


An efficient retrieval algorithm of encrypted speech based on inverse fast Fourier transform and measurement matrix

Qiuyu ZHANG*, Zixian GE, Liang ZHOU, Yongbing ZHANG

School of Computers and Communication, Lanzhou University of Technology, Lanzhou, P.R. China

Received: 22.08.2018

Accepted/Published Online: 11.02.2019

Final Version: 15.05.2019

Abstract: In this paper, we present an efficient retrieval algorithm for encrypted speech based on an inverse fast Fourier transform and measurement matrix. Our approach improves query performance, as well as retrieval efficiency and accuracy, compared to existing content-based encrypted speech retrieval methods. Our proposed algorithm constructs a perceptual hash scheme using perceptual hash sequences from original speech files. By classifying the sequences and applying run-length compression, we decrease the cloud storage required for the hash index. We secure the speech database by encrypting it with Henon chaos scrambling, which offers excellent resistance to attacks. Experimental results show that the robustness, discrimination, and feature extraction efficiency of our proposed method are better than the existing alternatives, with good recall and precision ratios and with high retrieval efficiency and accuracy.

Key words: Encrypted speech retrieval, perceptual hashing, inverse fast Fourier transform, measurement matrix, Henon chaotic scrambling, speech feature extraction

1. Introduction

With the rapid development of Internet technology, speech is widely used in the fields of radio and television, court evidence, telephone service, conference recording, etc. The growth in the amount of multimedia information has been exponential. How does one retrieve meaningful information from countless voice messages? How does one do so efficiently, accurately, and securely [1]? These are important challenges in the field of speech retrieval. A good content-based retrieval algorithm for encrypted speech realizes the retrieval process of speech content information by researching the physical features of speech, such as frequency spectrum and amplitude, and auditory features such as pitch and timbre [2], not only protecting the privacy of the speech data but also retrieving data efficiently and accurately. Therefore, research into such algorithms has significant theoretical and practical value.

The technologies used in current speech retrieval research fall into two general categories: text-based or keyword retrieval [3], and content-based retrieval [4]. The latter is further divided into those that make use of feature matching [2], deep learning [5], and sorting searches [6]. Feature extraction is an important step in speech retrieval, with most research making use of perceptual hashes [7–9] or audio fingerprints [10–12]. In addition, in order to improve system security, some good methods have been applied, such as zero-leakage biometric protection [13], cancelable biometrics [14,15], biometrics hashing [16,17], and secret key generation [18].

We now provide a brief summary of related work. In 2013, Wang et al. proposed a retrieval method for

*Correspondence: zhangqylz@163.com

encrypted speech using perceptual hashing [19]. It operates on speech encrypted with Chua's chaotic circuit and a piecewise linear (PWL) and uses the speech zero-crossing rate to extract speech features for retrieval. Wang's method is fast and reliable but requires a complex encryption algorithm and offers poor discrimination, resulting in low retrieval accuracy and efficiency. Ibrahim et al. proposed a multiple-keyword rank-order search method for securely encrypted cloud data [20]. It excels in safety and retrieval efficiency but with weaker retrieval accuracy. Wang et al. extracted perceptual hashing features using time and frequency domain change characteristics, dividing speech into these domains to extract the perceptual hashing digest [21]. It offers good discrimination, robustness, security, and retrieval accuracy, but at the expense of feature extraction speed. Lin proposed an encrypted speech retrieval algorithm using the speech pitch period as a feature to construct a hash and encrypting the speech with AES [22]. The algorithm is robust with high retrieval accuracy but with weak discrimination and a complex encryption algorithm. Zhao et al. proposed an encrypted perceptual hashing retrieval method using the multifractal characteristic. It has good robustness and high retrieval accuracy but poor discrimination and low retrieval efficiency [23]. He et al. proposed a retrieval method for encrypted speech using syllable-level perceptual hashing. It has good robustness and high security, but poor speech feature discrimination and mediocre retrieval efficiency [24]. Glackin et al. presented a cloud-based encrypted speech retrieval method with good security and retrieval accuracy [25]. However, the retrieval efficiency is not high due to its use of a high-complexity encryption algorithm. Cancelable biometrics [14,15] transform the original biometric identity of a user to a pseudo-biometric identity that is used for storage and matching purposes. The use of a pseudo-identity mitigates privacy risks and allows revocability in the case of compromise. It has good security, but for the false acceptance rate (FAR), the matching efficiency needs to be promoted. Similarly, zero-leakage biometric protection [13] and biometric hashing [16,17] have good security but are affected by algorithm complexity for weak system efficiency.

These research results show the many shortcomings in existing content-based encrypted speech retrieval approaches. In terms of speech feature extraction, robustness, discrimination, and algorithm efficiency are mutually exclusive in existing algorithms and cannot be balanced well. During speech encryption, most algorithms result in loss-of-speech features, which negatively and seriously affect subsequent retrieval results. The efficiency of an encrypted speech retrieval algorithm is affected by the construction of the speech hashing scheme, speech encryption, and matching retrieval. Using a more complex encryption algorithm that better preserves features leads to inefficient retrieval. A good retrieval algorithm needs excellent retrieval efficiency and accuracy but existing algorithms need improvements in both areas. In addition, in order to store the speech perceptual hashing feature, existing methods embed the extracted hash sequence into the encrypted speech with a digital watermark, which assists authentication but reduces retrieval efficiency and accuracy.

We organize our paper as follows. Section 2 introduces the related principles. Section 3 describes our system model. Section 4 presents our proposed method. Section 5 gives the experimental results and performance analysis of our method as compared with existing methods. Finally, Section 6 concludes our paper.

2. Related theory

2.1. Inverse fast Fourier transform

For the time function $x(t)$ ($-\infty < t < +\infty$), if $x(t)$ satisfies the Dirichlet condition and is absolutely integrable, then for the Fourier transform of $x(t)$ and its inverse transform [26], the Fourier transform function is as follows:

$$F(\omega) = \int_{-\infty}^{+\infty} x(t)exp(-j\omega t)dt. \quad (1)$$

The inverse Fourier transform is:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)exp(j\omega t)d\omega, \quad (2)$$

where $x(t)$ is the original signal, $F(\omega)$ is the spectrum of $x(t)$, ω is the angular frequency, and j is an imaginary number. In this paper, we use the inverse Fourier transform by treating the amplitude of each frame of the speech signal as $F(\omega)$ of Eq. (2).

2.2. Henon chaotic scrambling

A Henon map [27] is an iterative mapping of chaos in two-dimensional space, and its calculation function is as follows:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n, \\ y_{n+1} = bx_n. \end{cases} \quad (3)$$

From a set of original data (x_0, y_0) and parameters (a, b) , we can obtain two chaotic sequences $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_n\}$ and $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_n\}$ using Eq. (3). Through experimental comparison, the correlation between the speech sample points is lower following encryption using chaotic sequence \mathbf{Y} . Therefore, we use chaotic sequence \mathbf{Y} in the Henon map to encrypt the speech.

2.3. Henon chaotic scrambling-based partial Hadamard measurement matrix construction

The measurement matrix (MM) [28] is a data dimensionality reduction method based using compression sensing technology that can greatly reduce the amount of data without changing the features. The method is divided into a random MM and a deterministic MM. The deterministic MM has the advantages of low computational complexity and easy implementation in hardware or software. The partial Hadamard MM used in this paper is a type of deterministic MM. Compared with other deterministic MMs, the reconstruction effect of partial Hadamard MMs is better for a given number of measurements; that is, the accurate reconstruction of the MM requires fewer measurements. Therefore, we efficiently generate a deterministic MM by combining the Henon chaotic map with a partial Hadamard MM. The construction method has the following steps.

Step 1: Generate an $N \times N$ Hadamard matrix ϕ .

Step 2: Select the first 380 bits of the X sequence in the Henon map to generate chaotic sequences $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_L\}$, $L = 380$.

Step 3: Reclassify the natural sequence $n = [1, 2, 3, \dots, N]$ by using the chaotic sequence from Step 2 to obtain $S = \{S_1, S_2, \dots, S_N\}$, where $N = 380$, $i \in [1, N]$.

Step 4: Take M row elements $\phi(S_1, :), \phi(S_2, :), \dots, \phi(S_M, :)$ from the matrix ϕ according to the S sequence, and create an $M \times N$ measurement matrix \mathbf{B} .

3. The system model

The diagram in Figure 1 depicts the model of our system for efficient retrieval of encrypted speech using IFFT and MM. The model has three main parts: the system hash index table construction, encrypted speech library construction, and user speech retrieval.

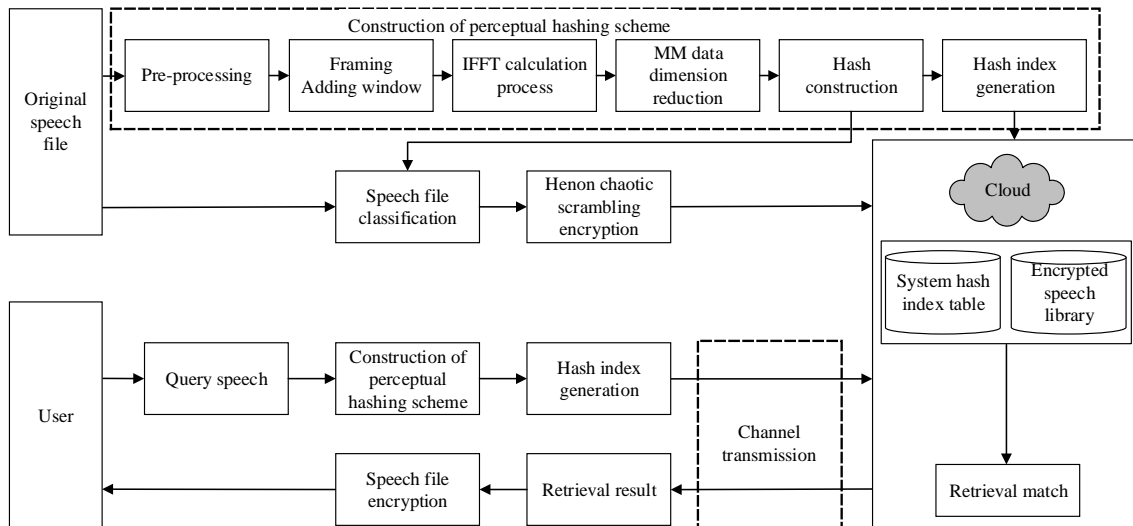


Figure 1. Diagram of the encrypted speech retrieval algorithm.

As shown in Figure 1, in the hash index table construction step, we process the original speech file by preprocessing, framing, and an adding window. In order to get a smooth signal we perform a framing process, and we add a Blackman window, which is a second-order raised cosine window with wide main lobe and low side lobes, commonly used to detect signals with different frequencies. After calculating the IFFT and performing MM data dimension reduction, we convert the original speech into a binary hash sequence within a larger hash structure. We classify the speech using the characteristics of the hash sequence and combine the characteristics with run-length compression to reduce the hash sequence data further. We then construct a hash index table for the generated hash sequence and upload it to the cloud. The speech hash sequence in the system hash index table establishes a one-to-one mapping relationship with the encrypted speech library.

For the encrypted speech library construction step, we classify the original speech file according to the characteristics of the speech hash sequence, and then we use the Henon chaotic scrambling encryption method to encrypt the original speech and upload it to the encrypted speech library on the cloud server.

During user speech retrieval, we first extract the hash sequence of the query speech using the previously constructed perceptual hashing scheme, and matching of the perceptual hashing sequence is done with the hashing sequence in the system’s hash index table. We set a similarity threshold to obtain the correct matching result. Finally, we provide the encrypted speech file in the encrypted speech library that corresponds to the matching result as a retrieval result, decrypting it and returning it to the user.

4. The proposed retrieval algorithm of encrypted speech

Having provided both some theoretical fundamentals and an overview of our system, we now present the implementation of the three main parts in more detail.

4.1. Construction of system hash index table

4.1.1. Construction of perceptual hashing scheme

The proposed efficient speech perceptual hashing scheme requires us to calculate the IFFT on the original speech file and then use a partial Hadamard measurement matrix to reduce the dimensionality. The algorithm steps for this process are as follows.

Step 1: Preprocessing. For better feature extraction, we preprocess the speech clip $s(t)$ by flattening the signal spectrum, producing the signal $s(t)'$ as a result.

Step 2: Framing and adding window. We divide the speech clip $s(t)'$ into m nonoverlapping frames denoted as $f_i = \{f_i(n)|n = 1, 2, \dots, L/m, i = 1, 2, \dots, m\}$. L is the length of speech clip, m is the total number of frames, and $f_i(n)$ is the n th sample value of the i th frame. We process these m frames by adding a Blackman window to reduce the truncation effect.

Step 3: Feature calculation. We calculate the feature vector $\mathbf{H}' = \{C_i|i = 1, 2, \dots, M\}$ of the i th frame for the original speech signal $s(t)$ according to IFFT transform.

Step 4: Feature data dimensionality reduction. We obtain the final feature vector $\mathbf{H} = \{D_i|i = 1, 2, \dots, N\}$ by compressing the feature vector \mathbf{H}' according to $\mathbf{H} = \mathbf{B} \times \mathbf{H}'$ using the partial Hadamard measurement matrix from Section 2.3.

Step 5: Hash structure. We construct the hash structure using vector \mathbf{H} to generate the hash sequence $\mathbf{h} = \{h(i)|i = 1, 2, \dots, m\}$. The construction formula is

$$h(i) = \begin{cases} 1, & \text{if } H(i+1) > H(i), \\ 0, & \text{Otherwise,} \end{cases} \quad (4)$$

where $i = 1, 2, \dots, m-1$, with the hash sequence length m determined to be 380 according to the following discussion.

A longer hash sequence length provides better robustness and discrimination of the speech perceptual hashing sequence at the expense of feature extraction efficiency and retrieval speed. Shortening the hash sequence length improves system efficiency but degrades the hash sequence performance. We conducted tests that indicate that a value of 380 for the length of the hash sequence maximizes system efficiency and preserves the hash sequence performance.

4.1.2. Construction of system hash table

The construction of the system hash index table is similar to the corresponding rule in the function. That is, we replace x in the function with the keyword used in the search record and then insert the selected keyword into the designated formula to obtain a value. By using this value to represent the hash address of the record store, the hash address of the data is $f(key)$. Figure 2 shows the construction process of the system hash index table.

As shown in Figure 2, the system hash index table maps the speech clips to binary hash sequences of the corresponding position in the hash table using the hash function. Then we establish a one-to-one mapping relationship between the encrypted speech clips in the cloud and the binary hash sequences in the system hash index table. Our algorithm extracts speech features by combining the IFFT and MM and then constructs the hash table, thus establishing a one-to-one mapping relationship between the encrypted speech clips and hash sequences. We sum all the elements of each hash sequence to obtain a value between 167 and 192 for a total of 24 different values used to classify the sequences. Finally, we perform run-length compression on the classified sequences to obtain the final features. For binary hash sequences, run-length compression replaces multiple repeated elements with a single value indicating the repetitions. For example, a 20-digit hash sequence ‘10111101000100111011’ becomes ‘11411312312’ after applying the run-length compression method. Finally, we upload the system hash index table to the cloud.

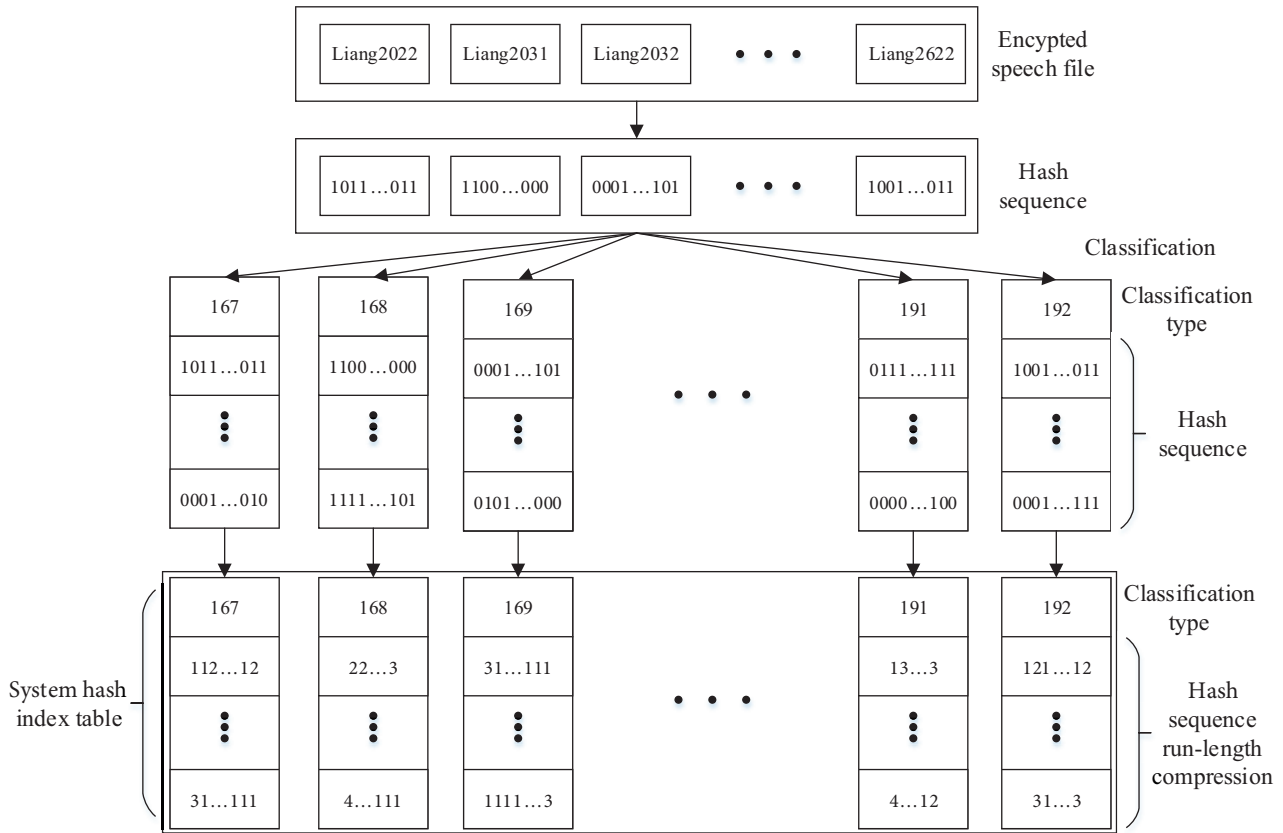


Figure 2. Construction process of the system hash index table.

However, during construction of the system hash index table, duplicate hash values (collisions) are possible. We address collisions using a re-hash method. If a speech file produces a duplicate hash, we reconstruct the hash sequence using Eq. (5) and then place the reconstructed hash sequence in the corresponding position of the system hash index table:

$$h(i) = \begin{cases} 1, & \text{if } H(i) > \text{median}(H), \\ 0, & \text{Otherwise,} \end{cases} \quad (5)$$

where $i = 1, 2, \dots, m$ and $\text{median}(H)$ is the median of the sorted sequence $\mathbf{h} = \{h(i)|i = 1, 2, \dots, m\}$.

4.2. Construction of encrypted speech library

For the sake of secure transmission, we encrypt original speech files with speech scrambling encryption technology using a Henon chaotic map. The speech encryption steps are as follows:

Step 1: We classify the corresponding speech files according to the hash sequence classification method in Section 4.1.

Step 2: We select the encrypted key $[\mu, y_0]$ and generate an original chaotic sequence $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_n\}$ through the Henon chaotic map of Eq. (3).

Step 3: We sort the original chaotic sequence $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_n\}$ (from Step 1) into ascending order to obtain a new sequence $\mathbf{K} = \{k_1, k_2, k_3, \dots, k_j\}$, where $j = 1, 2, 3, \dots, M$. For an original speech signal Y and

an encrypted speech signal Z , $Z(j) = Y(i)$ if the position between the original chaotic sequence position and the new sequence satisfies the mapping relationship $Z(j) = Y(i)$.

Step 4: We replace the original speech signal sample points according to the mapping relationship in Step 2 to obtain an encrypted speech file.

We upload the encrypted speech file to the encrypted speech library in the cloud after the above procedure.

4.3. User speech retrieval

Speech matching retrieval is based on the binary hash sequence generated in Section 4.1 using a normalized Hamming distance and a similarity threshold to find matching speech.

Step 1: Query speech hash construction. Construct the hash sequence of the query according to the method in Section 4.1.

Step 2: Hash sequence matching and retrieval. We match the hash sequence in Step 1 with the elements in the system hash index table. We use the normalized Hamming distance $D(:, :)$, also known as the bit error ratio (BER), whereby the BER is calculated as:

$$BER = D(\mathbf{h}_1, \mathbf{h}_2) = \frac{1}{m} \sum_{j=1}^m |h_1(j) - h_2(j)|, \tag{6}$$

where \mathbf{h}_1 and \mathbf{h}_2 are the perceptual hashing features corresponding to two speech clips and D is the normalized Hamming distance between \mathbf{h}_1 and \mathbf{h}_2 , which represents the ratio of the number of error perceptual hashing bits to the total number of bits.

Figure 3 shows the schematic diagram of the speech retrieval process.

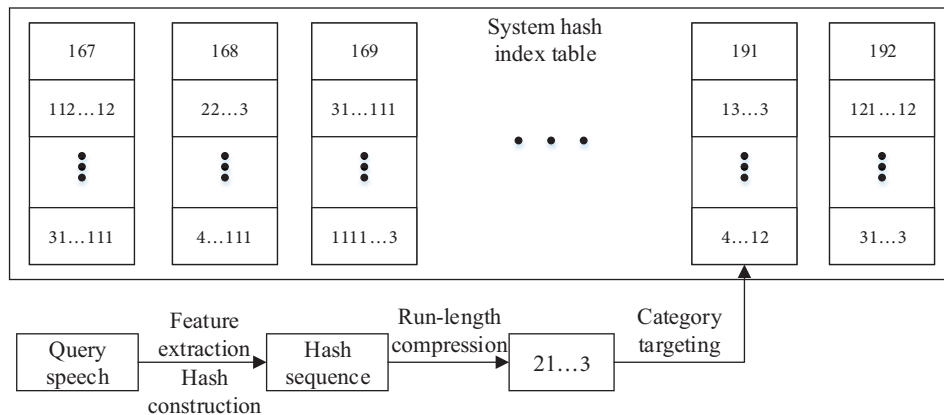


Figure 3. Schematic diagram of the speech retrieval process.

As shown in Figure 3, the overall retrieval process is divided into two stages: query speech feature generation and feature query. The query speech feature construction process first extracts the query using the IFFT and MM to construct the hash, generates the final speech feature via run-length compression, and calculates the sum of elements in each hash sequence to obtain the classification class label. To perform the query, we locate the speech feature in the corresponding classification category sequentially matching the speech features in the category using the normalized Hamming distance and then return these results. The matching process needs to separately calculate the normalized Hamming distance between the query hash sequence \mathbf{h}_1

constructed in Step 1 and the elements $\mathbf{h} = \{h(i)|i = 1, 2, \dots, M\}$ in the system hash table for each speech in the speech library. We compare the normalized Hamming distance $\mathbf{D} = \{d(i)|i = 1, 2, \dots, M\}$ with the set threshold τ to determine the similarity between speeches. If $\mathbf{D} \leq \tau$, then the perceptual content of speech clips is the same (i.e. they relate to the same topic). Otherwise, they are determined to be different.

We must decrypt the encrypted speeches matching the query. Speech file decryption is the inverse process of encryption according to the following steps.

Step 1: Select the encryption key $[\mu, y_0]$ used in the speech encryption process as the decryption key and generate the same chaotic sequence $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_n\}$ used by the encryption process using Eq. (3). The length of the sequence and the number of sample points are unchanged.

Step 2: Sort the chaotic sequences obtained in Step 1 from small to large to obtain a sequence $\mathbf{K} = \{k_1, k_2, k_3, \dots, k_j\}$. For an encrypted speech Z with decrypted speech Y' , $Y'(i) = Z(j)$ according to the positional relationship.

Step 3: Using the mapping relationship in Step 2, we replace the encrypted speech in the opposite way as the encrypted progress to obtain a complete decrypted speech and return the decrypted file to the user.

5. Experimental results and analysis

5.1. Experimental environment and main parameter settings

We conducted an experiment to evaluate our approach using speech samples from the standard Texas Instruments and Massachusetts Institute of Technology (TIMIT) and Text to Speech (TTS) speech library as the test speech. The library consists of 10,000 speech clips stored as 16-bit 16-kHz monophonic recordings. Each one is 4 s long. We ran our test on a computer with a second-generation Intel Core i5 CPU at 2.50 GHz, the operating system was Windows 7 (64 bits) Home, and the simulation platform was MATLAB R2013a. As parameters for our algorithm we chose nonoverlapping frames, speech clip length $L = 32,000$, and number of frames $m = 380$.

5.2. Performance analysis

5.2.1. Encryption and decryption performance analysis

We encrypted and decrypted the speech signal during transmission using the Henon map approach described previously. Figure 4 shows the speech waveforms before and after encryption and decryption.

In Figure 4a is the original speech waveform, Figure 4b the encrypted speech waveform, Figure 4c the speech waveform from incorrect decryption, and Figure 4d the speech waveform correctly decrypted. The encrypted speech waveform is evenly distributed and sounds like a piece of noise.

For key space analysis, the Henon chaotic map showed a rich variation in the mapping sequence under different parameter values. The specific values for parameters a and b determine the nature of the mapping sequence. We selected parameter b in $[-1, 1]$ with a chosen from one of several specific ranges of values so that the mapping sequence generates chaos. The key parameters $[a, b]$ have an infinite range of values with a very large key space, making it almost impossible to break the encryption algorithm in practice.

For key sensitivity analysis, Figure 4 shows that the original speech produced a chaotic, disorderly, and fuzzy speech waveform following encryption. It is impossible to find the characteristics of the original speech signal in the encrypted speech. Therefore, we conclude that this encryption algorithm offers strong security. Correct decryption is possible only with the same key $[a, b]$ and initial value $[x_0, y_0]$ used in the encryption process. Changing the decryption key only slightly produced the incorrect and unusable decryption waveform

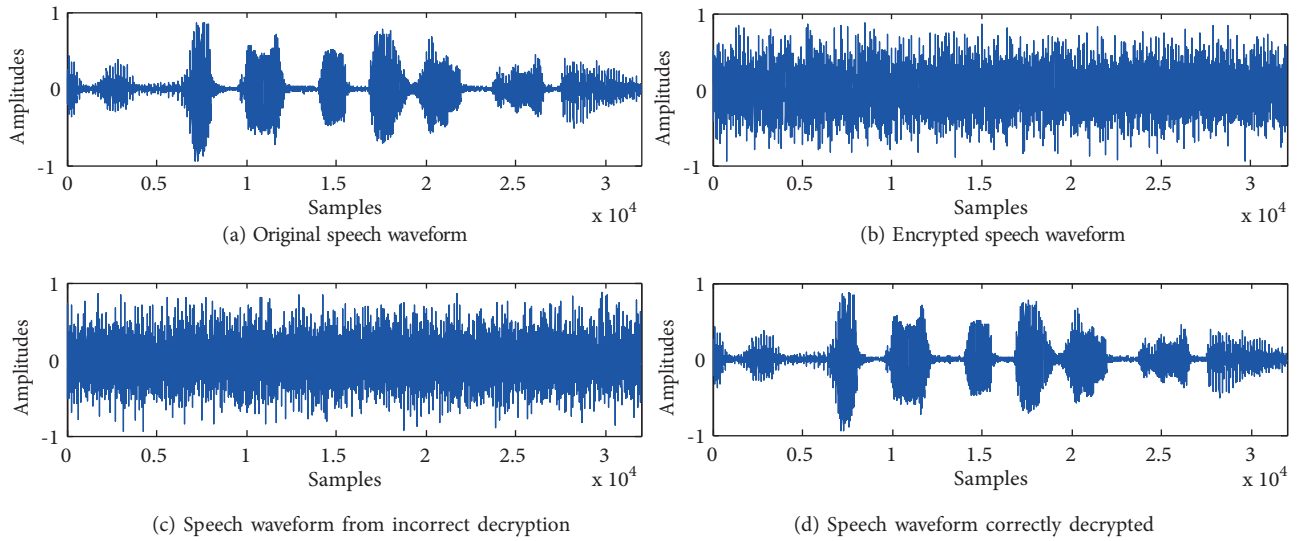


Figure 4. Speech encryption and decryption waveform comparison: (a) Original speech waveform; (b) encrypted speech waveform; (c) speech waveform from incorrect decryption; (d) speech waveform correctly decrypted.

shown in Figure 4c. Therefore, we conclude that our encryption and decryption algorithm offers good key sensitivity.

For correlation analysis before and after speech encryption, using a randomly selected speech clip and 32,000 sample points within it, we calculated $x(i)$ as the abscissa (x-axis) and $x(i + 1)$ as the ordinate (y-axis) and plotted the results before and after encryption, as shown in Figure 5.

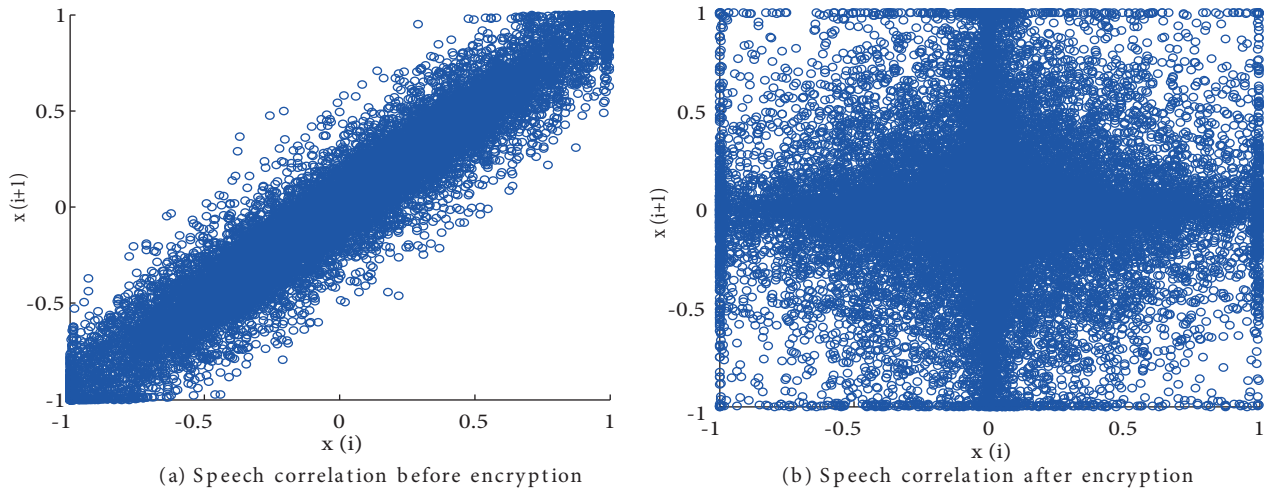


Figure 5. Speech correlation before and after encryption: (a) Speech correlation before encryption; (b) speech correlation after encryption.

To measure resistance to statistical attacks, we made use of Spearman’s rank correlation coefficient [29] as defined by Eq. (7). Smaller correlation coefficient values indicate better encryption performance.

$$\rho = \frac{\sum_i(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i(x_i - \bar{x})^2 \sum_i(y_i - \bar{y})^2}}. \tag{7}$$

We calculated the correlation coefficient of the original speech sample as 0.9182 and that of the encrypted sample as 0.0021. Compared with logistic scrambling encryption with the correlation coefficient of 0.9302 and 0.0073 of the original and encrypted sample, Henon scrambling encryption has smaller correlation coefficients after encryption. The speech has a strong correlation between the sample points before encryption, but very little after encryption. We conclude that our algorithm offers good resistance to statistical attacks.

5.2.2. Performance analysis of perceptual hashing

We evaluate the performance of our perceptual hashing sequence approach according to discrimination and robustness.

For discrimination analysis, as described previously, we measure the similarity between two pairs of speech perceptual hashing sequences by calculating the BER value. The BER of perceptual hashing values from our samples generally followed normal distribution. Comparing hash sequences of 1280 speech data clips produced 819,840 BER values. Figure 6 shows the normal distribution of these resulting BER values.

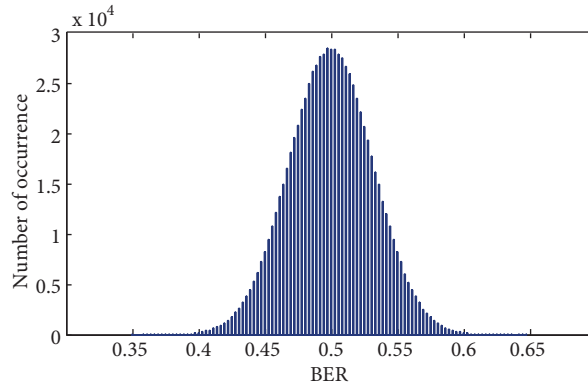


Figure 6. Statistic histogram of 1280 speech clips matching results.

The BER statistical results we obtained by matching the constructed perceptual hashing abstracts ranged between 0.3500 and 0.6472, following the Gaussian distribution $N(\mu, \sigma)$ with the mathematical expectation $\mu = 0.4998$, standard deviation $\sigma = 0.0318$, and minimum value of 0.3500. We conclude that our method offers good discrimination.

To further measure the discrimination of the algorithm under different thresholds, we also make use of the false acceptance rate (FAR) and false rejection rate (FRR) to reflect the discriminability of the proposed algorithm. Larger threshold and FAR values result in lower discriminability and increased collisions, while smaller FRR values result in good discrimination. The FAR and FRR are defined as:

$$FAR(\tau) = \int_{-\infty}^{\tau} f(x | \mu, \sigma) dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\tau} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \tag{8}$$

$$FRR(\tau) = 1 - \int_{-\infty}^{\tau} f(x | \mu, \sigma) dx = 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\tau} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \tag{9}$$

where τ is the similarity threshold, x is BER, and μ and σ are the mean value and standard deviation of the BER, respectively.

Table 1 compares the BER obtained from our proposed algorithm with those from the algorithms of Wang et al. [19], Zhao and He [23], and He and Zhao [24] under different thresholds.

Table 1. Comparison of FAR values.

τ	Proposed algorithm	Wang et al. [19]	Zhao and He [23]	He and Zhao [24]
0.02	2.1743×10^{-56}	1.8849×10^{-29}	4.2686×10^{-26}	8.4455×10^{-42}
0.04	6.0859×10^{-52}	4.2150×10^{-27}	4.2502×10^{-24}	1.6026×10^{-38}
0.06	1.1039×10^{-47}	7.5793×10^{-25}	3.4827×10^{-22}	2.2062×10^{-35}
0.08	1.2978×10^{-43}	1.0728×10^{-22}	2.3491×10^{-20}	2.2036×10^{-32}
0.10	0.8909×10^{-40}	1.1957×10^{-20}	1.3044×10^{-18}	1.5974×10^{-29}
0.12	4.8877×10^{-36}	1.0459×10^{-18}	5.9646×10^{-17}	8.4047×10^{-27}
0.14	1.5665×10^{-32}	7.2567×10^{-17}	2.2464×10^{-15}	3.2108×10^{-24}
0.16	3.2571×10^{-29}	3.9541×10^{-15}	6.9702×10^{-14}	8.9086×10^{-22}

The BER of our proposed algorithm is much smaller than those of the algorithms in [19], [23], and [24] for each threshold τ . For example, when the threshold $\tau = 0.16$, the BER of our algorithm is 3.2571×10^{-29} , which means that the number of incorrect judgments per 10^{29} speech clips is only 3.3. We thus conclude that our algorithm offers strong collision resistance, uniqueness, and discrimination.

At the same time, we combined DWT, MFCC, and DCT with MM to test its FAR. Obviously, compared with the proposed algorithm (see Table 2), the FAR of the hash sequence generated by these three methods combined with MM is higher than that of the proposed algorithm. Therefore, the method of selecting the IFFT combined with the MM to generate the hash sequence has excellent performance.

Table 2. Comparison of FAR values for different methods.

τ	Proposed algorithm	DWT+MM	MFCC+MM	DCT+MM
0.02	2.1743×10^{-56}	1.2201×10^{-47}	1.4020×10^{-27}	1.2653×10^{-42}
0.04	6.0859×10^{-52}	6.4142×10^{-44}	1.8830×10^{-25}	2.5976×10^{-39}
0.06	1.1039×10^{-47}	2.3484×10^{-40}	2.0529×10^{-23}	3.8683×10^{-36}
0.08	1.2978×10^{-43}	5.9933×10^{-37}	1.8172×10^{-21}	4.1797×10^{-33}
0.10	0.8909×10^{-40}	1.0664×10^{-33}	1.3601×10^{-19}	3.2773×10^{-30}
0.12	4.8877×10^{-36}	1.3231×10^{-30}	7.6254×10^{-18}	1.8651×10^{-27}
0.14	1.5665×10^{-32}	1.1451×10^{-27}	3.6167×10^{-16}	7.7065×10^{-25}
0.16	3.2571×10^{-29}	6.9141×10^{-25}	1.3940×10^{-14}	2.3124×10^{-22}

For robustness analysis, in order to test the robustness of the proposed algorithm, we applied the content-preserving operations (CPOs) shown in Table 3 to the speech files in the library, producing 6400 speech files from 1280 speech files in our sample.

We then extracted the perceptual hashing sequences from the resulting speech files and calculated the BER for our algorithm and the others as shown in Table 4.

These results indicate that the BER values of the same perceived content were all less than 0.1801 for our algorithm, which is much smaller than the minimum of discrimination. In contrast, the results for the

Table 3. CPO and corresponding parameters.

CPO/Operating means	Operating mode	Abbreviation
Requantizing	8–16 kbps	R
Amplitude increase	3 dB for amplitude increase	$A \uparrow$
Amplitude decrease	3 dB for amplitude decrease	$A \downarrow$
MP3 compression	128 kbps	M
Noise addition	SNR = 50 dB	N1
Noise addition	SNR = 30 dB	N2
FIR filter	2 kHz FIR filter	Ff
Butterworth filter	2 kHz Butterworth filter	Bf
Echo addition	attenuation 50%	E
Flipping	Speech signal flipping	F

Table 4. BER mean of the speech CPO.

CPO	Proposed algorithm	Wang et al. [19]	Zhao and He [23]	He and Zhao [24]
R	0.0039	0.1354	0.0959	0.0026
$A \uparrow$	0.0052	0.0183	4.9938×10^{-4}	0.0039
$A \downarrow$	0.0015	0.0018	4.5803×10^{-4}	0.0042
M	0.2028	0.0177	0.0028	0.0016
N1	0.0032	0.0833	0.0672	-
N2	0.0424	0.2719	0.1028	-
Ff	0.1631	0.1471	0.2125	-
Bf	0.1801	0.2004	0.2477	-
E	0.1467	0.1454	0.1293	-
F	0	0	0	0

algorithms in [19,23,24] were much larger, especially with requantization applied in [19,23], amplitude decrease applied in [24], and noise and requantization applied in [19,23]. Thus, the CPOs exerted greater effects on the speech features using the other algorithms.

We make special note of the results for MP3 compression. After applying MP3 compression, the speech is longer than the original form. The MP3 format increases the compression ratio for the high frequency portion of the speech and decreases the compression ratio for the low frequency portion to ensure that the signal is not distorted. This greatly affects the speech feature and resulted in a slightly higher mean BER for our algorithm. Therefore, the speech robustness of our algorithm is weaker than the other algorithms when the samples are MP3 compressed. However, the basic algorithm requirements have still been met. We conclude that our algorithm is robust in the presence of CPOs.

In order to further test the performance of our method, according to all of the CPOs in Table 3, all FAR values and FRR values are obtained, and the FAR-FRR curves of our method are shown in Figure 7.

Obviously, the FAR-FRR curve of our method does not cross in the figure, which shows that the algorithm has good discrimination and robustness and can accurately identify the CPOs and different speech content. When the threshold is chosen as 0.2833–0.3500, the FAR value and FRR value are simultaneously small enough.

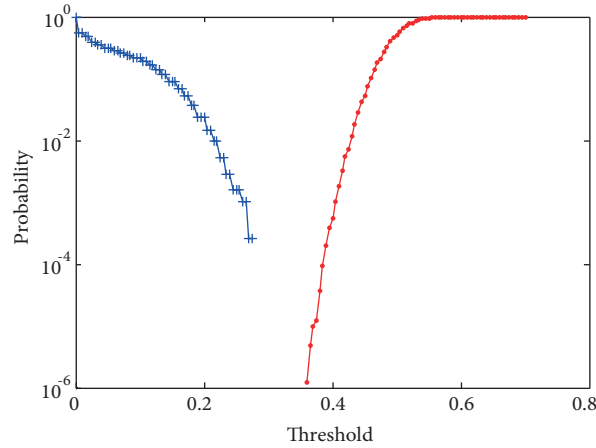


Figure 7. The FAR-FRR curves of our method.

In addition, FAR and FRR can not entirely evaluate the speech hash sequence performance. We introduce entropy rate (ER) [30] for further analysis of the performance of the hash sequence, which is the comprehensive evaluation of the algorithm discrimination and abstract. The range of ER is [0–1], and the larger the ER, the better the performance of the hash sequence. The calculation for ER is:

$$ER = q \log_2 q + (1 - q) \log_2 (1 - q), \tag{10}$$

where $q = \frac{1}{2} \sqrt{\frac{|\sigma^2 - \sigma_0^2|}{\sigma^2 + \sigma_0^2}}$, and σ is the variance of BER. We randomly selected 1000 speech clips to test ER, and experimental results show that the value of ER is 0.9986, very close to 1, which means that the performance of the algorithm hash sequence is good.

5.2.3. Performance analysis of retrieval method

We evaluated the retrieval performance of our proposed method and other algorithms [19,23] in 10,000 speech clips and another algorithms [24] in 5000 speech clips according to the common recall and precision ratios [23]. The calculations for the recall ratio R and the precision ratio P are:

$$R = \frac{f_T}{f_T + f_L} \times 100\%, \tag{11}$$

$$P = \frac{f_T}{f_T + f_F} \times 100\%, \tag{12}$$

where f_T is the number of speech clips related to the keyword in retrieval results, f_F is the number not related to the keyword, and f_L is the number related to the keyword but not retrieved.

We also used mean average precision (MAP) [31] to evaluate performance. The average accuracy of a single topic is the average of the accuracy of each relevant piece of information retrieved. The average accuracy of the primary set is the mean of the accuracy means for each topic. MAP is a single-valued indicator reflecting the performance of the system on all relevant information. The greater the correlation between the related information retrieved by the system and the query speech, the higher the MAP will be. MAP is 0 if the system

does not return any relevant information. MAP is calculated as:

$$MAP = \bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}, \tag{13}$$

where $P(r)$ is the average precision for recall ratio r , $P_i(r)$ is the precision of the i th query for precision ratio r , and N_q is the number of related documents.

When the perceptual hashing sequence of the query speech matches the features in the hash feature library sample, the similarity threshold T_2 is such that $0 < T_2 < 0.5$. If the normalized Hamming distance is $D(\mathbf{h}_1, \mathbf{h}_2) < T_2$, the match is successful. It is crucial to choose an appropriate similarity threshold for the recall and precision ratios. Our experimental results show that the minimum and maximum BER of the perceptual hashing sequence generated with our method ranged between 0.2028 and 0.3500. The appropriate similarity threshold T_2 is in that range, so we chose $T_2 = 0.25$.

In order to test whether the system can retrieve a certain speech clip accurately, we selected the 5000th speech clip as the target retrieval speech, extracted the features of this clip, and generated the hash sequence. We then used the hash sequence to match against the system hash index table and computed the BER between the query speech hash index and the index in the system hash index table. If the BER value was less than the preset threshold τ , we determined that the original speech corresponding to this digest was a retrieval result, with the matching results shown in Figure 8. Only when the BER is very small could we obtain an accurate perceptual hashing sequence to retrieve the corresponding speech, and the remaining 9999 matches failed.

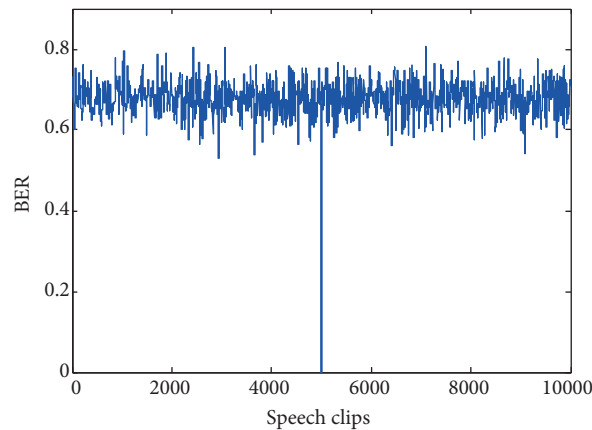


Figure 8. Matching result of query digest in system hash index table.

Different CPOs affect the speech data to varying degrees by causing changes in speech features. If the changes exceed a certain range, some speech clips cannot be distinguished, resulting in a lower recall ratio.

Tables 5–7 compare the recall ratio, precision ratio, and retrieval accuracy of our algorithm versus the others after various CPO operations. Our proposed algorithm still achieved high recall and precision ratios after the various CPOs. Moreover, the overall MAP value of our algorithm was higher than that of the others, indicating that the data returned by the retrieval engine ranked first by query similarity and had good retrieval accuracy.

The algorithms in [19,23,24] showed notable decreases in retrieval performance, especially in the presence of requantizing and MP3 compression processing. The reason is that there are certain defects in the feature

Table 5. Comparison of the recall ratio after CPO.

CPO	The recall ratio			
	Proposed algorithm	Wang et al. [19]	Zhao and He [23]	He and Zhao [24]
R	100%	95%	96%	98%
A↑	100%	96%	100%	96%
A↓	100%	97%	98%	95%
M	100%	92%	98%	96%
F	100%	100%	100%	100%

Table 6. Comparison of the precision ratio after CPO.

CPO	The precision ratio			
	Proposed algorithm	Wang et al. [19]	Zhao and He [23]	He and Zhao [24]
R	100%	92%	93%	97%
A↑	100%	93%	100%	95%
A↓	100%	96%	100%	95%
M	97%	92%	96%	96%
F	100%	100%	100%	100%

Table 7. Comparison of MAP after CPO.

CPO	The precision ratio			
	Proposed algorithm	Wang et al. [19]	Zhao and He [23]	He and Zhao [24]
R	100%	96.83%	98.58%	99.50%
A↑	100%	98.56%	100%	99.17%
A↓	100%	97.98%	100%	99.29%
M	98.47%	96.71%	98.21%	99.43%
F	100%	100%	100%	100%

extraction of components of the algorithms in [19,23,24], which offer weak discrimination of speech features and high FAR. Thus, they cannot distinguish some speech features during the retrieval process. At the same time, the algorithms in [19,23] are not very robust in determining speech features after CPO processing is applied. Such weaknesses directly affect retrieval performance and lead to greater numbers of errors and misses.

The length of the hash digest also had a significant impact on the retrieval performance. We tested 10,000 speech clips with various lengths as shown in Table 8. As the hash digest length increased, so did the retrieval accuracy, but with a decrease in retrieval efficiency. We determined the optimal hash digest length for our algorithm to be $L = 380$ after a comprehensive evaluation of algorithm efficiency and speech feature performance. For Wang et al. [19], $L = 1024$ was optimal, and for Zhao and He [23] and He and Zhao [24], $L = 256$. The length $L = 1024$ for Wang et al. [19] is long, which directly (and negatively) affected the efficiency of hash sequence generation and matching retrieval. The shorter length $L = 256$ in [23] and [24] led to poor performance of the speech feature with a concomitant decrease in retrieval performance. In summary, our algorithm's approach leads to a hash digest that is reasonably small while still offering good performance as measured by the recall ratio, precision ratio, and retrieval accuracy.

Table 8. Influence of hash digest length on retrieval accuracy and efficiency.

Retrieval performance	Length of hash digest						
	100	200	300	380	500	800	1000
Retrieval accuracy (%)	83.24	92.03	96.49	98.75	99.91	100	100
Matching time (s)	0.0088	0.0172	0.0247	0.0326	0.0439	0.0656	0.0713

Retrieval efficiency is also an important performance indicator. In order to measure the complexity and computational efficiency of our algorithm, we selected 10,000 clips from the speech library for testing. We computed the average efficiency of the retrieval algorithm, including the feature extraction and matching retrieval times, and compared them with the algorithms in [19,23,24]. The results are shown in Table 9.

Table 9. Efficiency comparison of different retrieval algorithms.

Algorithm	Frequency (GHz)	Speech length (s)	Average running time (s)
Wang et al. [19]	1.60	4	0.2613
Zhao and He [23]	3.20	4	3.7937
He and Zhao [24]	3.20	4	4.2032
Proposed algorithm	2.50	4	0.0649

Table 9 shows that the efficiency of our proposed algorithm is higher than that of the algorithms in [19,23,24], offering efficiency that is 4, 58, and 65 times greater than the algorithms in [19,23,24], respectively. Our algorithm calculates the IFFT on the original speech file and uses partial Hadamard MM to reduce data, which increases efficiency. Further, we classify the speech data and reduce the size by applying run-length compression, improving efficiency still further. In contrast, the other algorithms simply extract features of the speech sample points and directly match and retrieve according to the constructed hash, resulting in low retrieval efficiency. Thus, our algorithm achieves high retrieval efficiency as well.

6. Conclusions and future work

In this paper, we have proposed an efficient content-based encrypted retrieval algorithm. We have combined an IFFT with MM to implement a speech perceptual hashing scheme for matching speech content with query speech. The robustness, discrimination, and feature extraction efficiency of our approach are better than existing methods. We improve the indexing efficiency and required storage space of the system hash index table by classifying the speech data according to the characteristics of the speech hash sequence and applying run-length compression to compress hash sequences in the hash index. We improve the security and efficiency of speech encryption with less complexity by adopting a Henon scrambling encryption and decryption technique. On retrieval, we use a normalized Hamming distance to perform exact match results. Our experimental results show that the proposed algorithm has better retrieval efficiency and accuracy, as well as high recall and precision ratios. In addition, the encryption of the speech database does not require embedding a hash index as a digital watermark in the encrypted speech, which further improves retrieval efficiency.

As future work, we plan to improve the robustness of our perceptual hashing scheme in the presence of MP3 compression to achieve more accurate speech classification and to implement fuzzy retrieval of long speech samples.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61862041, 61363078) and the Research Project in Universities of Education Department of Gansu Province (2017B-16, 2018A-187). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

- [1] Thangavel M, Varalakshmi P, Renganayaki S, Subhupriya GR, Preethi T, Banu AZ. SMCSRC-Secure multimedia content storage and retrieval in cloud. In: IEEE 2016 Recent Trends in Information Technology; 8–9 April 2016; Chennai, India. New York, NY, USA: IEEE. pp. 1-6.
- [2] Vavrek J, Vizslay P, Lojka M, Juhar J, Pleva M. Weighted fast sequential DTW for multilingual audio Query-by-Example retrieval. *J Intell Inf Syst* 2018; 2018: 1-17.
- [3] Xiao X, Wang JQ. Improved lattice-based speech keyword spotting algorithm. *Journal of Tsinghua University* 2015; 55: 508-513 (in Chinese).
- [4] Zhao W. A high efficient music retrieval algorithm based on content. In: IEEE 2016 Measuring Technology and Mechatronics Automation; 11–12 March 2016; Macau, China. New York, NY, USA: IEEE. pp. 12-15.
- [5] Dorfer M, Arzt A, Widmer G. Towards end-to-end audio-sheet-music retrieval. arXiv preprint, arXiv:1612.05070, 2016.
- [6] Qin J, Liu X, Lin H. Audio retrieval based on manifold ranking and relevance feedback. *Tsinghua Sci Technol* 2015; 20: 613-619.
- [7] Lotia P, Khan DM. Significance of complementary spectral features for speaker recognition. *International Journal of Research in Computer and Communication Technology* 2013; 2: 579-588.
- [8] Li JF, Wu T, Wang HX. Perceptual hashing based on correlation coefficient of MFCC for speech authentication. *Journal of BUPT* 2015; 38: 89-93 (in Chinese).
- [9] Zhang QY, Xing PF, Huang YB, Dong RH, Yang ZP. Perceptual hashing algorithm for multi-format. *Journal of BUPT* 2016; 39: 77-82 (in Chinese).
- [10] Chen N, Xiao HD, Zhu J. Robust audio fingerprinting based on GammaChirp frequency cepstral coefficients and chroma. *Electron Lett* 2014; 50: 241-242.
- [11] Zhang XZ, Wang YS, Zeng Z, Niu B. An efficient filtering-and-refining retrieval method for big audio data. *Journal of Computer Research and Development* 2015; 52: 2025-2032 (in Chinese).
- [12] Coover B, Han J. A power mask based audio fingerprint. In: IEEE 2014 Acoustics, Speech and Signal Processing; 4–9 May 2014; Florence, Italy. New York, NY, USA: IEEE. pp. 1394-1398.
- [13] Stanko T, Chen B, Skoric B. Fingerprint template protection using minutia-pair spectral representations. arXiv preprint, arXiv:1804.01744, 2018.
- [14] Patel VM, Ratha NK, Chellappa R. Cancelable biometrics: a review. *IEEE Signal Proc Mag* 2015; 32: 54-65.
- [15] Kaur H, Khanna P. Random distance method for generating unimodal and multimodal cancelable biometric features. *IEEE T Inf Foren Sec* 2019; 14: 709-719.
- [16] Topcu B, Karabat C, Azadmanesh M, Erdogan H. Practical security and privacy attacks against biometric hashing using sparse recovery. *EURASIP J Adv Signal Proc* 2016; 1: 100-120.
- [17] Topcu B, Karabat C, Erdogan H. Unpredictability assessment of biometric hashing under naive and advanced threat conditions. In: IEEE 2016 Signal Processing Conference; 2016; Florence, Italy. New York, NY, USA: IEEE. pp. 2265-2269.

- [18] Hine GE, Maiorana E, Campisi P. A zero-leakage fuzzy embedder from the theoretical formulation to real data. *IEEE T Inf Foren Sec* 2017; 12: 1724-1734.
- [19] Wang H, Zhou L, Zhang W, Liu S. Watermarking-based perceptual hashing search over encrypted speech. In: *Springer 2013 International Workshop on Digital Watermarking*; 1–4 October 2013; Auckland, New Zealand. Berlin, Heidelberg: Springer. pp. 423-434.
- [20] Ibrahim A, Jin H, Yassin AA, Zou D. Secure rank-ordered search of multi-keyword trapdoor over encrypted cloud data. In: *IEEE 2012 Asia-Pacific Services Computing Conference*; 6–8 December 2012; Guilin, China. New York, NY, USA: IEEE. pp. 263-270.
- [21] Wang HX, Hao GY. Perceptual speech hashing algorithm based on time and frequency domain change characteristics. *China Patent No. 2015102405844*, 2015.
- [22] Lin L. Study on retrieval for encrypted speech and recovery watermarking-based speech authentication. MSc, Southwest Jiaotong University, Chengdu, China, 2015 (in Chinese).
- [23] Zhao H, He S. A retrieval algorithm for encrypted speech based on perceptual hashing. In: *IEEE 2016 Natural Computation, Fuzzy Systems and Knowledge Discovery*; 13–15 August 2016; Changsha, China. New York, NY, USA: IEEE. pp. 1840-1845.
- [24] He SF, Zhao H. A retrieval algorithm of encrypted speech based on syllable-level perceptual hashing. *Comput Sci Inf Syst* 2017; 14: 703-718.
- [25] Glackin C, Chollet G, Dugan N, Cannings N, Wall J, Tahir S, Rajarajan M. Privacy preserving encrypted phonetic search of speech data. In: *IEEE 2017 International Conference on Acoustics, Speech and Signal Processing*; 5–9 March 2017; New Orleans, LA, USA. New York, NY, USA: IEEE. pp. 6414-6418.
- [26] Wang Y, Wang Y, Shi Q. Optimized signal distortion for PAPR reduction of OFDM signals with IFFT/FFT complexity via ADMM approaches. *IEEE T Signal Proc* 2019; 67: 399-414.
- [27] Huang DM, Geng X, Wei LF, Su C. A secure query scheme on encrypted remote sensing images based on Henon mapping. *Journal of Software* 2017; 27: 1729-1740 (in Chinese).
- [28] Wang XW, Cui GW, Wang L, Jia XL, Nie W. Construction of measurement matrix in compressed sensing based on balanced Gold sequence. *Chinese Journal of Scientific Instruments* 2014; 35: 97-102 (in Chinese).
- [29] Zhang WY, Wei ZW, Wang BH, Xiao PH. Measuring mixing patterns in complex networks by Spearman rank correlation coefficient. *Physica A* 2016; 451: 440-450.
- [30] Zhang QY, Hu WJ, Qiao SB. Speech perceptual hashing authentication algorithm based on spectral subtraction and energy to entropy ratio. *International Journal of Network Security* 2017; 19: 752-760.
- [31] Li K, Huang Z, Cheng YC, Lee CH. A maximal figure-of-merit learning approach to maximizing mean average precision with deep neural network based classifiers. In: *IEEE 2014 International Conference on Acoustics, Speech and Signal Processing*; 4–9 May 2014; Florence, Italy. New York, NY, USA: IEEE. pp. 4503-4507.