips as retrieval digest. 860,030 BER data is obtained by comparing the two perceptual hashing values of 2,000 speech clips. The normal distribution of BERs is shown in Fig. 5.

Obviously, it can be seen that BERs are normal distributions with mean μ=0.4996 and standard deviation σ=0.0296. Under this normal distribution, FARs are generated at different thresholds. The experimental results show that the probability distributions of the BER values of different speech almost overlap the standard normal distributions. Therefore, the hash distance obtained by the proposed algorithm approximately obeys normal distribution.

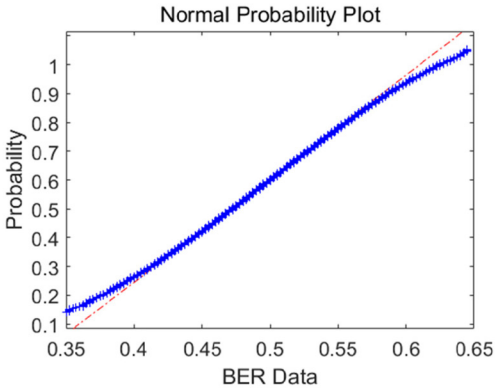


Fig. 5. Normal distribution of BER

As shown in Fig. 5, the FAR reflects the discrimination of the proposed algorithm. The larger the threshold τ is, the larger the FAR value is, the lower the discrimination is and the low the anti-collision performance is. The BER of 2,000 speech clips composed of five different speech formats under different thresholds is shown in Table 1.

Table 1. The FAR value of different algorithms comparison

Threshold τ	Proposed	Ref. [11]	Ref. [12]	Ref. [16]
0.10	7.8188×10^{-42}	1.3044×10^{-18}	1.1957×10^{-20}	1.5974×10^{-29}
0.15	1.7162×10^{-32}	1.2820×10^{-14}	5.5184×10^{-16}	5.5663×10^{-23}
0.20	2.2151×10^{-24}	3.7581×10^{-11}	5.7547×10^{-12}	2.6311×10^{-17}
0.25	1.6928×10^{-17}	3.3074×10^{-8}	1.3665×10^{-8}	1.6986×10^{-12}
0.30	7.7445×10^{-12}	8.8264×10^{-6}	7.4813×10^{-6}	1.5143×10^{-8}
0.35	2.1626×10^{-7}	7.2619×10^{-4}	9.6520×10^{-4}	1.9000×10^{-5}
0.40	3.8291×10^{-4}	0.0190	0.0306	0.0035

As shown in Table 1, when the matching threshold τ between 0.1 and 0.4, the proposed algorithm have good discrimination, which can distinguish different speech and complete the content preserving completely. From Table 1, it can be seen that when the matching threshold τ=0.40, the proposed algorithm has a better discrimination than the algorithm in Ref. [11-12, 16], there are only 3.83 speech clips will be falsely accepted in 10^4 speech clips, which indicates that the proposed algorithm is resistant to collisions. The ability is very strong, and the error rate is much lower than the BER in Ref. [11-12, 16]. It can be seen that the proposed algorithm has a good discrimination.

4.2 Robustness Analysis

In order to test the robustness of the proposed algorithm, firstly, selecting 400 speech clips with different speech format from 2,000 speech clips are used to perform 7 kinds of content preserving operations as shown in Table 2, then calculate the BER value of five different formats speech library after the content preserving operation.

Table 2. Content preserving operations

Operating means	Operation method
Echo addition	Superimposed attenuation 60%, delay 300 ms,
Volume1	Volume up 50%.
Volume2	Volume down 50%
Resampling 1	Sampling frequency decreased to 8 kHz, and then increased to 16 kHz
Resampling 2	Sampling frequency increased to 32 kHz, and then dropped to 16 kHz
Noise addition	*SNR* = 50 dB narrowband Gaussian noise
Butterworth Filter	6 order Butterworth low-pass filtering, Cutoff frequency of 3.4 kHz

As shown in Table 2, the speech signal is usually processed by reducing volume, increasing volume, the butter filter, the echo, and noise addition. The speech signal changes in some way without affecting the speech content. The robustness analysis makes the BERs of the original speech and content preserving operations speech clips smaller than the set threshold.

Calculating the BER averages of the five different format speech clips for various content preserving operations. The BER averages for each speech library are shown in Table 3.

Table 3. BER means for content preserving operations comparison

Parameters	WAV	MP3	M4A	OGG	FLAC
Echo addition	0.1420	0.1474	0.1652	0.1506	0.1452
Volume1	1.1875×10^{-4}	0.0013	0.0283	0.0185	5.6250×10^{-5}
Volume2	0.0230	0.0199	0.0695	0.0301	0.0173
Resampling 1	8.2500×10^{-4}	0.0017	0.0352	0.0063	8.1875×10^{-4}
Resampling 2	0.0074	0.0077	0.0555	0.0182	0.0081
Noise addition	0.0088	0.0070	0.0607	0.0392	0.0079
Butterworth Filter	0.1754	0.1565	0.1673	0.1573	0.1632

As shown in Table 3, that the maximum speech BER mean value of the same content in the five different formats is 0.1754, which shows that the proposed algorithm has good robustness.

Calculating the average BER and maximum BER of the proposed algorithm and Ref. [12-13] in the 400 WAV format speech clips of content preserving operations.

The average BER and maximum BER in Ref. [12-13] are shown in Table 4.

Table 4. Robustness comparison

Operating means	Mean			Max		
	Proposed	Ref. [12]	Ref. [13]	Proposed	Ref. [12]	Ref. [13]
Echo addition	0.1420	0.3016	0.1618	0.2125	0.3778	0.2375
Volume1	1.1875×10^{-4}	0.0174	0.0501	0.0050	0.1361	0.1194
Volume2	0.0230	0.0239	0.0404	0.0700	0.0944	0.0903
Resampling 1	8.2500×10^{-4}	0.0209	0.0014	0.0125	0.0889	0.0139
Resampling 2	0.0074	0.1728	0.0428	0.0775	0.4222	0.1597
Noise addition	0.0088	0.2502	0.0979	0.1050	0.5194	0.3681
Butterworth Filter	0.1754	0.2804	0.1874	0.2800	0.4556	0.2819

As shown in Table 4, it can be seen that the maximum value of the average BER of the proposed algorithm for speech content operation is 0.1754, which is lower than the algorithm in Ref. [12-13]. Where, the maximum value of the average BER in Ref. [12] is 0.3016, and the maximum value of the average BER in Ref. [13] is 0.1874. In addition, the maximum value of the maximum BER for the proposed algorithm is 0.2800, which is lower than the algorithm in Ref. [12-13]. Where, the maximum value of the maximum BER in Ref. [12] is 0.5194, and the maximum value of the maximum BER in Ref. [13] is 0.3681. Therefore, the proposed algorithm has better robustness.

4.3 Recall Rate and Precision Rate Analysis

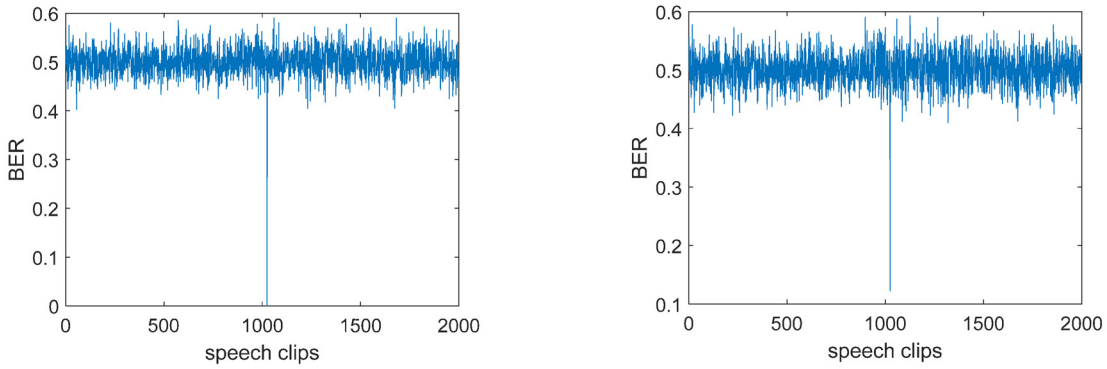
In this paper, recall and precision rate are used to measure retrieval characteristics. Where, f_T is a keyword-related speech clip in the query, f_L is a speech clip related to the keyword but not detected, and f_F is the number of keywords that are not related. The precision R and precision P can be calculated by Eq. (16) and Eq. (17).

$$R = \frac{f_T}{f_T + f_L} \times 100\% . \tag{16}$$

$$P = \frac{f_T}{f_T + f_F} \times 100\% . \tag{17}$$

When the query speech's perceptual hash sequence matches the system hash index table, this paper assumes that the similarity threshold is T_2, $0<T_2<0.5$, and if the Hamming distance is $\boldsymbol{DH}(S_1, S_2)<T_2$, the matching is successful. In the discrimination test, the resulting hash sequences are matched two by two, with a minimum BER of 0.2150. For robustness tests, the maximum BERs for the hash sequence after the original speech and content preserving operations is 0.1754. Therefore, the similarity threshold T_2 in the proposed algorithm is reduced to $0.1754<T_2<0.2150$. In order to avoid detection errors and make it have a high precision rate, the proposed algorithm sets the threshold T_2 to 0.2.

Firstly, the system hash index table is extracted from 2,000 speech clips by the above-mentioned biological hashing construction method, and 2,000 speech clips can be effectively matched in the system hash index table. Then selecting the 1,024th clips of the 2,000 original speech as the query speech. The hash sequence of the query speech is extracted by the biological hashing extraction algorithm mentioned previously as retrieval digest. The hash sequence of the query speech clips and the system hash index tables formed by all 2,000 speech clips are respectively matched, and the BERs to be calculated and compared has similar threshold, if it is less than the threshold, the match is successful, otherwise, it continues to match the next hash sequence. The proposed algorithm selects the 1,024th speech in the encrypted speech library as the query speech, and the matching results of the original speech and the added echo speech are shown in Fig. 6.



(a) Original speech match results (b) Add echo match results

Fig. 6. Original speech and echo matching results

As shown in Fig. 6(a), we can see that when the speech clips in the 1,024th clip is used as the query speech, the hash sequence generated by the query speech and the 2,000-speech system hash index table are queried in turn. Only the 1,024th clip hash sequence is successful, the other 1,999 clips hash sequence failed to match. It can be seen from the Fig. 6(a) that it is an exact match and the threshold is 0, so the recall rate and precision rate are all 100%.

As can be seen from Fig. 6(b), the speech clips in the 1,024th clip is used as the query speech, and the speech content is echo preserving operations. The hash sequence of the query speech is extracted. The hash sequence generated by the query speech is queried in turn with the system hash index table consisting of 2,000 speeches. Only the 1,024th clip hash sequence is successfully matched. The other 1,999 clips hash sequences fail to match. It can be seen from the Fig. 6(b) that it match completely, and the threshold is less than 0.1. Since the hash sequence is unchanged after the content preserving operations, its recall and precision rate are all 100%.

According to Eq. (16) and Eq. (17), the more similar the two speech clips are, and the smaller the threshold value is, which indicate that the recall rate is higher. In order to avoid detection errors and have a high precision, this paper sets the threshold to 0.2. Therefore, the threshold is inversely proportional to the recall rate and the precision rate. Table 7 shows the threshold comparison results for the proposed algorithm and the algorithm in Ref. [12-13, 16].

As shown in Table 5, we can see that the threshold of the proposed algorithm is 0.2, which is lower than Ref. [12-13, 16], so the proposed algorithm is better than the algorithm in Ref. [12-13, 16].

Table 5. Different algorithm thresholds comparison

Algorithm	Proposed	Ref. [12]	Ref. [13]	Ref. [16]
Threshold τ	0.2	0.3	0.3	0.25

After the query, the 1,024th speech clip echo addition, noise addition, and the hash sequence generated by the query speech clip is matched with the hash sequence in the system hash index table. Since the hash sequence is unchanged after the content preserving operations, its recall rate and precision rate are all 100%. The recall rate and precision rate of the proposed algorithm under different content preserving operations are shown in Table 6.

Table 6. Retrieval rate and precision rate

Operating means	Retrieval rate	Precision rate
Echo addition	100%	100%
Volume1	100%	100%
Volume2	100%	100%
Resampling 1	100%	100%
Resampling 2	100%	100%
Noise addition	100%	100%
Butterworth Filter	100%	100%

Combining the thresholds in Table 5, the proposed algorithm has better retrieval efficiency than the algorithm in Ref. [12-13, 16], and has good robustness and discrimination. In addition, the hashing length of the proposed algorithm is 400, the length of hash sequence in Ref. [12] is 1,024, and the length of the hash sequence in Ref. [13] is 640, the length of the hash sequence in Ref. [16] is 256, which indicates that compactness of the proposed algorithm is also better, so the matching time is reduced and the retrieval efficiency is improved. And after different content preserving operation of the speech data, recall and precision rate are all 100%.

4.4 Retrieval Efficiency Analysis

In order to test the retrieval efficiency of the proposed algorithm, we constructed five speech library of 10 speech clips, 500 speech clips, 1,000 speech clips, 1,500 speech clips, and 2,000 speech clips. A speech was randomly selected from five speech library as query speech. The retrieval time in the five speech library is the total time for submitting the speech clip to extracting biological hashing sequence and matching system hash index table. The retrieval efficiency of the proposed algorithm and the algorithm in Ref. [12-13, 16] is shown in Fig. 7.

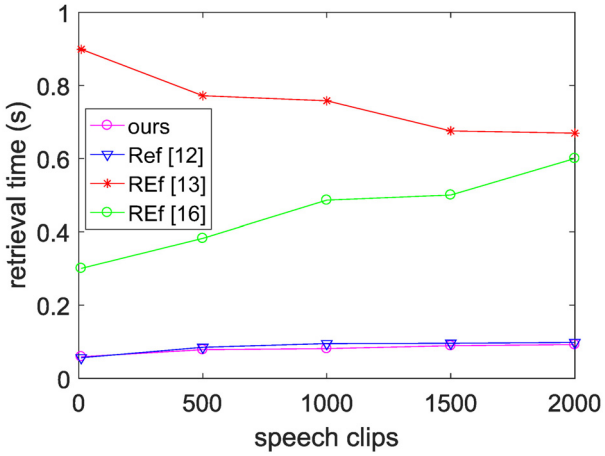


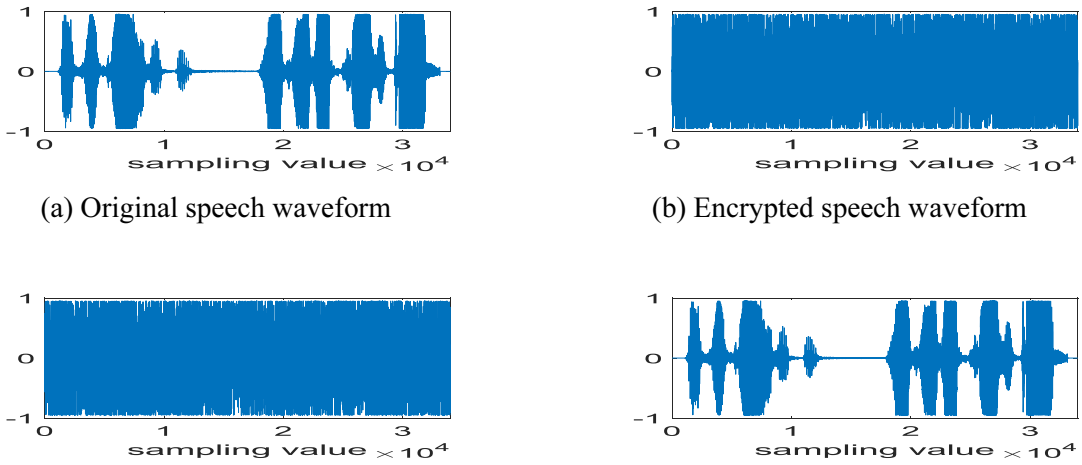
Fig. 7. Different algorithm retrieval efficiency comparison

As shown in Fig. 7, that the retrieval time of each speech clips of the proposed algorithm is much lower than the time consumption of the algorithm in Ref. [13, 16], which is slightly lower than the

algorithm in Ref. [12]. The proposed algorithm has a good retrieval efficiency, because the proposed algorithm uses the combination of DWT transform and Logistic pseudo random matrix to extract speech perceptual features to construct biological hashing, the schedule is simple and the data is reduced obviously, so the retrieval efficiency is improved. However, the algorithm in Ref. [12-13, 16] only uses speech feature to extract hashing value, and does not reduce the amount of speech data, which causes the reason for the large amount of computation and low efficiency of retrieval. Due to the advantages of the proposed algorithm in the retrieval efficiency, it can meet the fast retrieval requirements in the encrypted speech library.

4.5 Security Analysis

In this paper, the 2D Henon chaotic map encryption algorithm is used to encrypt the speech, and the speech is encrypted by using the chaotic sequence XOR and modulo subtraction operations. Fig. 8 is speech waveform for correct decryption and error decryption.



(a) Original speech waveform

(b) Encrypted speech waveform

(c) Error decryption waveform

(d) Correct decryption waveform

Fig. 8. Speech waveform for correct decryption and error decryption

In Fig. 8(a) is the original speech, Fig. 8(b) is the encrypted speech, Fig. 8(c) is the speech waveform in which the key error is not decrypted, and Fig. 8(d) is the speech waveform in which the key is correctly decrypted. The encrypted speech waveform is evenly distributed and sounds like a piece of noise.

(1) Key sensitivity: As shown in Fig. 8(c), it can be seen that initial values subtle differences in the initial values can lead to speech decryption failure. This is due to the fact that the chaotic sequence is sensitive to the initial value. When the initial value is deviated, two completely different chaotic sequences will be generated.

(2) Anti-attack: This paper uses a chaotic sequence based on a 2D Henon map. From Section 2.1 it is known that by adjusting x_0, x_1 (as two keys), more complex chaotic sequences can be generated. This complex chaotic sequence has higher security and anti-attack than the chaotic sequence generated by the one-dimensional Logistic mapping (a key x_1).

(3) Complexity and Security: The chaotic sequence generated by the 2D Henon map is more complex, and the chaotic sequence is sensitive to the initial values x_0, x_1. When the initial values x_0, x_1 have small changes, and iterations of 16,000 times will generate large chaotic sequences.

In this paper, we use the Henon encryption algorithm to encrypt speech. In the process of biological hashing extraction, we extract speech feature by using logistic to generate pseudo random matrix, and it could be regarded as one-time encryption process. Therefore, this paper encrypts the speech and the hash sequence, the security is very high.

4.6 Recoverability Analysis

In this paper, if the feature template is destroyed, the original biometric template is invalidated by changing the logistic pseudo random key x_0 and x_1, and a new template is generated by the new x_0 and x_1. Therefore, the proposed algorithm has a good revocation.

5 Conclusions

In the cloud storage environment, in order to achieve efficient content-based encrypted speech retrieval, a retrieval algorithm for encrypted speech based on biological hashing was proposed. The experimental results show that the biological hashing scheme of the proposed algorithm has good robustness, discrimination and one-wayness, as well as high retrieval efficiency and accuracy of the retrieval model, it can handle with five different formats of speech clips include: WAV, MP3, FLAC, OGG and M4A. In order to improve the security of the encrypted speech data uploaded to the cloud and the security of the hash digest in the system hash index table, the proposed algorithm uses the Henon encryption algorithm to encrypt the speech library, when constructing the biological hashing, we extract speech feature by using Logistic to generate pseudo random matrix, and it could be regarded as one-time encryption process. Therefore, the security of encrypted speech in the cloud and retrieval digest of the system hash index table is guaranteed. When the generated hash sequence is destroyed, a new biological hash sequence can be generated with the new key. Therefore, it has good revocability. The shortcoming of this paper is that the encrypted speech retrieval object can only be a fixed-length speech clips, without considering the fuzzy retrieval of the long speech clips.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61862041, 61363078). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] C. Glackin, G. Chollet, N. Dugan, N. Cannings, J. Wall, S. Tahir, M. Rajarajan, Privacy preserving encrypted phonetic search of speech data, in: Proc. Acoustics, Speech and Signal Processing, 2017.

[2] L. Dietz, C. Xiong, J. Dalton, E. Meij. The second workshop on knowledge graphs and semantics for text retrieval, analysis, and understanding (KG4IR), in: Proc. ACM International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.

[3] Z. Xia, Y. Zhu, X. Sun, Towards privacy-preserving content-based image retrieval in cloud computing, IEEE Transactions on Cloud Computing 6(1)(2018) 276-286.

[4] Y. Chen, J. Wang, Y. Bai, Probabilistic semantic retrieval for surveillance videos with activity graphs, IEEE Transactions on Multimedia 21(3)(2019) 704-716.

[5] J. Ahn, B. Jo, S. Jung, Multiple domain-based spatial keyword query processing method using collaboration of multiple IR-Trees, in: Proc. the 7th International Conference on Emerging Databases, 2018.

[6] F. Gao, M.A. Basu, E-business information fuzzy retrieval system based on block chain anti-attack algorithm, Journal of Intelligent & Fuzzy Systems 35(4)(2018) 4475-4486.

[7] L. Teng, H. Li, J. Liu, An efficient and secure cipher-text retrieval scheme based on mixed homomorphic encryption and multi-attribute sorting method, International Journal of Network Security 20(5)(2018) 872-878.

[8] Q.-Y. Zhang, W.-J. Hu, Y.-B. Huang, S.-B. Qiao, An efficient perceptual hashing based on improved spectral entropy for speech authentication, Multimedia Tools and Applications 72(2)(2018) 1555-1581.

[9] H. Kelkoul, Y. Zaz, H. Tribak, A robust combined audio and video watermark algorithm against cinema piracy, in: Proc. IEEE 6th International Conference on Multimedia Computing and Systems, 2018.

[10] Y. Zheng, Y. Cao, C.H. Chang, Facial biohashing based user-device physical unclonable function for bring your own device security, in: Proc. IEEE International Conference on Consumer Electronics, 2018.

[11] H. Zhao, S.-F. He, A retrieval algorithm for encrypted speech based on perceptual hashing, in: Proc. the 12th Natural Computation, Fuzzy Systems and Knowledge Discovery, 2016.

[12] H.-X. Wang, L. Zhou, W. Zhang, S. Liu, Watermarking-based perceptual hashing search over encrypted speech, in: Proc. the International Workshop on Digital Watermarking, 2013.

[13] G.-Y. Hao, H.-X. Wang, Perceptual Speech hashing algorithm based on time and frequency domain change characteristics, in: Proc. the Symposium on Information, Electronics, and Control Technologies, 2015.

[14] S.-S. Jin, A resilience mask for robust audio hashing, IEICE Transactions on Information and Systems 100(1)(2017) 57-60.

[15] S. Aljawarneh, M.B. Yassein, A resource-efficient encryption algorithm for multimedia big data, Multimedia Tools and Applications 76(21)(2017) 22703-22724.

[16] S.-F. He, H. Zhao, A retrieval algorithm of encrypted speech based on syllable-level perceptual hashing, Computer Science and Information Systems 14(3)(2017) 703-718.

[17] J.-S. Sun, J.-Y. Zhang, Y. Yang, Effective audio fingerprint retrieval based on the spectral sub-band centroid feature, Journal of Tsinghua University (Science and Technology) 57(4)(2017) 382-387.

[18] D. Chen, W. Zhang, Z. Zhang, Audio retrieval based on wavelet transform, in: Proc. 16th Computer and Information Science (ICIS), 2017.

[19] Y. Liu, D. Hatzinakos, Biohashing for human acoustic signature based on random projection, Canadian Journal of Electrical and Computer Engineering 38(3)(2015) 266-273.

[20] P. Lacharme, Revisiting the accuracy of the biohashing algorithm on fingerprints, IET Biometrics 2(3)(2013) 130-133.

[21] B. Ramalingam, D. Ravichandran, A.A. Annadurai, A. Rengarajan, J.B.B. Rayappan, Chaos triggered image encryption-a reconfigurable security solution, Multimedia Tools and Applications 77(10)(2018) 11669-11692.

[22] P. Ping, F. Xu, Y.-C. Mao, Z.-J. Wang, Designing permutation–substitution image encryption networks with Henon map, Neurocomputing 283(2018) 53-63.