Article

Simultaneous Audio Encryption and Compression Using Parallel Compressive Sensing and Modified Toeplitz Measurement Matrix

Changchun Cai [†], Enjian Bai ^{*,†}, Xue-Qin Jiang and Yun Wu

School of Information Science & Technology, Donghua University, Shanghai 201620, China; cccllyg@163.com (C.C.); xqjiang@dhu.edu.cn (X.J.); wuyun_hit@dhu.edu.cn (Y.W.)

* Correspondence: baiej@dhu.edu.cn

† These authors contributed equally to this work.

Abstract: With the explosive growth of voice information interaction, there is an urgent need for safe and effective compression transmission methods. In this paper, compressive sensing is used to realize the compression and encryption of speech signals. Firstly, the scheme of linear feedback shift register combined with inner product to generate measurement matrix is proposed. Secondly, we adopt a new parallel compressive sensing technique to tremendously improve the processing efficiency. Further, the two parties in the communication adopt public key cryptosystem to safely share the key and select a different measurement matrix for each frame of the voice signal to ensure the security. This scheme greatly reduces the difficulty of generating measurement matrix in hardware and improves the processing efficiency. Compared with the existing scheme by Moreno-Alvarado et al., our scheme has reduced the execution time by approximately 8%, and the mean square error (MSE) has also been reduced by approximately 5%.

Citation: Cai, C.; Bai, E.; Jiang, X.-Q.; Wu, Y. Simultaneous Audio Encryption and Compression Using Parallel Compressive Sensing and Modified Toeplitz Measurement Matrix. *Electronics* **2021**, *10*, 2902. <https://doi.org/10.3390/electronics10232902>

Academic Editor: Paris Kitsos

Received: 4 November 2021

Accepted: 22 November 2021

Published: 24 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: logistic map; toeplitz matrix; linear feedback shift register (LFSR); inner product; UACI; NSCR

1. Introduction

With the rapid development of Internet and digital technology, a large amount of information is presented in the form of pictures and voices. This information is stored in various types of hard disks and contains a huge amount of information, which involves information security issues. Generally, in a sending and receiving system, compression and encryption are performed by the sending end, and decryption and decompression are performed by the receiving end. The purpose of compression is to reduce the size of data, transmit as much data as possible and reduce storage space, and the purpose of encryption is to ensure the confidentiality of data and prevent information leakage [1]. Therefore, the compressed information is stored before encryption, so the compressed information can be accessed before the encryption task is performed, so its security may be affected [2]. While compressed sensing (CS) can realize encryption and compression at the same time [3], there is no such problem.

As a new signal sampling technology, CS has attracted widespread attention in many fields such as image processing and speech processing. Many CS based compression and encryption schemes have been developed. Zhang et al. proposed a chaotic system and two-dimensional fractional Fourier transform for image encryption, which resulted in good compression performance, reconstruction robustness and high security [4]. In [5], Gong et al. first performed the Arnold transformation on the original image to reduce the blocking effect in the compression process, and then scrambled the compressed and sensed data to improve the reliability of the encryption algorithm. Zh et al. designed a picture embedding scheme based on adaptive threshold sparse algorithm and parallel compressed sensing to improve the quality of reconstructed pictures and the processing

speed of compressive sensing [6]. Combining compressive sensing and least significant bit (LSB) embedding [7], Chai proposed an effective visually meaningful image compression encryption scheme, which shows the effectiveness of the cryptographic system. At the same time, Chai et al. used the zigzag confusion to obfuscate the picture, and compress the picture into a cipher image, to then embed it in the carrier image using compressed sensing to obtain a visually secure cryptographic image. The algorithm is highly sensitive to plaintext images and has good visual security and data security [8]. Ye et al. embed the singular value of the secret image into the singular value of the carrier image with a certain embedding strength to obtain the final visually meaningful encrypted image [9]. In [10], two logistic maps are used to scramble the measurement matrix DCT and the sparse base matrix DWT, which improves the space of the key and increases the security. CS can also be used for encryption and recovery of speech signals. The article [11] used the contourlet transform to increase the sparsity of the CS required signal, and uses the randomness of the chaotic system and the high sensitivity of the initial conditions to design an effective speech encryption system. In [12], Haneche et al. designed a speech enhancement system based on compressive sensing, and the experimental results show the superiority of the method. In [2], Moreno-Alvarado et al. proposed a chaotic mixing method to generate the measurement matrix, which satisfies the encryption strength and saves the size of the shared key. As a traditional technique, the linear feedback shift register (LFSR) is widely used in information security owing its high-efficiency in hardware. In [13], a random bit stream generator based on LFSR is used to generate the initial state matrix of the cellular automaton. The results show that the encryption system has good reconstruction performance, robustness to noise, and security against multiple forms of attacks. In [14], the article develops a CS-based energy-saving encryption scheme based on linear feedback shift registers. The experimental simulation results show that the system can resist various attacks.

There are several problems in the known audio signal compression and encryption methods. Firstly, in the traditional method, compression and encryption are performed sequentially. Compressed data will inevitably be touched before performing encryption tasks, which will cause data security issues. For encryption and then compression, this kind of scheme must use a lossless compression scheme to avoid the leakage of encrypted information. Secondly, most compression schemes based on CS use the entire measurement matrix as the key, which makes the key consumption and storage space too large. Furthermore, even if the key is used to generate the measurement matrix, it takes a lot of time to generate a large number of measurement matrices. Thirdly, in practical applications, common compressed measurement matrices, such as random Bernoulli matrices and random Gaussian matrices, can meet the conditions of compressive sensing with a high probability, but practical applications are very difficult.

Based on the above analysis, this paper uses CS to encrypt and decrypt the audio signal, which improves the security of the signal transmission process. At the same time, this paper proposes a method to generate a measurement matrix using LFSR, which is conducive to the hardware realization and reduce the size of the shared key [15]. In addition, in order to improve the performance of measurement matrix in compressive sensing, after generating the Toeplitz matrix, the inner product selection method is adopted [16] in the process of generating the measurement matrix. Furthermore, in order to reduce the time to generate the measurement matrix, this paper uses a parallel CS method [6].

The rest of this paper is organized as follows. The second section introduces compressed sensing, Toeplitz matrix, logistic map and evaluation parameters, the third section introduces the proposed compressive sensing scheme, and the fourth section is the analysis of simulation results. Finally, the conclusions are presented.

2. Preliminary Work

2.1. Compressive Sensing

The mathematical derivation of the compressive sensing can be expressed in the following formula [17]:

$$\begin{cases} y = \Phi x \\ x = \Psi s \end{cases} \Rightarrow y = \Phi \Psi s, \quad (1)$$

where Φ, Ψ is measurement matrix and sparse transform matrix, respectively, x is the original signal and y is encryption signal x , s is the transformed sparse representation of signal. Figure 1 explains the frame of CS. $\hat{x}$ is the recovered signal and $\hat{s}$ is the recovered sparse representation of x .

The recovery algorithm of the compressive sensing is to reconstruct sparse signals s from a small number of linear observations y . In order to solve this equation, the restricted isometry property (RIP) condition needs to be satisfied, with the RIP is as follows:

$$(1 - \delta_k) \|z\|_2^2 \leq \|\Phi z\|_2^2 \leq (1 + \delta_k) \|z\|_2^2 \quad (2)$$

for all k -sparse signals z , where $\delta_k \in (0, 1)$.

Figure 1. The frame of CS.

The original signal can be recovered using convex optimized pairs [6]:

$$\min \|s\|_0 \quad s.t. \quad y = \Phi \Psi s, \quad (3)$$

where $\|s\|_0$ is the l_0 norm of the vector s , which is a non-deterministic polynomial NP problem, which can be transformed into a convex optimization problem:

$$\min \|s\|_1 \quad s.t. \quad y = \Phi \Psi s, \quad (4)$$

where $\|s\|_1$ is the l_1 norm of the vector s .

The performance of compressive sensing mainly depends on the sparse base matrix, measurement matrix and recovery algorithm. This paper uses discrete cosine transformation (DCT) as sparse transform, modified Toeplitz matrix as measurement matrix. In addition, we adopt convex optimization method to recovery original signal.

2.2. Toeplitz Matrix

As a method of compressive sensing measurement matrix, the Toeplitz matrix has the advantage that few elements need to be stored or transmitted. In particular:

$$T = \begin{bmatrix} t_{0,0} & t_{0,1} & \cdots & t_{0,n-2} & t_{0,n-1} \\ t_{1,0} & t_{1,1} & \cdots & t_{1,n-2} & t_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{m-2,0} & t_{m-2,1} & \cdots & t_{m-2,n-2} & t_{m-2,n-1} \\ t_{m-1,0} & t_{m-1,1} & \cdots & t_{m-1,n-2} & t_{m-1,n-1} \end{bmatrix} \quad (5)$$

is a Toeplitz matrix. For $\forall i, j, \theta \in N$ and $0 \leq i + \theta \leq m - 1, 0 \leq j + \theta \leq n - 1$, there are $t_{i+1, j+1} = t_{i, j}$. It can be easily found from (5) that the Toeplitz matrix can be directly determined by the elements in the first row and the first column. That is only $m + n - 1$ elements need to be transmitted for a $m \times n$ Toeplitz measurement matrix, and the elements along any diagonal parallel to the main diagonal (including the main diagonal) are the same [18].

LFSR is a mechanism that can generate a binary bit sequence. Its working principle is simply summarized, given the output of the previous state, the linear function of the output is used as the input again, and this cycle is performed. The XOR function is often used as the single-bit linear function [19].

To construct a Toeplitz matrix based on LFSR, the following conditions must be met. The number of bits in the register are equal to the number of rows in the Toeplitz matrix, and the current value of the register represents the LFSR state. In the Toeplitz matrix $T_{m \times n}$, each column of the Toeplitz matrix is a continuous LFSR state of length m . The number of matrix columns n are the total number of LFSR states. Therefore, the LFSR-based Toeplitz matrix is constructed as follows: (1) firstly, initialize the first column of the matrix, that is, determine the initial state of the LFSR; (2) then, move each column of the Toeplitz matrix down one unit, i.e., the LFSR moves one unit to the right; (3) next, update the first element of the current column by adding all XOR values obtained by XORing the elements of the previous column and the corresponding elements of the feedback polynomial (when the feedback polynomial of LFSR is a primitive polynomial, its output is a m -sequence); (4) finally, repeat (2) and (3) until all elements in the last column of the matrix are determined.

The elements of the primitive polynomial of degree m are XORed with the corresponding position elements of the previous column, and then the sum of all XOR values obtained is determined to the top elements of all columns except the first column. Suppose $P(x) = P_{m-1}x^m + P_{m-2}x^{m-1} + \dots + P_0x + 1$ is a primitive polynomial of degree m , and the corresponding coefficients except constant coefficient are $P = (P_0, P_1, \dots, P_{m-1})$, the current status of LFSR is $S_0 = (S_{0,0}, S_{1,0}, \dots, S_{m-1,0})$. The top element of the j th-LFSR state is

$$S_{0,j} = \bigoplus_{i=0}^{m-1} S_{i,j-1} \cdot P_i = S_{j-1} \cdot P^T, j = 1, 2, \dots \quad (6)$$

Once the top element of all columns are determined, the entire Toeplitz can be determined [7]. Figure 2 shows the generation process of Toeplitz matrix based on LFSR. The LFSR-based Toeplitz matrix only needs to store m elements of the first column, not the sum of the elements $m + n - 1$ of the first row and the first column. Therefore, this method saves the required hardware memory resources. The random generation of the first column of the matrix and the random selection of the primitive polynomial also provide the randomness of the Toeplitz matrix construction method at the same time.

The following is an example to illustrate the Toeplitz matrix generation process. The initial value of the shift register is $S_0 = (1, 0, 0, 0, 1)$. The primitive polynomial $P(x)$ is $x^5 + x^2 + 1$ and its corresponding coefficients vector is $P = (0, 1, 0, 0, 1)$. The number of columns in the Toeplitz matrix is 6. By shifting down we can obtain that the second to fifth elements of the second LFSR state are 1000. The first element of the second LFSR state is $S_0 \cdot P^T = 1$. Then the second LFSR state $S_1 = (1, 1, 0, 0, 0)$. Similarly, by shifting down we can obtain that the second to fifth elements of the third LFSR state are 1100. The first element of the third LFSR state is $S_1 \cdot P^T = 1$. Then the third state of LFSR is $S_2 = (1, 1, 1, 0, 0)$. Repeat above steps until the number of columns in the Toeplitz matrix reaches 6. The generated Toeplitz matrix follows that:

Figure 2. LFSR-based Toeplitz matrix generation process.

After generating the Toeplitz matrix, we use the algorithm proposed in [8] to perform inner product processing to produce measurement matrix pool (MMP) of size $m \times 2n$, where $\frac{m}{n}$ is the compression rate of compressive sensing. The following Algorithm 1 performs this process.

Algorithm 1: MMP.

- Step 1 Take the first column of Toeplitz as the i -th ($i = 1$) column of the measurement matrix pool.
- Step 2 Starting from the second column of the Toeplitz matrix, each column has an inner product with the first column. If the inner product is 0, the corresponding column is put into the measurement matrix pool as the $(i + 1)$ -th column. If it is not 0, discard it.
- Step 3 Repeat Step 2 until a $2n$ columns measurement matrix pool is generated.
-

2.3. Logistic Map

In this paper, we also use the logistic map to select the measurement matrix from measurement matrix pool. The logistic map is defined as [20]:

$$x_{n+1} = F(x_n) = \mu \cdot x_n \cdot (1 - x_n) \quad x_n \in [0, 1], \mu \in [3.57, 4]. \quad (7)$$

Algorithm 2 is the selection method proposed in [4], named parallel compression sensing (PCS).

Algorithm 2: PCS.

-
- Step 1 Use logistic map to generate a $2cn$ points sequence $S = \{s_1, s_2, \dots, s_{2cn}\}$ and convert S to $c \times 2n$ index matrix. Where c and $2n$ correspond to the total number of measurement matrices, the number of columns in the measurement matrix pool.
- Step 2 Let I_i the i -th row of the index matrix. Rearrange I_i from big to small to form $\text{Sort}(I_i)$. Take the original index position of the first n elements of $\text{Sort}(I_i)$ in I_i , record as W_i .
- Step 3 Take out the corresponding column from the matrix pool according to W_i to form the measurement matrix.
- Step 4 Repeat Steps 2 and 3 until all c measurement matrices are generated.
-

2.4. NSCR, UACI, MSE and PCCs

The following introduces the indicators for evaluating the strength of audio encryption and the quality of recovery.

The sampling rate of change (NSCR), determines the number of samples to be changed, and its formula is given as:

$$\text{NSCR} = \frac{1}{N} \sum_{i=1}^N D_i \times 100\%, D_i = \begin{cases} 1, & x_i \neq \hat{x}_i \\ 0, & x_i = \hat{x}_i \end{cases} \quad (8)$$

The average change intensity (UACI) determines the average number of encrypted speech intensity changes, defined as:

$$\text{UACI} = \frac{1}{N \times \max x_i} \sum_{i=1}^N |x_i - \hat{x}_i| \times 100\%, \quad (9)$$

where x_i and $\hat{x}_i$ are the i -th sample of two cyphered audio signals, whose original version is only different in one sample, and N represents the length of the audio frame.

Mean Square Error (MSE), a parameter of the quality of speech recovery, its physical meaning is the normalized mean square error between the original signal and the decoded signal, which is calculated as follows:

$$\text{MSE} = \frac{\sum_{i=1}^N (x_o(i) - x_d(i))^2}{\sum_{i=1}^N (x_o(i))^2}, \quad (10)$$

where $x_o(i)$ and $x_d(i)$ are the original and recovered signals, respectively.

Person correlation coefficient (PCCs), used to evaluate the similarity between the original signal and the decoded signal, is as follows:

$$\text{PCCs} = \frac{N \sum_{i=1}^N x_o(i) x_d(i) - \bar{x}_o \bar{x}_d}{\sqrt{N \bar{x}_o^2 - (\bar{x}_o)^2} \sqrt{N \bar{x}_d^2 - (\bar{x}_d)^2}}, \quad (11)$$

where

$$\bar{x}_{o,d} = \sum_{i=1}^N x_{o,d}(i), \quad (12)$$

$$\overline{x_{o,d}^2} = \sum_{i=1}^N (x_{o,d}(i))^2, \quad (13)$$

and $\bar{x}_{o,d}$ denotes either $\bar{x}_o$ or $\bar{x}_d$, $\overline{x_{o,d}^2}$ denotes either $\overline{x_o^2}$ or $\overline{x_d^2}$.

3. Proposed Algorithm

Figure 3 and 4 below are the overall block diagram of the solution proposed in this article. Figure 3 is the encryption block diagram and Figure 4 is the recovery block diagram. The following Table 1 is the parameters in Figures 3 and 4.

Table 1. Parameters and corresponding explanations

Parameter	Explanation
k_1	a set of keys, which is initial value of value, given by Sender
k_2	initial condition μ and x_n of logistic mapping
x	original signal
x_i	the i -th frame of x signal
Φ_i	the measurement matrix corresponding to x_i
s_i	DCT transform of x_i
y_i	CS processed transmission signal of x_i
m	frame length after compression
n	frame length before compression
$\hat{s}_i$	DCT domain signal of the i -th frame recovered by Receiver
$\hat{x}_i$	time domain signal of the i -th frame recovered by Receiver
$\hat{x}$	original signal recovered by Receiver

Figure 3. Sender's compression and encryption process.

As shown in Figure 3, the sender's compression and encryption process is as Algorithm 3.

The sender and the receiver share the key k_1 and k_2 through public key cryptography system such as RSA, ECC, etc. In this scheme, different frame use different CS measurement matrix for higher security, which will increase the data processing time, but the use of PCS and the simplicity of LFSR-based measurement matrix can make up for this problem. The security of the above encryption framework has been proven in [2]. As long as the key is only used once, the security of the entire encryption system can be guaranteed.

Contrary to the compression and encryption process, the decryption and decompression of receiver is shown in Figure 4 and Algorithm 4.

Algorithm 3: ECS.

-
- Step 1 Secret key k_1 is used as the initial value of the LFSR, given by sender. The corresponding Toeplitz matrix based on LFSR can be generated by k_1 .
- Step 2 The MMP algorithm is used to generate the measurement matrix pool of size $m \times 2n$ from Toeplitz matrix.
- Step 3 The PCS algorithm is used to generate the different measurement matrix Φ_i of size $m \times n$ from measurement matrix pool. The initial condition of Logical mapping is obtained from secret key k_2 , which is the hash function output of original signal.
- Step 4 Framing the original speech signal x into x_i with frame length of 400, then DCT transform x_i to produce s_i .
- Step 5 y_i is generated by compressed sensing of Φ_i and s_i , then sent to receiver.
-

Figure 4. Receiver's decryption and decompression process.

Algorithm 4: Receiver's decryption and decompression process.

-
- Step 1 The receiver obtains the key k_1 and k_2 after decrypting the public key cryptosystem.
- Step 2 k_1 is used to generate the same Toeplitz matrix.
- Step 3 Using MMP algorithm to generate the corresponding measurement matrix pool.
- Step 4 k_2 is the initial value of the logistic mapping, Use PCS algorithm to generate the measurement matrix Φ_i corresponding to each audio frame.
- Step 5 y_i is the encrypted voice frame sent by sender. The convex optimization method is used to recover $\hat{s}_i$, then the IDCT transform is used to generate $\hat{x}_i$, and the original audio signal $\hat{x}$ is finally recovered.
-

4. Results

In this scheme, we use a 20 KHz audio signal as experimental object, and a frame is 20 ms. There are 400 sample values in a frame, and a total of 100 frames are selected. In this experiment, the operating environment of MATLAB R2018a (64 bit) is 11th Gen Intel(R) Core(TM) i5-1135G7, CPU 2.40GHZ and memory 8.00GB, and the operating system is Microsoft Windows 10.

4.1. MSE and MSEError

Using the proposed scheme, we calculate the MSE under condition of m/n equal to 0.5, 0.7, 0.8 and 0.9, respectively. The results are shown in Figure 5a–d, where the data recovered using the correct measurement matrix. The data recovered using the error measurement matrix, called MSEError, is also shown in the figure. It can be clearly seen that MSEError fluctuates between 0.75 and 1.5 when $m/n = 0.5$, fluctuates between 0.5 and 1.5 when $m/n = 0.7$, fluctuates between 0.75 and 1.75 when $m/n = 0.8$ and 0.9. At the same time, the MSE under various compression conditions always remains at 0 and slightly fluctuates.

Figure 5. (a) Comparison of MSE and MSEError under $m/n = 0.5$. (b) Comparison of MSE and MSEError under $m/n = 0.7$. (c) Comparison of MSE and MSEError under $m/n = 0.8$. (d) Comparison of MSE and MSEError under $m/n = 0.9$.

4.2. PCCs and PCCsError

The data recovered from the correct and wrong measurement matrix, PCCs and PCCsError, are shown in Figure 6a–d under the condition of m/n equal to 0.5, 0.7, 0.8 and 0.9, respectively. It can be clearly seen that PCCsError fluctuates around 0 and PCCs is always maintained at 1 with a slight fluctuation when $m/n = 0.5, 0.7, 0.8, 0.9$. At the same time, the MSEError fluctuates between 0.75 and 1.8.

Figure 7a,b below shows the MSE and PCCs in the uncompressed state. It can be clearly seen that PCCsError fluctuates around 0 and PCCs is always maintained at 1 with a slight fluctuation when $m/n = 1$. At the same time, the MSE always remains at 0 and slightly fluctuates while MSEError fluctuates between 0.75 and 1.75 when $m/n = 1$.

Figure 8a,b is the comparison of the original signal and the recovery signal in the case of a frame of signal. Figure 8a is in the case of $m/n = 0.5$. It can be seen that the original signal and the recovery signal have obvious positions that do not overlap in time domain

and frequency domain. While in the case of $m/n = 0.8$, this phenomenon is significantly reduced in Figure 8b.

Figure 6. (a) Comparison of PCCs and PCCsError under $m/n = 0.5$. (b) Comparison of PCCs and PCCsError under $m/n = 0.7$. (c) Comparison of PCCs and PCCsError under $m/n = 0.8$. (d) Comparison of PCCs and PCCsError under $m/n = 0.9$.

Figure 7. (a) MSE comparison in the uncompressed state. (b) PCCs comparison in the uncompressed state.

The NSCR, UACI, MSE and PCCs obtained by the our scheme and the scheme proposed in Moreno-Alvarado [2] under the conditions when $m/n = 0.5, 0.7, 0.8, 0.9, 1$ are shown in Table 2. In the work by Moreno-Alvarado [2], the measurement matrix is constructed using a Gaussian random number generator and a chaotic mixing scheme. It can be seen that the scheme proposed in this paper performs well on both UACI and NSCR. Our

scheme has a certain improvement in the recovery of voice signal quality through the comparison of MSE indicators in different CR. The signal quality is improved by about 5%. At the same time, through the comparison of PCCs indicators, our scheme also has a better performance.

Table 2. NSCR, UACI, MSE, PCCs under different compression ratios (CR) in Moreno-Alvarado [2] and our scheme.

CR	NSCR		UACI		MSE		PCCs	
	[2]	Our	[2]	Our	[2]	Our	[2]	Our
$m/n = 0.5$	1	1	0.3132	0.3392	0.0240	0.0226	0.9641	0.9841
$m/n = 0.7$	1	1	0.3376	0.3266	0.0071	0.0067	0.9759	0.9959
$m/n = 0.8$	1	1	0.3215	0.3335	0.0028	0.0026	0.9782	0.9982
$m/n = 0.9$	1	1	0.3143	0.3353	0.0009711	0.00090757	0.9894	0.9994
$m/n = 1$	1	1	0.3365	0.3235	0.0000078464	0.00000074331	1.0000	1.0000

4.3. Running Time Analysis

Figure 9 shows the comparison of the running time in the entire encryption and decryption process between the method proposed in this article and the method proposed by Moreno-Alvarado [2]. It can be seen that our algorithm saves about 8% in time, which is due to the use of parallel compressive sensing algorithms.

4.4. Key Space Analyses

In terms of the size of the shared key, our scheme requires both parties to share the keys k_1 and k_2 . Among them, k_1 is the initial value of generating the Toeplitz matrix and its length is m , and k_2 is the initial value of the logistic mapping μ and x_n . So using our scheme to generate the measurement matrix instead of sharing the entire measurement matrix greatly reduces the key size. In addition, not only the initial value $key1$ generated by the measurement matrix needs to be shared, but also the permutation parameter $key2$ and the each frame also needs a random number of permutations $key3$ to ensure that the measurement matrix is generated randomly in [2], the shared key size in our scheme is more advantageous.

4.5. Robustness Analysis

4.5.1. Occlusion Attack

Robustness against occlusion in encryption systems is an important indicator. Furthermore, the loss of voice encryption samples has a bad impact on the recovery of the voice signal. The following Figure 10 shows the time domain and frequency domain recovery when the encrypted sample is intact in the encrypted sample, and the encrypted sample is lost by 1%, 5%, and 10%. Their corresponding signal-to-noise ratio (SNR) is 28.3123 dB, 25.2313 dB, 15.2321 dB respectively (the SNR is 32.8097 dB when the encrypted sample is no lost). It is very obvious that as the number of missing encrypted voice signal samples increases, the recovered voice signal becomes worse and worse. However, the recovered voice signal is still similar to the original signal. This shows that the voice encryption system has a certain degree of robustness against occlusion.

4.5.2. Noise Attack

The encrypted signal is transmitted in the channel and is often affected by various noises, such as Gaussian noise (GN), pepper noise (SPN). When the Gaussian noise intensity is 0.001%, 0.003%, 0.005%, 0.009%, the corresponding SNR is 32.8083 dB, 32.7103 dB, 32.5832 dB, 32.5353 dB, respectively. The value of SNR changes very slightly. The following Figure 11 shows the performance of the recovered speech signal in the time domain and frequency domain when the Gaussian noise intensity of the speech encryption sample is 0.001%, 0.003%, 0.005% and 0.009%, respectively. Analyzing from Figure 11 and the data, GN has

little effect on speech recovery, indicating that our scheme has anti-interference ability against Gaussian noise.

Figure 8. (a) Comparison of the original signal and the recovery signal in the case of a frame of signal and $m/n = 0.5$, the top is the time domain and the bottom is the frequency domain. (b) Comparison of the original signal and the recovery signal in the case of a frame of signal and $m/n = 0.8$, the top is the time domain and the bottom is the frequency domain.

Figure 9. The percentage of time that our proposed algorithm saves compared to the Moreno-Alvarado [2] scheme is under different compression ratios.

Figure 10. (a) $m/n = 0.7$, the time domain and frequency domain recovery of the encrypted sample without losing signal. (b) $m/n = 0.7$, the time domain and frequency domain recovery of the signal when the encrypted sample loses 1%. (c) $m/n = 0.7$, the time domain and frequency domain recovery of the signal when the encrypted sample is lost 3%. (d) $m/n = 0.7$, the time domain and frequency domain recovery of the signal when 5% of the encrypted sample is lost.

Figure 11. (a) When $m/n = 0.7$, Gaussian noise intensity is 0.001%, the performance of the recovered speech signal in the time domain and frequency domain. (b) $m/n = 0.7$, Gaussian noise intensity is 0.003%, the performance of the recovered speech signal in the time domain and frequency domain. (c) $m/n = 0.7$, the performance of the recovered speech signal in the time domain and frequency domain when the Gaussian noise intensity is 0.005%. (d) $m/n = 0.7$, the performance of the recovered speech signal in the time domain and frequency domain when the Gaussian noise intensity is 0.009%.

5. Discussion

The solution in this paper is mainly to use the feature of compressive sensing that can be encrypted and compressed at the same time to improve the security of the system, and to use the feature of the Toeplitz matrix to reduce the size of the shared key.

In the scheme, we use a public key cryptography system such as RSA, ECC to share secret key k_1 and k_2 , which can be realized before the sending audio signal. Therefore, RSA or ECC has no impact on the performance of the proposed scheme.

In the scheme, we use a one-dimensional logistic map to select the measurement matrix, which may bring some security problems by the method of phase space reconstruction. In fact, the proposed scheme is a general framework, we can use a complex high-dimensional chaotic system to obtain better security.

In addition, the design of sparse basis matrix is also a method to improve system security, which will be studied in the future.

6. Conclusions

This article uses compressive sensing to realize the compression and encryption of audio signals. The key is shared in a confidential channel. At the sending end, the signal is divided into frames and then processed by DCT, and the measurement matrix generated by the LFSR-based Toeplitz matrix and the inner product selection. Since the measurement matrix required for each frame is different, this paper designs a parallel compressive sensing algorithm. At the decryption side, the shared key is used to generate the required

measurement matrix, the original frame is obtained through the convex optimization algorithm and IDCT, and the original signal is restored.

Furthermore, the Toeplitz matrix based on LFSR used in this paper is easy to implement in hardware, and it also reduces the size of the shared key.

Finally, from the simulation results, it can be seen that the encryption and recovery of audio signals are more secure, and the encryption and recovery time is greatly reduced, which is about 8% less than [2]. NSCR and UACI meet security requirements. At the same time, the signal recovery quality is improved to a certain extent. Compared with [2], it has increased by about 5%. Furthermore, the encryption system has certain robustness and can resist occlusion attack and noise attack.

Author Contributions: C.C., conceptualization, methodology, software, writing—original draft preparation; E.B., conceptualization, methodology, writing—review and editing; X.-Q.J., writing—review and editing, project administration, funding acquisition; Y.W., supervision, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of Shanghai (Grant no. 20ZR1400700), the Shanghai Municipal Science and Technology Major Project (Grant no. 2019SHZDZX01), the National Natural Science Foundation of China (Grant no. 61772129) and the State Key Laboratory of Advanced Optical Communication Systems and Networks (Grant no. 2020GZKF002), China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs Publicly available datasets were analyzed in this study. This data can be found here: [<https://github.com/Wonder555123/Simultaneous-Audio-Encryption-and-Compression-Using-Parallel-Compressive-Sensing-and-Modified-Toepl.git>], accessed on 4 November 2021].

Acknowledgments: This work was supported by the National Natural Science Foundation of Shanghai (Grant no. 20ZR1400700), the Shanghai Municipal Science and Technology Major Project (Grant no. 2019SHZDZX01), the National Natural Science Foundation of China (Grant no. 61772129) and the State Key Laboratory of Advanced Optical Communication Systems and Networks (Grant no. 2020GZKF002), China.

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

References

1. Zhu, S.; Zhu, C.; Wang, W. A Novel Image Compression-Encryption Scheme Based on Chaos and Compression Sensing. *IEEE Access* **2018**, *6*, 67095–67107. [[CrossRef](#)]
2. Moreno-Alvarado, R.; Rivera-Jaramillo, E.; Nakano, M. Simultaneous Audio Encryption and Compression Using Compressive Sensing Techniques. *Electronics* **2020**, *9*, 863. [[CrossRef](#)]
3. Donoho, D.L. Compressed Sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
4. Zhang, M.; Tong, X.J.; Liu, J. Image Compression and Encryption Scheme Based on Compressive Sensing and Fourier Transform. *IEEE Access* **2020**, *99*, 1. [[CrossRef](#)]
5. Gong, L.; Qiu, K.; Deng, C. An Image Compression and Encryption Algorithm Based on Chaotic System and Compressive Sensing. *Opt. Laser Technol.* **2019**, *115*, 257–267. [[CrossRef](#)]
6. Zh, A.; Kz, A.; Yl, B. Visually Secure Image Encryption Using Adaptive-thresholding Sparsification and Parallel Compressive Sensing. *Signal Process.* **2021**, *18*, 107998.
7. Chai, X. An Efficient Visually Meaningful Image Compression and Encryption Scheme Based on Compressive Sensing and Dynamic LSB Embedding. *Opt. Lasers Eng.* **2020**, *124*, 105837. [[CrossRef](#)]
8. Chai, X.; Gan, Z.; Chen, Y. A Visually Secure Image Encryption Scheme Based on Compressive Sensing. *Signal Process.* **2016**, *134*, 35–51. [[CrossRef](#)]
9. Ye, G.; Pan, C.; Dong, Y. Image Encryption and Hiding Algorithm Based on Compressive Sensing and Random Numbers Insertion. *Signal Process.* **2020**, *172*, 107563. [[CrossRef](#)]
10. Wang, Z.; Hussein, Z.S.; Wang, X. Secure Compressive Sensing of Images Based on Combined Chaotic DWT Sparse Basis and Chaotic DCT Measurement Matrix. *Opt. Lasers Eng.* **2020**, *134*, 106246. [[CrossRef](#)]
11. Mahmood, A.; Gaze, A.M. Combined Speech Compression and Encryption Using Chaotic Compressive Sensing with Large Key Size. *IET Signal Process.* **2017**, *12*, 6.

12. Haneche, H.; Boudraa, B.; Ouahabi, A. A New Way to Enhance Speech Signal Based on Compressed Sensing. *Measurement* **2019**, *151*, 107117. [[CrossRef](#)]
13. George, S.N.; Augustine, N.; Pattathil, D.P. Audio Security through Compressive Sampling and Cellular Automata. *Multimed. Tools Appl.* **2015**, *74*, 10393–10417. [[CrossRef](#)]
14. Unde, A.S.; Dp, P. Design and Analysis of Compressive Sensing Based Lightweight Encryption Scheme for Multimedia IoT. *IEEE Trans. Circuits Syst. Express Briefs* **2019**, *67*, 167–171. [[CrossRef](#)]
15. Li, D.; Peng, H.; Zhou, Y. Memory-saving Implementation of High-Speed Privacy Amplification Algorithm for Continuous-Variable Quantum Key Distribution. *IEEE Photonics J.* **2018**, *5*, 1. [[CrossRef](#)]
16. Nouasria, H.; Et-Tolba, M. New Constructions of Bernoulli and Gaussian Sensing Matrices for Compressive Sensing. In Proceedings of the International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco, 1–4 November 2017; pp: 1–5.
17. Zhang, Y.; Zhang, L.Y.; Zhou, J. A Review of Compressive Sensing in Information Security Field. *IEEE Access* **2017**, *4*, 1 [[CrossRef](#)]
18. Bennett, C.H.; Brassard, G. Quantum cryptography: Public Key Distribution and Coin tossing. *Theor. Comput. Sci.* **2014**, *560*, 7–11. [[CrossRef](#)]
19. Diamanti, E.; Leverrier, A. Distributing Secret Keys with Quantum Continuous Variables: Principle, Security and Implementations. *Entropy* **2015**, *17*, 6072–6092. [[CrossRef](#)]
20. Lei, Y.; Barbot, J.P.; Gang, Z. Compressive Sensing with Chaotic Sequence. *IEEE Signal Process. Lett.* **2010**, *17*, 731–734. [[CrossRef](#)]