Multimedia Tools and Applications (2024) 83:25675–25693
https://doi.org/10.1007/s11042-023-16493-5

An intelligent homomorphic audio signal encryption algorithm for secure interacting

Yingjie Hu[1] · **Qiuyu Zhang**[1] · **Qiwen Zhang**[1] · **Yujiao Ba**[1]

Received: 10 May 2022 / Revised: 21 July 2023 / Accepted: 8 August 2023 /
Published online: 22 August 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
The present encryption algorithm has high computational complexity for the large data volume as audio and usually focuses on one audio format. Furthermore, the data in the cloud cannot be modified before decrypting, which means the interaction between the encrypted audio and the users is difficult. To address the above problems, a new homomorphic audio signal encryption algorithm has been developed for secure processing and interaction in the cloud. The original audio signal is encrypted without converting to binary to reduce the computational complexity. Adaptive parameters were generated to deal with varied audio formats and attributes. The users can choose the appropriate encryption level to balance the security and complexity. Meanwhile, this scheme supports additive and multiplicative homomorphism to perform some operations (e.g., volume adjustment and audio editing) correctly with the encrypted data. Security analysis and experiments show that the proposed algorithm has better encryption performance, stronger sensitivity to key, higher efficiency, less data expansion, and could be against attacks from statistical analysis. Overall, the proposed algorithm is secure and efficient to satisfy requirements for interaction with encrypted audio files in the cloud.

Keywords Audio encryption · Homomorphic encryption · Decimal integers · Adaptive encryption · Ciphertext calculation

1 Introduction

With the development of cloud technology and mobile, large amounts of audio files have been stored in the cloud, such as voice messages, meeting records, etc. The privacy of audio is becoming more and more concern. Unlike text files, audio files usually have large sizes, which makes high efficiency become the focus of encryption technology [7]. Classical encryption algorithm takes a long time to encrypt or decrypt audio files and is difficult to practical [20], such as Data Encryption Standard (DES), Advanced

✉ Qiuyu Zhang
zhangqylz@163.com

[1] School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu, China

Encryption Standard (AES), RSA, etc. Chaotic systems [4, 8] gain popularity in the efficient encryption field, for their sensitivity to initial conditions and parameters, topological transitivity, ergodicity, etc. To improve security, several chaotic maps are combined as a composite chaotic system. For the same reason, the hyper chaotic system with more than one Lyapunov exponent is published. Although the chaotic system has high efficiency and good encryption quality, it is hard to perform interaction with the audio files in the cloud safely and correctly. Any operations could not be executed correctly with the encrypted data, but the decryption may leak the information to the cloud. The users have to modify audio files after downloading and decrypting for safety, and it will increase the costs of network transmission.

For that reason, homomorphic encryption technology is on the rise in recent years, it can keep cloud service providers from gaining sensitive information [14]. If using homomorphic encryption, the decrypted result of calculating encrypted data would equal the value of calculating the original data. Some cloud computing can be done without any privacy problems, so it has a wide range of applications for cloud computing, blockchain, etc. [10]. But the present homomorphic encryption has a large data extension and high computational complexity. It's difficult to apply to audio, so most researches mainly focus on the core data such as audio features [27]. However, the user often wants to interact with their audio files stored in the cloud, such as playback, adding or deleting parts of audio, volume adjustment, etc. Therefore, it is very useful to design a homomorphic audio encryption algorithm with small data expansion and high efficiency for practical application. In addition, the audio files are in a variety of formats, with different attributes such as the number of channels and maximum data size of each sample. Self-adaption is required to deal with that encryption and interactive operations.

The contributions to this paper are as follows:

1). We proposed a new homomorphic audio encryption scheme for less complexity and data expansion, and it supports both additive and multiplicative homomorphism with conditions.
2). We designed an intelligent encryption algorithm to adapt to various common audio formats. The proper encryption parameters are generated automatically, and the encrypted data could be reconverted to an audio file for kinds of interactions.
3). We gave a solution to secure interaction with encrypted audio, some operations can be done without decryption.

The rest of this paper is laid out as follows. Section 2 discusses the research works related to audio encryption and homomorphic encryption. Section 3 describes the proposed homomorphic encryption scheme, the intelligent encryption algorithm, and the method of interactive operation with encrypted audio. Section 4 gives details of the experimental analysis. Section 5 concludes this paper and future prospects.

2 Related work

In [18], a modified Generalized Feistel Network (non-chaotic) and a generalized modified tent map (chaotic) were used to encrypt signals separately. Both methods have similar encryption quality and time consumption, but the chaotic algorithm decrypted much

more quickly. To improve efficiency, Kalpana et al. [11] globally synchronized two chaotic systems (encryption and decryption). Due to that low-dimensional chaotic systems could be predicted in the short term, multiple chaotic systems were proposed to improve security. As in [12], four types of chaotic maps combined with SHA-1 were used to encrypt the signal. Then hyper chaotic systems have been used for higher initial value sensitivity and larger key space. Farsana et al. [6] have introduced a non-orthogonal quantum state with an improved 4-D chaotic system to gain higher security. Likewise, multiple hyper chaotic systems have been presented. Sathiyamurthi et al. [16] have permuted the real and imaginary values of the signal with that of a reference sample by a 3D Lorenz-Logistic map. Then they present a hybrid-hyper chaotic system to further improve the security level [17].

Even though the encrypted audio signal can be reconverted with audio format utilizing chaotic systems, it is hard to perform calculations on them without decryption. While the homomorphic encryption algorithm can achieve that [1]. The classical encryption algorithm RSA supports multiplicative homomorphism, but it is not suitable for handling large amounts of data. The El-Gamal encryption algorithm also supports multiplicative homomorphism, which some researchers have implemented for audio signal encryption and improved efficiency [9, 13]. Paillier homomorphic cryptosystem can directly encrypt decimal integers, unlike RSA, which makes it more efficient and supports additive homomorphism [25]. Many researchers have developed audio signal encryption based on probability and matrix, the proposed homomorphic cryptosystem [19] has lower computational complexity and less data expansion for running on smartphones. However, it still took more than 20 seconds to encrypt an audio about 5 seconds, which is not suitable for long-playing audio. Furthermore, it has many limits in audio applications for only supporting additive homomorphism. Dijk and Gentry et al. [3] proposed a fully homomorphic encryption on the integers, it could calculate multiplication and addition infinitely in theory by bootstrapping. But this scheme must convert each integer to binary to encrypt, the high complexity and huge data expansion make it stay in theory. Later, the BGV scheme was proposed to improve the old bootstrapping by using a modulus-switching technique, which has lower complexity to be more practical. Multithreading technology has been used to improve the efficiency of the DGHV fully homomorphic encryption scheme [26]. In [28], the VGG-Vox outputs were encrypted by the BFV scheme (TFHE and SEAL library), then were employed in the speaker identification system directly. Thaine et al. [21] also applied BFV (PALISADE library) to encrypt speech signals and extracted MFCCs and BFCCs features from them for speech recognition. The CKKS encryption scheme could process real numbers by scaling the fractional part to the nearest integer, which is used in deep learning. As in [2], the spectrograms of speeches were encrypted by CKKS (SEAL library), then were used for a convolutional neural network to identify speakers. Although the new homomorphic encryption scheme has improved performances and reduced data expansion by utilizing batch processing and other methods, it is not suitable for audio files yet. Therefore, the present researches are mainly on the encryption of core data, such as audio features, except for the work in [21]. Meanwhile, a lightweight solution is required to process encrypted audio stored in the cloud, and there is no such research in this area before, as far as we have known.

In summary, chaotic encryption is more efficient, but its practical applications are limited. Homomorphic encryption algorithm allows some calculations on encrypted data, but the poor efficiency and huge data expansion increase storage spaces and transmission times. Especially in the case of mass data onto long-playing audio, how to solve these problems becomes a difficulty. Besides that, encryption and calculation require different

parameters for different audio formats and property values. This problem has been less considered in related research so far. Given that, this paper proposed intelligent audio homomorphic encryption for secure interaction. It could intelligently encrypt and decrypt according to different formats and attributes to good efficiency and less data extension, and keep interactive operations in security.

3 Proposed algorithm

3.1 Homomorphic encryption scheme based on decimal integers

Our scheme was inspired by a binary symmetric encryption scheme in [3]. It is a fully homomorphic encryption by bootstrapping and could be an asymmetric cryptosystem by generating a public key set. Its security relies on the approximate GCD problem. But its data expansion is huge, the size of the encrypted audio file would be unacceptable.

The proposed scheme η has three key parts:

KeyGen: J, K are keys, chosen randomly from positive integers.

Encrypt (J, K, m): Set $c = m + \mathrm{J} \times \mathrm{d} + \mathrm{K} \times \mathrm{a} \times \mathrm{b}$, where message m is a decimal integer, $m \in \mathbf{N}$, $c \in \mathbf{N}$. a, b, d are random positive integers.

Decrypt (J, K, c): (c mod J) mod K.

Theorem 1 *The above scheme η is correct, if* $K > m$, *and* $K \times a \times b + m < J/2$.

Proof From the Decrypt (J, K, c),

$$\begin{aligned} & c \bmod \mathrm{J} \\ & = (m + \mathrm{J} \times \mathrm{d} + \mathrm{K} \times \mathrm{a} \times \mathrm{b}) \bmod \mathrm{J} \\ & = (m \bmod \mathrm{J} + (\mathrm{K} \times \mathrm{a} \times \mathrm{b}) \bmod \mathrm{J}) \bmod \mathrm{J} \\ & \because \mathrm{K} \times \mathrm{a} \times \mathrm{b} + m < \mathrm{J}/2 \\ & \therefore \text{Above} \\ & = (m + \mathrm{K} \times \mathrm{a} \times \mathrm{b}) \bmod \mathrm{J} \\ & = m + \mathrm{K} \times \mathrm{a} \times \mathrm{b} \end{aligned}$$

Then

$$\begin{aligned} & (c \bmod \mathrm{J}) \bmod \mathrm{K} \\ & = (m + \mathrm{K} \times \mathrm{a} \times \mathrm{b}) \bmod \mathrm{K} \\ & = (m \bmod \mathrm{K} + (\mathrm{K} \times \mathrm{a} \times \mathrm{b}) \bmod \mathrm{K}) \bmod \mathrm{K} \\ & \because \mathrm{K} > m \\ & \therefore \text{Above} \\ & = m \bmod \mathrm{K} \\ & = m \end{aligned}$$

When the keys K and J meet the given conditions, the decryption is successful.

Theorem 2 *Let* c_1 *be the result of encrypting message* m_1, $c_1 = m_1 + J \times d_1 + K \times a_1 \times b_1$, *Let* c_2 *be the result of encrypting message* m_2, $c_2 = m_2 + J \times d_2 + K \times a_2 \times b_2$. *Then the scheme η is addition homomorphic if* $m_1 + m_2 < K$ *and the keys are chosen as required by Theorem*1.

Proof

$$
\begin{aligned}
&c_1 + c_2 = \left(m_1 + m_2\right) + \mathrm{J} \times \left(\mathrm{d}_1 + \mathrm{d}_2\right) + \mathrm{K} \times \left(\mathrm{a}_1 \times \mathrm{b}_1 + \mathrm{a}_2 \times \mathrm{b}_2\right) \\
&\text{Decrypt}\left(\mathrm{J}, \mathrm{K}, c_1 + c_2\right) \\
&= \left(\left(m_1 + m_2\right) \bmod \mathrm{J} + \mathrm{K} \times \left(\mathrm{a}_1 \times \mathrm{b}_1 + \mathrm{a}_2 \times \mathrm{b}_2\right) \bmod \mathrm{J}\right) \bmod \mathrm{J} \bmod \mathrm{K} \\
&\because \mathrm{K} \times \mathrm{a} \times \mathrm{b} + m < \mathrm{J}/2 \\
&\therefore \text{Above} \\
&= \left(\left(m_1 + m_2\right) + \mathrm{K} \times \left(\mathrm{a}_1 \times \mathrm{b}_1 + \mathrm{a}_2 \times \mathrm{b}_2\right)\right) \bmod \mathrm{K} \\
&\because m_1 + m_2 < \mathrm{K} \\
&\text{Then Decrypt}\left(\mathrm{J}, \mathrm{K}, c_1 + c_2\right) \\
&= m_1 + m_2
\end{aligned}
$$

If the operand is a constant, the sum of message m and this constant must be smaller than the key K too.

Theorem 3 *Let c_1 be the result of encrypting message m_1, $c_1 = m_1 + J \times d_1 + K \times a_1 \times b_1$. Set r is an integer, if $m_1 \times r < K$ and $K \times a_1 \times b_1 \times r < J$, both keys are chosen as required by Theorem*1*. Then the decrypted value of $c_1 \times r$ is equal to $m_1 \times r$, and the scheme η is multiplication homomorphic.*

Proof

$$
\begin{aligned}
&c_1 \times r = m_1 \times r + \mathrm{J} \times \mathrm{d}_1 \times r + \mathrm{K} \times \mathrm{a}_1 \times \mathrm{b}_1 \times r \\
&\text{Decrypt}\left(\mathrm{J}, \mathrm{K}, c_1 \times r\right) \\
&= \left(\left(m_1 \times r + \mathrm{J} \times \mathrm{d}_1 \times r + \mathrm{K} \times \mathrm{a}_1 \times \mathrm{b}_1 \times r\right) \bmod \mathrm{J}\right) \bmod \mathrm{K} \\
&= \left(\left(m_1 \times r\right) \bmod \mathrm{J} + \left(\mathrm{K} \times \mathrm{a}_1 \times \mathrm{b}_1 \times r\right) \bmod \mathrm{J}\right) \bmod \mathrm{J} \bmod \mathrm{K} \\
&\because m_1 \times r < \mathrm{K}, \mathrm{K} \times \mathrm{a}_1 \times \mathrm{b}_1 \times r < \mathrm{J} \\
&\therefore \text{Above} \\
&= m_1 \times r
\end{aligned}
$$

3.2 Adaptive intelligent audio encryption algorithm

Figure 1 illustrates the main process of the proposed encryption algorithm. Adaptive cryptographic parameters are generated for supporting multiple audio formats, then encrypted data are reconverted to audio files automatically.

Definition 1 (Adaptive Cryptographic Parameters): For an audio file, the maximum value of each sample is 2^s, (usually s is 8, 16, or 32), and the encryption level is l, where $l \in \mathbf{N+}$. Then let the length of secret key K be $s+2$ bits and the length of secret key J be ($s \times 4 \times l$

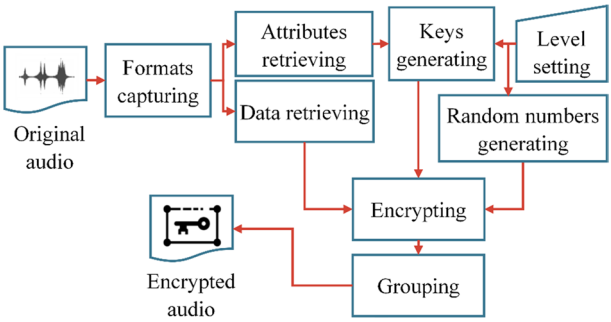


Fig. 1 Processes of the proposed audio encryption algorithm

$-2-7\times l$) bits, set random numbers a, b, and d: $1\le a, b\le 2^{4\times l}$, $1\le d\le 2^{7\times l}$. These parameters always guarantee the success of decryption.

Proof The length of K is $s+2$ bits, $2^{s+1}<K<2^{s+2}$, the maximum value of an audio sample is 2^s, and K is always bigger than m. Another condition is $K\times a\times b+m<J/2$ by Theorem1, where $K\times a\times b+m$ is no more than $2^{s+8\times l+2}$, J/2 is $2^{s\times 4\times l-3-7\times l}$. $s+8\times l+2-(s\times 4\times l-3-7\times l)=(1-4\times l)\times s+15\times l+5$. According to the standards of audio files, the minimum value of s is 8. Then $K\times a\times b+m$ always be less than J/2, even if l is 1, the minimum value of it.

From the head information on an audio file, the format can be identified, such as 'WAV', 'MP3', 'WMA', etc. Then get the related attributes: h (the number of channels) and s bits (the maximum length of an audio sample). Let the sum of the samples be n, the original audio sampling data $M=\{m_1, m_2, \cdots m_n\}$, where $-2^s\le m_i\le 2^s-1$. Keys K and J can be generated adaptively by any random number generator under Definition1.

The proposed encryption algorithm can be split into five specific steps:

Step 1: Each m_i in M is considered to be two parts, sgn_i for the plus/minus sign and um_i for the unsigned value, where $sgn_i\in\{+,-\}$, $0\le um_i\le 2^s$ and $um_i\in\mathbf{N}$. Restrict the operation of modulo to positive numbers, because different programming languages process negative values in totally different ways.

Step 2: For each m_i, generate new adaptive random numbers a_i, b_i, and d_i under Definition 1.

Step 3: The uc_i (encrypted data without sign) is computed by equation $(um_i+J\times d_i+K\times a_i\times b_i)$.

Step 4: Set $c_i\leftarrow sgn_i+uc_i$. If the length of c_i is less than $s\times 4\times l-1$, pad c_i to the left with zeros. If the length of c_i is more than 32 bits, divide c_i into several groups of 32 bits each, and the last group has 31 bits, $c_i\leftarrow\{x_1, x_2, \ldots x_j\}$, where $j=(s\times 4\times l)/32$.

Step 5: Reconvert encrypted data to WAV format to keep quality, since MP3 and other formats would compress data again. Write each x_i to the file, If x_1 is zero, x_j is written with sgn_i. The maximum length of an audio sample of the encrypted file is 32 bits, and the rest of the attributes are unchanged.

The proposed decryption algorithm can fully recover the originaldecryption algorithm audio signal without any additional record, it can be split into four steps:

Step 1: Read in an encrypted file, get h, l, s, and encrypted audio data C.

Step 2: Let $j\leftarrow(s\times 4\times l)/32$, each j sample is a group, and obtain sgn_i according to whether the first value of each group is 0 or not.

Step 3: Fill each sample with zeros from the left to make sure the size is 32 bits (31 bits for the last one). Then recombine data onto a group to get uc_i.

Step 4: Let $ud_i\leftarrow(uc_i \bmod J) \bmod K$, $d_i\leftarrow sgn_i+ud_i$. Write d_i to the decrypted file. WAV format is preferred, other formats also can be appointed.

3.3 Security interactive operation on encrypted audio

Theorem 4 According to the relationship between audio signals with volume, to increase approximately 6 dB means to double the value of each sample. The adaptive parameters satisfy the conditions of multiplicative homomorphism under audio format.

Proof. There are two conditions according to Theorem 3: $m \times 2 < K$ and $K \times a \times b \times 2 < J$. Where m is no more than 2^s, *so* $m \times 2$ would be no more than 2^{s+1}, and certainly be less than K. $K \times a \times b \times 2$ is also certainly smaller than J.

Figure 2 shows the solution to edit encrypted audio in the cloud without decryption and downloading. The proposed method can improve the efficiency and security of interaction between the user and encrypted audio. The user sends a request for volume adjustment to the server. When receiving it, the server processes the encrypted audio file to get c_i and multiply each c_i by 2 without decryption. By Theorem 4, Decrypt $(c_i \times 2) = m_i \times 2$.

Likewise, If the user wants to delete a clip (t_1 th- t_2 th seconds), then the server just removes the encrypted data from $t_1 \times r \times 2 \times l$ *to* $t_2 \times r \times 2 \times l$, where r is the number of samples per second. If the user wants to insert a new clip at t th second, the server adds the new encrypted data after the y th data, where $y = t \times r \times 2 \times l$.

Moreover, the modified data can be sent back to the user, the user can play while decrypting. Other signal processing programs about addition, subtraction, and multiplication also can be introduced.

4 Security analysis and performance analysis

For the test, we use some audio files from the THCHS30 database and the TIMIT database. All of them were sampled at 16 kHz and 256 kbps. All tests are programmed in Python 3.6 on a PC with a 2.30 GHz CPU and 4.00 GB main memory.

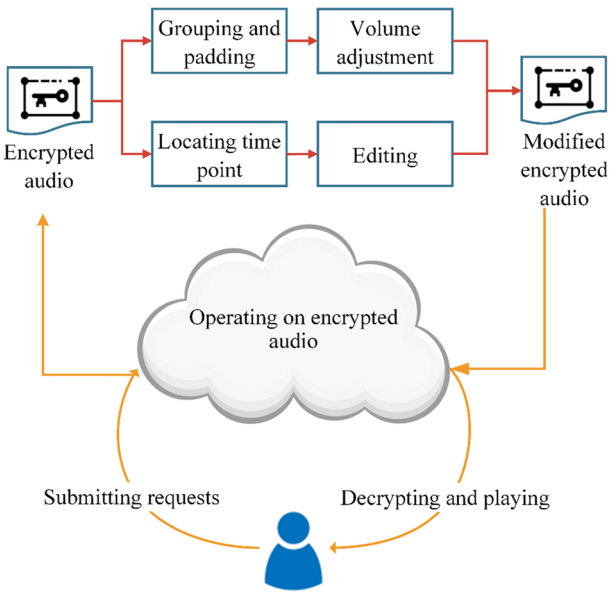
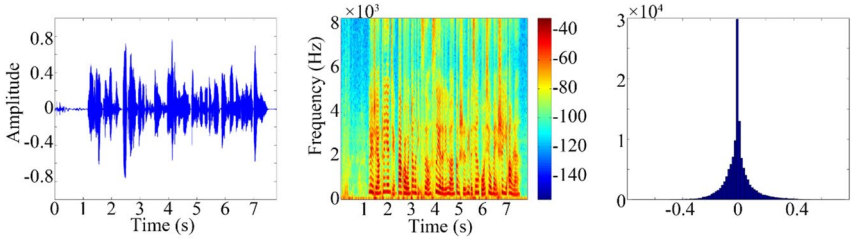
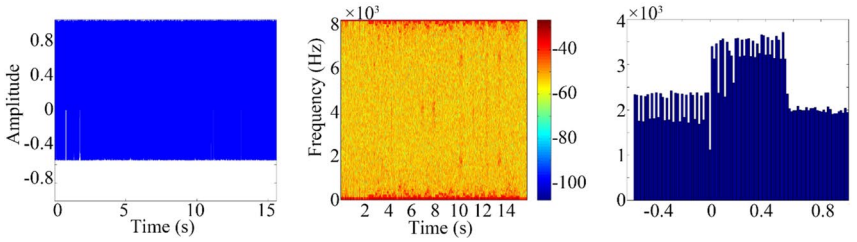


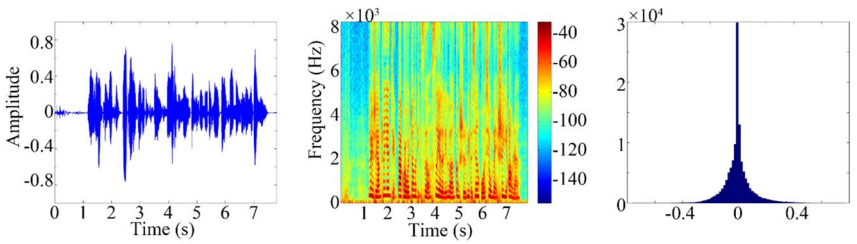
Fig. 2 Sketch map of the interactive operation


Amplitude
Time (s)
Frequency (Hz)
Time (s)

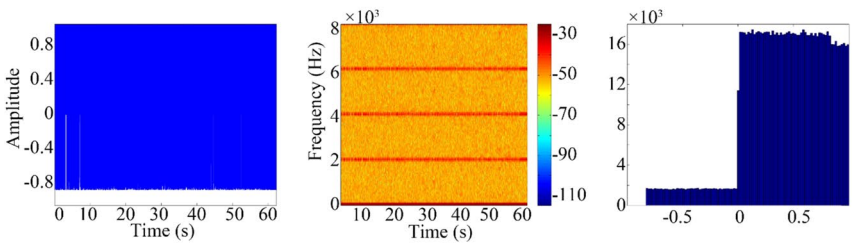
(a)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

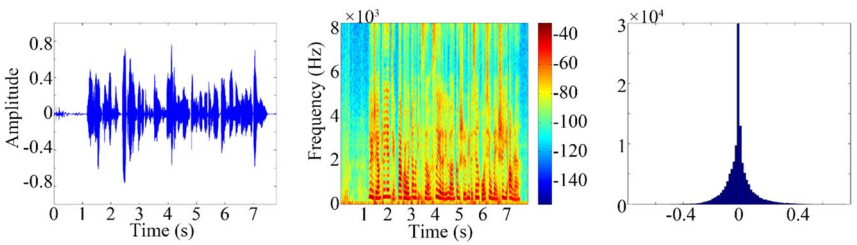
(b)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(c)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(d)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(e)

◀**Fig. 3** The waveforms, spectrograms, and histograms for the format WAV. **a** The original audio, **b** Encrypted audio ($l=1$), **c** Decrypted audio ($l=1$), **d** Encrypted audio ($l=4$), **e** Decrypted audio ($l=4$)

4.1 Statistical analysis

4.1.1 Histograms and residual intelligibility

Figures 3 and 4 show the waveforms, spectrograms, and histograms of one test file, encrypted and decrypted files in different formats and encryption levels.

As shown in Figs. 3 and 4, the waveforms and spectrograms are changed without any original features even if the encryption level is 1. For the poor residual intelligibility of encrypted audio, attackers cannot get any information about the original signal. In the histogram, the encrypted data is distributed more evenly, which makes the statistical analysis by attackers hard to succeed.

When the encryption level is up to 4, the spectrogram of the encrypted audio signal is more like noise. Since only one in a group may have the plus/minus sign when the encrypted data was reconverted, the number of positive samples of the histogram is much more than that of the negative. But the data is distributed evenly among positive and negative separately.

In addition, the waveforms and spectrograms of decrypted and original audio files are about the same, the audio signals have been fully recovered after decryption.

4.1.2 Correlation coefficients

The correlation coefficients [5] between the original and encrypted audio signal, and the original and decrypted audio signal of four test files are listed in Table 1. Note: Some of the encrypted audio signals have been truncated in the calculation to keep the same data volume of the formula.

It can be seen from Table 1 that the correlation coefficients are close to zero between the encrypted audio and the original one. Even if the attackers obtained some original data, they could not recover adjacent samples based on the correlation analysis. The correlation coefficients between the decrypted audio and the original one are nearly 1, suggesting again that the decrypted audios have the high quality as the original.

4.1.3 Entropy

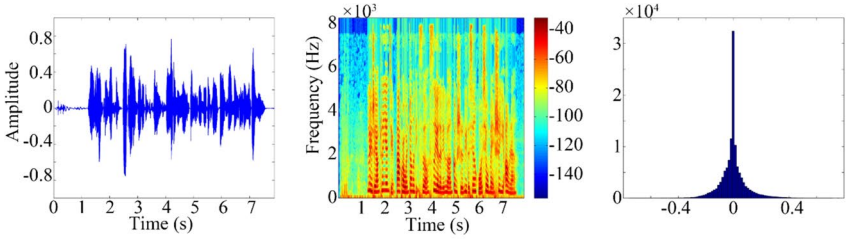
The entropy [22] of the four test audio files and the corresponding encrypted audio files are shown in Table 2.

The entropy is usually used to measure the randomness and unpredictability of the distribution of samples of audio. The entropy of encrypted audio is higher than that of the original, and it increases significantly to the bigger encryption level. The distribution of signal after encryption is more random and enough to guard against attacks of statistical analysis.

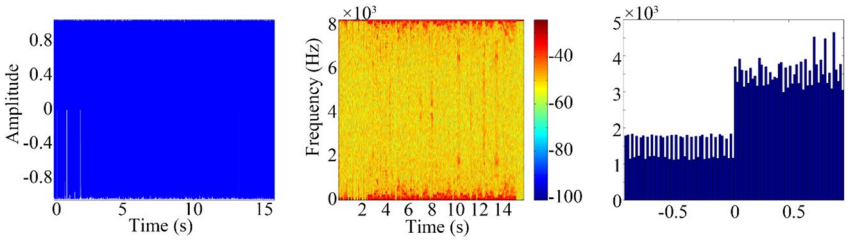
4.2 Sensitivity analysis

4.2.1 Key space and security analysis

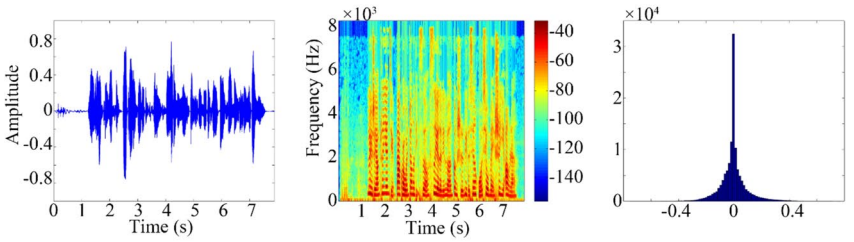
Our algorithm has two keys J and K as well as random numbers a, b, and d. The key space is a total of $2^{7+4\times l+s\times 4\times l}$. If $s=16$ and $l=4$, the key space is up to 2^{279}, it exceeds the traditional


Amplitude
Time (s)
Frequency (Hz)
Time (s)

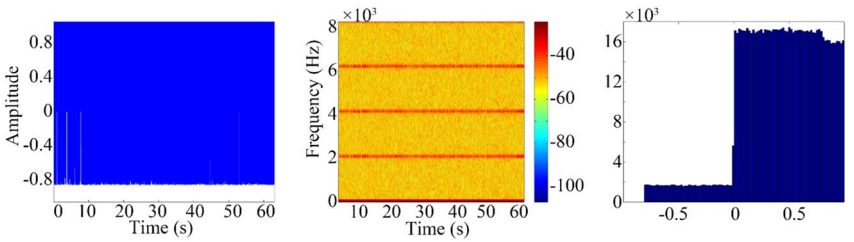
(a)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

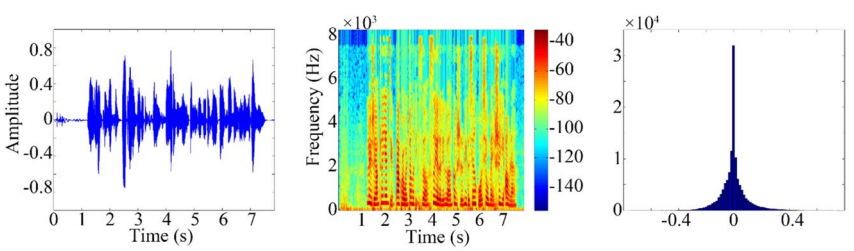
(b)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(c)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(d)


Amplitude
Time (s)
Frequency (Hz)
Time (s)

(e)

◂**Fig. 4** The waveforms, spectrograms, and histograms for the format MP3. **a** The original audio, **b** Encrypted audio ($l=1$), **c** Decrypted audio ($l=1$), **d** Encrypted audio ($l=4$), **e** Decrypted audio ($l=4$)

AES algorithms and could prevent brute force attacks. Meanwhile, the key space can be further enlarged by increasing the encryption level for the security of important audio.

In addition, the proposed algorithm can effectively prevent known/ chosen plaintext attacks. Because it involves three random numbers, the encrypting results of the same key are different from the same data.

The header information only has general properties of audio, only the length and the size of K can be obtained from it. Even if the attacker intercepts the encrypted file, it is hard to recover the original audio based on the header information without the keys. Therefore, keeping the format information unencrypted is barely harmful to the security of the scheme.

4.2.2 Key sensitivity

There are two important indexes for key sensitivity, the Number of Samples Change Rate (NSCR) and the Unified Average Changing Intensity (UACI) [23] as follows:

$$NSCR = \frac{\sum_i D(i)}{q} \times 100\%, D(i) = \left\{ \begin{array}{l} 1, \ c(i) \neq c'(i) \\ 0, \ c(i) = c'(i) \end{array} \right\} \tag{1}$$

$$UACI = \frac{1}{q} \sum_i \frac{\|c(i)| - |c'(i)\|}{2^{31}} \times 100\% \tag{2}$$

Where q is the number of samples, c(i) is the encrypted signal using the keys J and K, and c'(i) is the encrypted signal using the keys J' (change the least significant bit of J) and K. Unlike the pixel, the values of the signal may be negative, so the absolute values of signal are used instead to avoid interference in Eq. (2).

NSCR gives the number of different encrypted signals when the least significant bit of the key is changed. UACI further quantifies how much difference between two encrypted audio files. The results of four test files are given in Table 3.

According to the measurement of the image encryption, the ideal value of NSCR and UACI is 99.6094% and 33.4635%. The closer to the ideal value, the more sensitive to the

Table 1 Correlation coefficients

Test file	Encryption Level	Encrypted audio	Decrypted audio
A11_0.wav	1	−0.00468	1.00
	4	−0.00012	1.00
sa2.wav	1	−0.00772	1.00
	4	0.00286	1.00
si1824.wav	1	−0.00359	1.00
	4	−0.00744	1.00
A11_0.mp3	1	−0.00166	1.00
	4	0.00326	1.00

Table 2 Entropy

Test file	Encryption Level	Original audio	Encrypted audio
A11_0.wav	1	13.0940	13.4454
	4		19.9287
sa2.wav	1	10.8046	12.6332
	4		18.2926
si1824.wav	1	8.9803	12.9832
	4		19.0351
A11_0.mp3	1	13.0485	13.4575
	4		19.9439

key. From Table 3, the proposed encryption algorithm is sensitive to keys with strong security.

4.3 Quality of encrypted audio

To measure the encryption quality, we calculate the Signal-to-noise ratio (SNR), segmental signal-to-noise ratio (SegSNR), peak signal-to-noise ratio (PSNR) [15], and the mean squared error (MSE). Table 4 lists some of the results. In addition, many data of the MP3 files are zero, and $\log_{10}0$ would lead to an incorrect result (infinite value) in the calculation of SegSNR. To avoid this, $\log_{10}0.1$ is used instead. And some data from encrypted audio is truncated in the calculation to keep the same length as the original one.

From the results, no matter whether the encryption level is 1 or 4, the values (SNR, SegSNR, and PSNR) are all less than zero. The MSE values are to reach 10^{17}. The difference between the encrypted signal and the original is too large to detect any information, which means that the encryption quality of the proposed method is high.

4.4 Comparison with the present works

Table 5 compares the proposed method with a multiple chaotic system [12], a multiple hyper chaotic system [16], a hybrid-hyper chaotic system [17], the ElGamal encryption [9, 13], and DGHV encryption based on multithreading [26]. The results take the

Table 3 NSCR and UACI

Test file	Encryption Level	NSCR (%)	UACI (%)
A11_0.wav	1	99.6022	25.5229
	4	100.0000	32.6502
sa2.wav	1	99.6238	26.4877
	4	100.0000	31.8167
si1824.wav	1	99.6256	25.1989
	4	100.0000	32.2732
A11_0.mp3	1	99.5842	33.1839
	4	100.0000	32.5794

Table 4 SNR, SegSNR, PSNR and MSE

Test file	Encryption Level	SNR (dB)	SegSNR (dB)	PSNR (dB)	MSE (10^{17})
A11_0.wav	1	−149.08	−116.91	−92.08	10.25
	4	−150.65	−118.49	−93.65	14.74
sa2.wav	1	−148.74	−130.31	−101.24	9.98
	4	−150.24	−131.82	−102.74	14.10
si1824.wav	1	−145.72	−142.81	−110.18	9.85
	4	−147.42	−144.51	−111.87	14.56
A11_0.mp3	1	−150.72	−119.46	−93.92	15.52
	4	−150.49	−119.23	−93.69	14.72

average value of the main indexes. The test uses 6 audio files from the TIMIT database. The performance of the proposed method is superior to others, for all the values of indexes are less than zero. The absolute SNR of the proposed algorithm is the largest except that of Ref. [26], and the absolute values of PSNR and SegSNR are the largest. The correlation coefficient between the original and the encrypted of the proposed algorithm is the minimum except that of Ref. [13]. But the decrypted audio in Ref. [13] cannot maintain the same quality as the original (the correlation coefficient between the original and the decrypted is only 0.8006).

4.5 Complexity analysis

Overall, the proposed algorithm is simple and easy to implement, only one loop is needed for encrypting an audio with n samples. From the encryption function $\boldsymbol{C} = m + J \times d + K \times a \times b$, the computational complexity is O (n).

$$T(n) = \sum_{i=1}^{n} \left(m_i + J \times d + K \times a \times b \right) = \sum_{i=1}^{n} \mathrm{O}\left(m_i \right) = \mathrm{O}(n)$$

The decryption function is (c mod J) mod K, and its computational complexity is O (n) too.

Table 5 Comparison of the main evaluation metrics

	SNR(dB)	SegSNR(dB)	PSNR(dB)	Correlation coefficients (Encrypted)	Correlation coefficients (Decrypted)
Ref. [12]	50.01	–	65.62	0.0120	1.0000
Ref. [16]	122.49	122.77	50.10	0.0418	0.9925
Ref. [17]	123.89	122.83	51.33	0.0104	0.9981
Ref. [13]	–	–	–	0.0004	0.8006
Ref. [9]	−35.02	38.02	–	–	–
Ref. [26]	152.49	−118.45	–	0.0032	1.0000
Proposed	−148.12	−135.90	−102.80	−0.0020	1.0000

$$T(n) = \sum_{i=1}^{n} \left(\left(c_i \; mod J \right) mod K \right) = \mathrm{O}(n)$$

The computational complexity of encryption based on the Paillier cryptosystem [24] is over O (g^n), where g is the public key. The computational complexity of decryption is over O (n^{λ}), where λ is the private key.

The encryption $\mathbf{c} = \mathbf{P_1} \times \boldsymbol{m} \times \mathbf{P_2^{-1}}$ and decryption $\boldsymbol{m} = \mathbf{P_1^{-1}} \times \mathbf{c} \times \mathbf{P_2}$ are used in [19], where the matrix $\mathbf{P}_1$ and $\mathbf{P}_2$ are keys. The computational complexity of both is O (n^3).

We used twenty audio files from the THCHS30 database to test the time consumption of encryption and decryption. When l (encryption level) = 1, It takes an average of 1.79 s to encrypt each audio file (7–8 seconds). Encrypting 16,000 samples (about 1 s) takes only 0.2215747 s. When $l = 4$ it takes an average of 2.51 s for each file and 0.3112282 s for 16,000 samples. The algorithm used in [19] and the Paillier cryptosystem takes 50s around to encrypt an 8-second audio, about 6.2348027 s for 16,000 samples. And the time consumption non-linearly increases with the number of data. In decryption, when $l = 1$ ($l = 4$), our algorithm takes an average of 1.34 s (2.85 s) for each audio file. The algorithm used in [19] and the Paillier cryptosystem takes 40s around to decrypt an audio (about 8 seconds).

Table 6 lists the comparison results of time consumption, which show that the proposed method encrypts and decrypts much more quickly than most other Homomorphic encryption, and it could process large data volumes in real-time for applications in the cloud. Although the time consumption of the ElGamal encryption [13] is less, it only supports multiplicative homomorphism.

In terms of data expansion, the proposed algorithm has the best performance. When $l = 4$, the data volume raises only 16 times after encryption. In [19], they take the remainder of dividing encrypted data by 2^{15} to reduce data expansion. But all of the quotients need to be saved to reconstruct the original encrypted audio.

4.6 The simulation results of interactive operation on encrypted audio

Figure 5 shows the simulation results of the proposed algorithm to amplify the volume of the encrypted audio. The decibels (dB) are a measure of amplitude, compared with Fig. 5a, the decibel values of Fig. 5b and 5c are significantly increased. Meanwhile, the volume

Table 6 Comparison of time consumption and data expansion

	Time for encryption (μs / Kb)	Time for encryption (μs / Kb)	Data expansion	Encrypted audio editing	Amplify volume of encrypted
Ref. [11]	99.6	99.9	–	✗	✗
Ref. [13] (ElGamal)	1268.9	710.4	10^4	✗	✗
Ref. [9] (ElGamal)	–	–	10^4	✗	✗
Ref. [19]	199,513.7	159,610.9	10^{10}	✗	✗
Ref. [21] (BFV)	28,433,801.0	15,645,062.2	$10^{6.6}$	✗	✗
Ref. [26] (DGHV)	55,542.4	26,310.4	10^4	✗	✗
Proposed	7090.4	5333.6	$4 \times l$	✓	✓

(a)

(b)

(c)

Fig. 5 The comparison of volume adjustment. **a** The original audio, **b** The decryption audio using the proposed security interactive operation, **c** Original audio after increasing 6 dB using an Editor

amplification can also be heard during playback. And the waveforms of Fig. 5b and 5c are almost identical, verifying the effectiveness of the proposed algorithm.

Figure 6 gives some numerical examples of performing volume adjustment, the server executes calculation on the encrypted data and cannot get information during the process. The proposed solution is secure and efficient, it takes only 0.0679390 s to multiply 2 by the whole encrypted audio (7-second original audio, 124,800 samples).

If combined with various audio filters and transformed the relevant calculations into approximate addition and multiplication, more refined adjustments could be achieved. It's beyond the scope of this paper and will not be discussed further.

Original audio [860 -155 750 781 764 ⋯]

Secret keys (J: 20926704276558077 K: 201362)

Encrypted speech data [964730849 456770391 -623664387 794187715 380045486 122675405 -916007068 2022090435 769835728 358158263 ⋯]

Encrypted data (Before reconverting) [2071743723405427543 -1339309073716631491 816141466803888333 -1967110202004514499 1653209637884334007 ⋯]

Encrypted data (Multiplied by 2) [4143487446810855086 -2678618147433262982 1632282933607776666 -3934220404009028998 3306419275768668014 ⋯]

Encrypted speech data (Modified) [1929461698 913540782 -1247328774 1588375430 760090972 245350810 -1832014137 1896697222 1539671456 716316526 ⋯]

Downloaded and decryption

Decrypted audio [1720 -310 1500 -1562 1528 ⋯]

Fig. 6 Numerical examples

Figure 7 shows the result of the proposed deleting operation on encrypted audio. The duration of the original audio is 7.80s, and the clip need to be deleted is from 2.35 s to 5.75 s.

Many operations on audio involve multiplication, such as volume adjustment, it's useless if the algorithm only supports additive homomorphism. For the BFV scheme, it is difficult to reconstruct encrypted data into audio files and hard to perform interactive operations efficiently.

(a)

(b)

Fig. 7 Screenshots of the audio editor. **a** The original audio, **b** The modified decrypted audio using the proposed security interactive operation

5 Conclusions

This paper presents intelligent homomorphic audio encryption for secure interacting. It has good performance on SNR, SegSNR, PSNR, and MSE, and can guard against attacks such as statistical analysis. Moreover, the lower data expansion and time complexity make real-time encryption and secure interaction possible. The proposed algorithm can deal with various audio formats commonly used in the original and compressed domains and can generate proper encryption parameters adaptively. In addition, the proposed scheme supports homomorphic multiplication and addition and can realize some interactive operations on encrypted audio files efficiently and safely. And the users can set the encryption levels to further balance the relationship between security and time/space consumption.

For future works, we intend to improve the time/space consumption and design an asymmetric encryption system.

Acknowledgments This work is supported by the *National Natural Science Foundation of China* (No. 61862041).

Author contributions All authors contributed to the study's conception and design. Yingjie Hu and Qiuyu Zhang wrote the main manuscript text. Qiwen Zhang and Yujiao Ba prepared Table 6. All authors read and approved the final manuscript.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

The authors have no competing interests to declare that are relevant to the content of this article.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alaya B, Laouamer L, Msilini N (2020) Homomorphic encryption systems statement: trends and challenges. Comput Sci Rev 36(100235):1–14. https://doi.org/10.1016/j.cosrev.2020.100235
2. Chindriş MC, Togan M, Arseni ŞC (2020) Secure speaker recognition system using homomorphic encryption. In: International Conference on Information Technology and Communications Security. pp 198–211. https://doi.org/10.1007/978-3-030-69255-1_13
3. Dijk MV, Gentry C, Halevi S, Vaikuntanathan V (2010) Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (eds) Advances in Cryptology–EUROCRYPT 2010. pp 24–43. https://doi.org/10.1007/978-3-642-13190-5_2
4. Elsafty AH, Tolba MF, Said LA, Madian AH, Radwan AG (2020) Enhanced hardware implementation of a mixed-order nonlinear chaotic system and speech encryption application. AEU-Int J Electron Commun 125(153347):1–13. https://doi.org/10.1016/j.aeue.2020.153347
5. Farsana FJ, Gopakumar K (2017) Private key encryption of speech signal based on three dimensional chaotic map. In: International Conference on Communication and Signal Processing (ICCSP). pp 2197–2201. https://doi.org/10.1109/ICCSP.2017.8286804
6. Farsana FJ, Gopakumar K (2020) Speech encryption algorithm based on nonorthogonal quantum state with hyperchaotic keystreams. Adv Math Phys 8050934:1–12. https://doi.org/10.1155/2020/8050934
7. Gupta BB, Yamaguchi S, Zhang Z, Psannis KE (2018) Editorial: recent advances on security and privacy of multimedia big data in the critical infrastructure. Multimed Tools Appl 77:31517–31524. https://doi.org/10.1007/s11042-018-6426-2

8. Hashemi S, Pourmina MA, Mobayen S, Alagheband MR (2021) Multiuser wireless speech encryption using synchronized chaotic systems. Int J Speech Technol 24:651–663. https://doi.org/10.1007/s10772-021-09821-3
9. Imran OA, Yousif SF, Hameed IS, Abed WNAD, Hammid AT (2020) Implementation of El-Gamal algorithm for speech signals encryption and decryption. Procedia Comput Sci 167:1028–1037. https://doi.org/10.1016/j.procs.2020.03.402
10. Joshi B, Joshi B, Mishra A, Arya V, Gupta AK, Peraković D (2022) A comparative study of privacy-preserving homomorphic encryption techniques in cloud computing. Int J Cloud Appl Comput (IJCAC) 12(1):1–11. https://doi.org/10.4018/IJCAC.309936
11. Kalpana M, Ratnavelu K, Balasubramaniam P (2019) An audio encryption based on synchronization of robust BAM FCNNs with time delays. Multimed Tools Appl 78:5969–5988. https://doi.org/10.1007/s11042-018-6373-y
12. Kaur G, Singh K, Gill HS (2021) Chaos-based joint speech encryption scheme using SHA-1. Multimed Tools Appl 80:10927–10947. https://doi.org/10.1007/s11042-020-10223-x
13. Khoirom MS, Laiphrakpam DS, Tuithung T (2021) Audio encryption using ameliorated ElGamal public key encryption over finite field. Wirel Pers Commun 117:809–823. https://doi.org/10.1007/s11277-020-07897-9
14. Mishra A (2023) Homomorphic encryption: securing sensitive data in the age of cloud computing. Insights2techinfo. https://insights2techinfo.com/homomorphic-encryption-securing-sensitive-data-in-the-age-of-cloud-computing/. Accessed 13 July 2023
15. Sathiyamurthi P, Ramakrishnan S (2017) Speech encryption using chaotic shift keying for secured speech communication. EURASIP J Audio, Speech, Music Process 20:1–11. https://doi.org/10.1186/s13636-017-0118-0
16. Sathiyamurthi P, Ramakrishnan S (2020) Speech encryption algorithm using FFT and 3D-Lorenz–logistic chaotic map. Multimed Tools Appl 79:17817–17835. https://doi.org/10.1007/s11042-020-08729-5
17. Sathiyamurthi P, Ramakrishnan S (2022) Speech encryption using hybrid-hyper chaotic system and binary masking technique. Multimed Tools Appl 81:6331–6349. https://doi.org/10.1007/s11042-021-11757-4
18. Sayed WS, Tolba MF, Radwan AG, Abd-El-Hafiz SK, Soliman AM (2018) Security and efficiency of feistel networks versus discrete chaos for lightweight speech encryption. In: International Conference on Microelectronics (ICM). pp 92–95. https://doi.org/10.1109/ICM.2018.8704022
19. Shi C, Wang H, Hu Y, Qian Q, Zhao H (2019) A speech homomorphic encryption scheme with less data expansion in cloud computing. KSII Trans Int Inf Syst (TIIS) 13(5):2588–2609. https://doi.org/10.3837/tiis.2019.05.020
20. Slimani D, Merazka F (2018) Encryption of speech signal with multiple secret keys. Procedia Comput Sci 128:79–88. https://doi.org/10.1016/j.procs.2018.03.011
21. Thaine P, Penn G (2019) Extracting mel-frequency and bark-frequency cepstral coefficients from encrypted signals. In: INTERSPEECH. pp: 3715–3719. https://doi.org/10.21437/Interspeech.2019-1136
22. Tolba MF, Sayed WS, Radwan AG, Abd-El-Hafiz SK (2018) Chaos-based hardware speech encryption scheme using modified tent map and bit permutation. In: International Conference on Modern Circuits and Systems Technologies (MOCAST). pp 1–4. https://doi.org/10.1109/MOCAST.2018.8376621
23. Tolba MF, Elwakil AS, Orabi H et al (2020) FPGA implementation of a chaotic oscillator with odd/even symmetry and its application. Integration. 72:163–170. https://doi.org/10.1016/j.vlsi.2020.02.003
24. Xiang S, Luo X (2018) Reversible data hiding in homomorphic encrypted domain by mirroring Ciphertext group. IEEE Trans Circ Syst Vid Technol 28(11):3099–3110. https://doi.org/10.1109/TCSVT.2017.2742023
25. Yan X, Wu Q, Sun Y (2020) A homomorphic encryption and privacy protection method based on blockchain and edge computing. Wirel Commun Mob Comput 8832341:1–9. https://doi.org/10.1155/2020/8832341
26. Zhang QY, Jia YG (2022) A speech fully homomorphic encryption scheme for DGHV based on multithreading in cloud storage. Int J Netw Secur 24(6):1042–1055. https://doi.org/10.6633/IJNS.202211124(6).09
27. Zhang S, Gong Y, Yu D (2019) Encrypted Speech Recognition using deep polynomial networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp 5691–5695. https://doi.org/10.1109/ICASSP.2019.8683721

28. Zuber M, Carpov S, Sirdey R (2020) Towards real-time hidden speaker recognition by means of fully homomorphic encryption. In: International Conference on Information and Communications Security. pp 403–421. https://doi.org/10.1007/978-3-030-61078-4_23

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.