applied sciences

MDPI

Article

Spreading-Based Voice Encryption by Means of OVSF Codes

Diego Renza [1,*,†], Dora M. Ballesteros [1,†] and Estibaliz Martinez [2,†]

[1] Faculty of Engineering, Universidad Militar Nueva Granada, Bogotá 110111, Colombia; dora.ballesteros@unimilitar.edu.co

[2] Computer Systems Architecture and Technology, Universidad Politecnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain; emartinez@fi.upm.es

* Correspondence: diego.renza@unimilitar.edu.co; Tel.: +57-1-650-0000

† These authors contributed equally to this work.

Received: 13 November 2019; Accepted: 18 December 2019; Published: 21 December 2019

Abstract: This paper presents a new methodology to encrypt voice signals, in such a way that they simulate being a noise signal. The objective is to obtain a signal that does not generate suspicions about its content, while protecting the message. The process is based on the spread and scrambling of the signal through the use of OVSF (orthogonal variable spreading factor) codes. The security of the method is based mainly on the input value used for the randomization phase. From a computational cost point of view, the use of fixed-length codes allows for a shorter retrieval time compared to a similar method based on variable-length codes. Regarding the encrypted signal, its main characteristic is its high entropy (very close to the ideal value). Finally, in the recovery process, a signal identical to the original is obtained.

Keywords: signal encryption; ciphered voice; spreading; privacy; noise

1. Introduction

Lately, with the advance of technology, the volume of information processed daily has grown exponentially. In most cases, the information transmitted becomes important, especially in communication systems where the content of the information is confidential and can be valuable. Therefore, the security of the information has become a necessity when transmitting private information.

In recent years, some authors have proposed new methodologies aimed at providing a level of data security and integrity. The purpose of transmitting information under any secure communication method is to provide a level of privacy over the data, which means limiting access by a third party or unauthorized people [1,2].

In this regard, encryption is a security mechanism used to provide confidentiality and protection to a system, guaranteeing reliability and protection over the information to be transmitted. Accordingly, different techniques have been implemented based on the codification of voice signals in order to obtain an undecipherable audio signal whose characteristics are totally different with respect to the original one, including security, confidentiality, and integrity aspects [3].

Some encryption algorithms such as DES (data encryption standard), AES (advanced encryption standard), Blowfish, and RSA (Rivest, Shamir y Adleman) are standard algorithms frequently used for the encryption of different data types, such as voice [2,4]. Methods based on scrambling systems are also commonly used to modify the information of an input signal. For example, using a key or seed value to generate a variety of coded signals provides high reliability [5]. Chaotic maps contribute an essential part in the encryption of voice signals and algorithms that are implemented and use chaotic

maps are effective and safe [6]. For example, the ADPCM (adaptive differential pulse code modulation) coding and the generation of logistic maps for transformation into binary words are used to perform substitution operations on audio signals with sensitive information [7].

Algorithms based on Henen chaotic maps and chaotic economic map have also been proposed, to create independent sequences with each other, adding more sensitivity and complexity to the scheme [8,9]. Another approach of dual-channel audio encryption is based on chaotic systems with changeable multi-scroll, in order to generate one time-key reliant on the file's hash value, and transforming the original speech samples position in order to make brute-force attack impossible [10]. In addition, voice encryption in communication systems can be used for multichannel digital audio signals, according to the technique called binaural cue coding (BCC). In this case, the audio signal is broken down into frequency bands to extract the most important information and the use of pseudo-random sources generates a binary flow that allows the mixing of information to transmit an encoded audio signal [11].

Finally, an encryption technique for audio signals based on the Collatz conjecture has been proposed in [12]. The authors use a coding process using variable length codes based on the Collatz conjecture. In order to increase the level of security and uncertainty of the output signal, there is a spread process through a coding based on the Collatz conjecture, ensuring that samples of the modified signal are uncorrelated with the original one. As a result, an audio signal with similar characteristics to a noise signal is transmitted. Since the used codes are a variable length, the process has a high computational cost, mainly in the recovering stage.

According to the above, this document presents an improved methodology aimed at encrypting the original information of a voice signal. In this case, the principal goal is to transform the original content of a voice signal to an unintelligible signal by third parties through the spread and scrambling of the input samples. For the spreading process, we use OVSF-based (orthogonal variable spreading factor) codes, previously randomized. As a result, we obtain a modified audio signal whose appearance resembles a noise signal, i.e., it does not represent correlation with the original samples, providing a high level of security and confidentiality to third parties who wish to know the content of the said signal.

2. Orthogonal Variable Spreading Factor Codes

In Universal Mobile Telecommunications Systems (UMTS), a spreading process is applied in the physical layer. This process involves two operations: Channelization and scrambling. Channelization increases the bandwidth of the signal, transforming each input symbol into a number of chips, being the Spreading Factor (SF) the number of chips per symbol. Then, a scrambling code is applied to the spread signal [13].

The channelization codes used in UMTS are known as OVSF (orthogonal variable spreading factor) codes, and its main characteristic is to preserve the orthogonality between different physical channels. To construct OVSF codes, Equations (1)–(3) can be used.

$$C_{1,0} = [1] \tag{1}$$

$$\begin{bmatrix} C_{2,0} \\ C_{2,1} \end{bmatrix} = \begin{bmatrix} C_{1,0} & C_{1,0} \\ C_{1,0} & -C_{1,0} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

$$C_{SF,k} = \begin{bmatrix} C_{2^{n+1},0} \\ C_{2^{n+1},1} \\ C_{2^{n+1},2} \\ C_{2^{n+1},3} \\ \vdots \\ \\ C_{2^{n+1},2^{n+1}-2} \\ C_{2^{n+1},2^{n+1}-1} \end{bmatrix} = \begin{bmatrix} C_{2^n,0} & C_{2^n,0} \\ C_{2^n,0} & -C_{2^n,0} \\ C_{2^n,1} & C_{2^n,1} \\ C_{2^n,1} & -C_{2^n,1} \\ \vdots & \\ \\ C_{2^n,2^n-1} & C_{2^n,2^n-1} \\ C_{2^n,2^n-1} & -C_{2^n,2^n-1} \end{bmatrix} \quad (3)$$

Each code is formally specified as $C_{SF,k}$ where SF stands for spreading factor, and k is the code number, $0 \leq k \leq SF - 1$.

3. Proposed Method

The proposed encryption scheme for voice signals is based on the spreading of the input signal using OVSF codes, obtaining a signal that will sound like noise. Next, the operation of the proposed method will be explained, including the encryption and recovery stages.

3.1. Voice Encryption

The goal of the voice encryption stage is to convert a secret voice signal ($S(n); 1 \leq n \leq N$) in a noise signal ($T(m); 1 \leq m \leq M$), where N and M are the number of samples of the input signal and the ciphered signal, respectively. The number of bits per sample in the input signal will be referred as b, whereby the value of each sample of the input signal may have 2^b possible amplitude values.

The proposed process to cipher the voice signal involves the spreading and scrambling of the signal. The spreading consists of replacing each sample of the input signal with a binary code. For this to be possible, there must be as many codes as sample amplitude values. To achieve this goal, the first step is to generate 2^b OVSF codes using Equations (1)–(3), and $SF = 2^b$. This results in 2^b different codes of 2^b bits each, that will be called $C_{SF,k}$, where $0 \leq k \leq 2^b - 1$. In other words, C_{SF} is a $2^b \times 2^b$ matrix, where each row corresponds to an OVSF code.

The scrambling of the signal is made directly in the OVSF code matrix. To achieve this, the position of each bit in the OVSF code matrix is reordered. First, a 2^b-element vector (P), which contains the position of each bit, is generated and obtained as follows:

$$P_0 = \begin{bmatrix} 0 & 1 & 2 & \dots 2^b - 1 \end{bmatrix}, \quad (4)$$

and,

$$P = perm\left(R_0\right), \quad (5)$$

the $perm(P_0)$ function reallocate the P_0 vector, and returns a vector containing a pseudo-random permutation of the elements in P. The seed to initialize the pseudo-random process is an input of the system (given by, for example, a cryptographically secure pseudo-random number generator).

To randomize the bits in the C_{SF} matrix according to the pseudo-randomly permuted vector, every value of P is used as the index to extract the values in the C_{SF} matrix, as follows:

$$SC\left(i\right) = C_{SF}\left(P\left(i\right)\right), \quad (6)$$

Accordingly, the ith element in the SC matrix is the $P(i)$th position of $C_{SF,k}$ matrix. The scrambled codes (SC) is a matrix of size 2^b.

To spread the $S(n)$ voice signal using the scrambled OVSF codes, each sample value of the voice signal will be replaced by a row of the SC matrix. Since the value of the sample can take one of 2^b values, its integer value ($[0 \;\; 2^b - 1]$) is used to select the corresponding row number in the SC matrix.

In other words, if the sample value is 200, this sample value will be replaced by the row number 200 (a 2^b bits vector). Following, the input voice signal will be converted to a spread binary signal (SB), as follows:

$$SB = \left[SC_{S(1),*} \quad SC_{S(2),*} \quad \ldots SC_{S(N),*} \right], \tag{7}$$

where $SC_{i,*}$ denotes the ith row of SC.

After this stage, the voice input has been converted to a $2^b * N$ bits vector.

To obtain the samples of the output signal, the binary vector is divided into words of w bits. Each of this words, will be obtained as follows:

$$T_b(m) = \left\{ SB(w * m + j), \quad 1 \leq j \leq w, \quad 0 \leq m \leq M - 1 \right\}, \tag{8}$$

where M is the number of samples of the covert signal, and is given by:

$$M = \frac{2^b * N}{w}, \tag{9}$$

where N is the number of samples of the input signal, b is the number of bits per sample of the input signal, and w is the number of bits per sample of the output signal. Therefore, the relationship between the signal time of the output and that of the input signal, maintaining the sampling frequency, is given by M/N.

Samples of the binary signal T_b can be converted to a floating point value, whose amplitude is normalized to a standard value (for example $[-1\ 1]$). After the voice encryption process, a signal with unintelligible content (T) is obtained. This signal can be transmitted without raising suspicion of its content.

3.2. Voice Recovery

Once the signal has been received, it is possible to recover the original content. The inputs of this module are the secret signal ($T_r(m); 1 \leq m \leq M$) and the secret key (seed value). The process starts with the generation of the 2^b OVSF codes using Equations (1)–(3), and $SF = 2^b$, in a similar way to how they were generated in the encryption process. In addition, the OVSF code matrix is scrambled according to Equations (4) and (5), where the seed used to initialize the pseudo-random process is the same as that used in the encryption process.

On the other hand, the amplitude value of each sample of the received signal ($T_r(m)$) is converted to a non-negative integer decimal value ($[0\ 2^w - 1]$). Subsequently, this value is represented in binary with at least w bits. The received binary signal will be represented as T_{rb}. This binary signal is divided into words of b bits. Each of this words, will be obtained as follows:

$$R_b(n) = \left\{ T_{rb}(b * n + j), \quad 1 \leq j \leq b, \quad 0 \leq n \leq N - 1 \right\}, \tag{10}$$

To extract the corresponding sample value, a search algorithm is used. Each data of b bits in the vector R_b, is compared against each row of the SC matrix. The output of this process is the index (or row) in SC that matches $R_b(n)$. The position (value) of the ith data of R_b corresponds to the ith sample value of the recovered signal ($R(n)$).

$$R(n) = i \quad where \quad R_b(n) = SC_{i,*} \ . \tag{11}$$

Samples of the recovered signal $R(n)$ can be converted to a floating point value, whose amplitude is normalized to a standard value (for example $[-1\ 1]$). The content of the recovered signal, $(R(n); 1 \leq n \leq N)$, must be identical to the original signal $S(n)$.

4. Experimental Results and Analysis

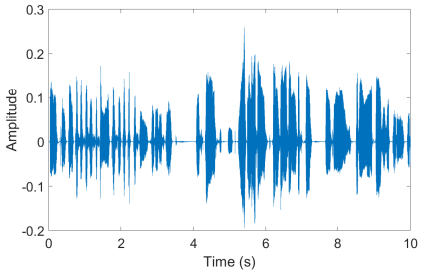
4.1. Dataset and Parameters

For the application and validation of the method, 85 voice recordings were used, with a sampling frequency of 8 kHz and 8 bits per sample ($b = 8$). The duration of each recording is 10 s (80,000 samples), and they are monophonic.

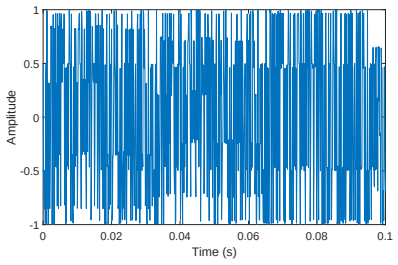
Accordingly, if $b = 8$, C_{SF} is a 256 × 256 matrix, where each row corresponds to an OVSF code. After scrambling, the scrambled codes (SC, size: 256 × 256) are obtained with a different random seed value used to encrypt each of the 85 test signals. From the matrix SC, the values of the input signal were spread, replacing each input sample with a 256-bit vector. The size of the obtained binary vector (SB) is 256 × 80,000 = 20.48 × 10^6 bits.

Meanwhile, the number of bits chosen for the secret signal is $w = 16$. In this way, the secret signal will have 1.28 × 10^6 samples, equivalent to a signal of 160 s with a sampling frequency of 8 kHz. In other words, the expansion rate of the signal is 1:16. This expansion rate is constant as long as b, w, and f_s (sampling frequency of the encrypted signal) are constant. If w or f_s are doubled, the expansion rate is reduced by half, and vice versa.

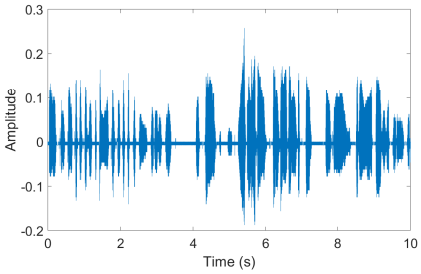
The signal obtained shows no traces of the original signal and its content is unintelligible. This signal together with the seed value of the randomization will be the input data for the recovery process. Figure 1 shows an example of the original signal, the encrypted signal and the recovered signal.



(**a**) Original signal



(**b**) Encrypted signal (0.1-second zoom)



(**c**) Recovered signal

Figure 1. Example of the proposed method (signals in time domain).

4.2. Assessment

In order to evaluate the proposed method, the uncertainty of the signals and the similarity between them were evaluated.

- Uncertainty: The uncertainty of the signals was assessed quantitatively through the entropy and the disorder level (DL). Entropy measures the information content of the signal, and it can be calculated using the Shannon entropy (Equation (12)) [14].

$$H(X) = -\sum_{i}^{M} P(x_i) \log_2(P(x_i)), \tag{12}$$

where X is the entire speech signal, x_i is the ith sample of the signal, $P(x_i)$ is the probability of the occurrence of x_i, and M is the finite number of samples. The maximum entropy value is reached when all amplitudes of the signal are of equal likelihood, in which case it is equal to the number of bits per sample. On the contrary, if the signal has a unique symbol, the entropy is zero. Thus, entropy is a measure of the uncertainty of data [15], with higher entropy values indicating higher uncertainty.

In a complementary way, the disorder level (DL) is a metric that compares each sample with its previous and subsequent sample. For a signal with a low uncertainty level, the values of the adjacent samples will be very similar, whereas in a signal with a high uncertainty level, the difference between adjacent samples will be significant. DL is calculated using Equation (13) [16].

$$DL = \frac{\sum_{i=2}^{M-1} \sqrt{|x_i - x_{i+1}| + |x_i - x_{i-1}|}}{M-2}, \tag{13}$$

where, x_i is the ith sample of the voice signal and M is the number of samples. According to the above, the higher the DL value, the lower the intelligibility of the signal. The maximum DL value is $\sqrt{2Vpp}$, where Vpp is the signal's peak-to-peak amplitude (i.e., 2 for signals ranging between -1 and 1).

- Similarity: The idea here is to evaluate if the encrypted signal has traces of the original signal (ideally the similarity must be low), and if the content of the recovered signal corresponds to that of the original signal. Thus, the similarity between the signals must be assessed both in time domain (correlation coefficient (CC), Equation (14)) [17] and frequency domain (spectral distortion (S_pD), Equation (15)) [18].

  If X and Y are the original and encrypted signals, σ_X and σ_Y their standard deviations, CC is given by,

$$CC(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}. \tag{14}$$

  If $V_x(j)$ and $V_y(j)$ are the spectra (dB) of the original and encrypted signals, J the number of points used in the transform, S_pD (dB) is given by,

$$S_pD = \frac{1}{J} \sum_{j=0}^{J-1} |V_x(j) - V_y(j)|. \tag{15}$$

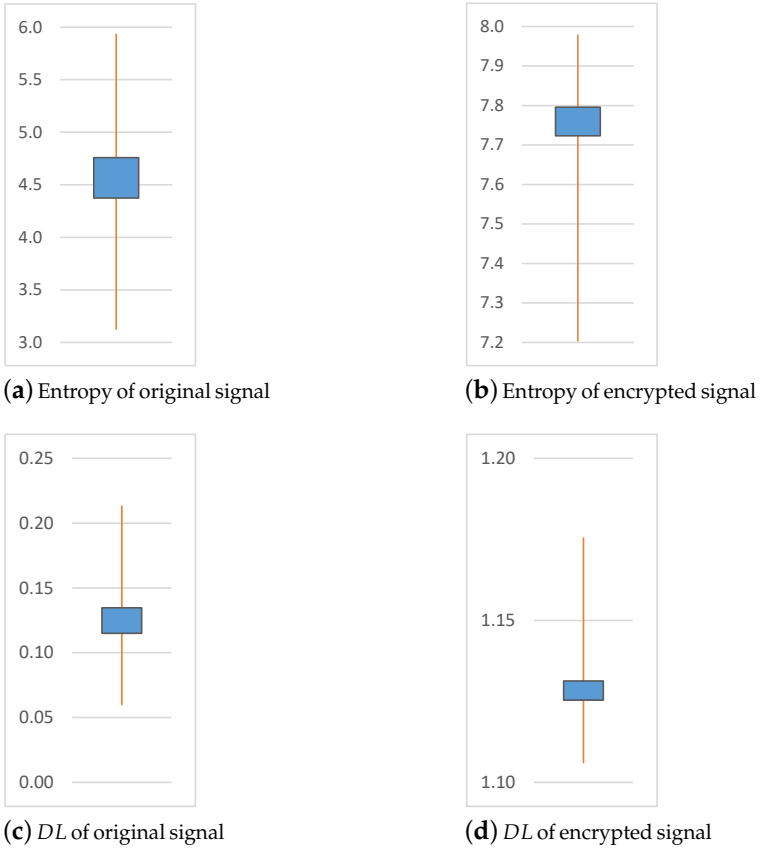
4.3. Results and Analysis

To implement and evaluate the proposed method, each voice file of the dataset was encrypted with a different seed value (85 tests) and then recovered back. Here, the encrypted file was compared against the original one, through CC and S_pD. Additionally, the entropy and DL of both signals were calculated. Finally, the similarity between the original and recovered signals was assessed through CC.

4.3.1. Uncertainty

The entropy and the disorder level of the original and recovered signal are shown in Figure 2 through confidence ranges (95%). For the case of entropy, most of the values from the original signal were around 4.6 (Figure 2a), whereas for the encrypted signal they were around 7.76 (Figure 2b). Something similar happened with the level of disorder, where the original signal had DL values that concentrated around 0.13 (Figure 2c), while the DL values in the encrypted signal oscillated around 1.13 (Figure 2d). This means that the degree of uncertainty of the data in the encrypted signal is

significantly greater than in the original signal, approaching its ideal value (i.e., 8 for entropy, and 2 for *DL*).



(**a**) Entropy of original signal (**b**) Entropy of encrypted signal

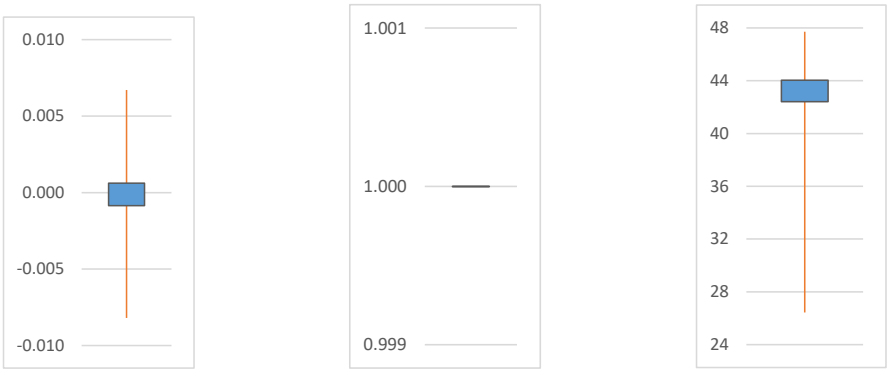
(**c**) *DL* of original signal (**d**) *DL* of encrypted signal

Figure 2. Uncertainty analysis. Confidence range charts (95%) for Entropy and Disorder Level (*DL*).

Thus, the high entropy and *DL* values of the encrypted signal would not raise any suspicion when it is transmitted, and in turn the behavior of the signal moves away from deterministic behavior.

4.3.2. Similarity

Regarding the similarity between the original, encrypted, and recovered signals, the *CC* and the *DL* results are shown in Figure 3. In these results it can be seen that the similarity between the original and encrypted signal is practically null. In this case, the correlation between the two signals was practically zero (Figure 3b), while the spectral distortion between the two was concentrated around 43 dB (Figure 3c). On the other hand, the correlation between the input signal and recovered signal showed that the original content of the signal was recovered. Here it is important to highlight that the encrypted signal differed both in duration and content, with respect to the original signal. In addition, its content was unintelligible.

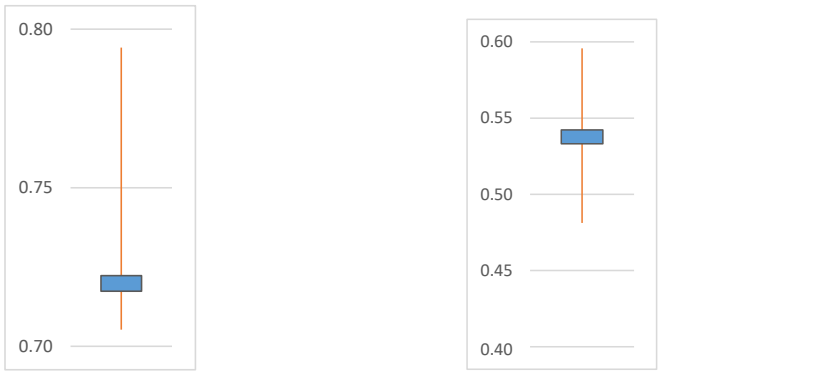


(**a**) *CC* between original and encrypted signal (**b**) *CC* between original and recovered signal (**c**) S_pD between original and encrypted signal

Figure 3. Similarity analysis. Confidence range charts (95%) for correlation coefficient (*CC*) and spectral distortion (S_pD).

4.3.3. Computational Cost

As discussed above, the tests performed used a 10 s signal, and a 160 s signal was obtained. Figure 4 shows the computational times for both the encryption process and recovery process. In either case, the execution times were shorter than the duration of the original signal, which is important for real-time systems. These tests were performed on a PC with Windows 10 Pro 64-bit operating system, an Intel Core i7-4790S processor at 3.2 GHz, 8 GB of RAM using Matlab R2019b.



(**a**) Time (s) for encryption execution (**b**) Time (s) for decryption execution

Figure 4. Computational cost analysis. Confidence range charts (95%) for encryption (*CC*) and decryption execution times.

4.3.4. Comparison with Other Methods

To compare the proposed algorithm against similar approaches, key space, entropy of the encrypted signal, and correlation between the original and encrypted signals were taken into account (Table 1).

In the proposed scheme, the size of the key was the largest of the compared methods, which allowed a greater computational requirement for brute force attacks. In turn, the entropy was very close to the ideal value, which guaranteed a high uncertainty in the behavior of the data. Likewise, the proposed scheme presented a null correlation between the original and encrypted signal, and consequently the scheme presented a low probability against statistical attacks.

Table 1. Comparison of the proposed algorithm with similar approaches.

Method	Key Space	Entropy (Encrypted Signal)	Correlation (Original-Encrypted)	Average Recovery Time Relative to Signal Duration (%)
[19]	2^{477}	Not given	0.000236	Not given
[10]	3.4×10^{80}	Not given	−0.000198	Not given
[20]	2^{256}	Not given	0.002000	Not given
[12]	$(2^8)! \approx 8.58 \times 10^{506}$	7.4	0.000080	8.82 %
Ours	$(2^{2b})! = (2^{16})!$ for $b = 8$ bits	7.76	0.000119	5.38 %

In terms of computational cost in the recovery process, the proposed approach was compared to a similar method using variable length codes obtained through a binarization process based on the Collatz Conjecture [12]. For this comparison, all 85 test signals were used, coded, and then decoded. The execution time of the recovery process was measured and the relationship between the recovery time and signal duration was evaluated. The PC used for these tests was the same as the one used in Section 4.3.3. The results showed that the recovery time in the proposed method was 5.38% of the duration of the original signal, while in the method based on variable length codes was 8.82%. This means that, in the proposed method, the recovery time was reduced by approximately 39% compared to the method based on variable length codes. The reduction of the recovery time of the proposed method was mainly based on the use of fixed length codes. This allows data to be structured in matrices, and operations to be performed with vectorized implementation.

5. Conclusions

The proposed scheme presented an alternative for the secure transmission of voice signals. Its main characteristic is the generation of a signal with a high level of uncertainty (which sounds like noise), that has no relation to the original signal. The experimental validation showed a high entropy value of the encrypted signal and a very low correlation between the original and encrypted signal. On the other hand, the key space was very high thanks to the process of spreading and scrambling. Finally, the execution times for encryption and recovery allowed the application of the algorithm in real-time systems.

As future work we propose to analyze the memory consumption of this solution and the implementation of this solution in mobile phone systems.

Author Contributions: Conceptualization, D.R.; Formal analysis, D.M.B.; Investigation, D.R.; Methodology, D.M.B.; Supervision, E.M.; Validation, D.R.; Visualization, E.M.; Writing—original draft, D.M.B.; Writing—review and editing, E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Ballesteros, D.M.; Renza, D.; Camacho, S. An unconditionally secure speech scrambling scheme based on an imitation process to a gaussian noise signal. *J. Inf. Hiding Multimedia Signal Process.* **2016**, *7*, 233–242.
2. Talha, S.K.; Barry, B.I.A. Evaluating the impact of AES encryption algorithm on Voice over Internet Protocol (VoIP) systems. In Proceedings of the 2013 International Conference o Computing, Electrical and Electronic Engineering (ICCEEE), Khartoum, Sudan, 26–28 August 2013. doi:10.1109/icceee.2013.6634023. [CrossRef]
3. Kaur, M.; Kaur, M.S. Survey of Various Encryption Techniques for Audio Data. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2014**, *4*, 1314–1317.
4. Knudsen, R.A.E.B.L. Serpent: A proposal for the advanced encryption standard. In Proceedings of the First Advanced Encryption Standard (AES) Conference, Ventura, CA, USA, 20–22 August 1998.

5. John, J.E.; Ajai, A.R.; Poornachandran, P. Effective Implementation of DES Algorithm for Voice Scrambling. In Proceedings of the International Conference on Security in Computer Networks and Distributed Systems, Trivandrum, India, 11–12 October 2012; pp. 75–84.
6. Lian, S.; Sun, J.; Wang, Z. A block cipher based on a suitable use of the chaotic standard map. *Chaos Solitons Fractals* **2005**, *26*, 117–129. [CrossRef]
7. Hamdi, M.; Rhouma, R.; Belghith, S. An appropriate system for securing real-time voice communication based on ADPCM coding and chaotic maps. *Multimedia Tools Appl.* **2017**, *76*, 7105–7128. [CrossRef]
8. Zhou, X.; Tang, X. Research and implementation of RSA algorithm for encryption and decryption. In Proceedings of the 2011 6th International Forum on Strategic Technology, Harbin, China, 22–24 August 2011; Volume 2, pp. 1118–1121.
9. Parvees, M.M.; Samath, J.A.; Bose, B.P. Audio encryption—A chaos-based data byte scrambling technique. *Int. J. Appl. Syst. Stud.* **2018**, *8*, 51–75. [CrossRef]
10. Liu, H.; Kadir, A.; Li, Y. Audio encryption scheme by confusion and diffusion based on multi-scroll chaotic system and one-time keys. *Optik* **2016**, *127*, 7431–7438. [CrossRef]
11. Jaillet, F.; Virette, D. Encoding of Multichannel Digital Audio Signals. U.S. Patent 8,964,994, 24 February 2015.
12. Renza, D.; Mendoza, S.; Ballesteros, D.M. High-uncertainty audio signal encryption based on the Collatz conjecture. *J. Inf. Secur. Appl.* **2019**, *46*, 62–69. [CrossRef]
13. Renza, D.; Ballesteros, D.M.; Lemus, C. Authenticity verification of audio signals based on fragile watermarking for audio forensics. *Expert Syst. Appl.* **2018**, *91*, 211–222. doi:10.1016/j.eswa.2017.09.003. [CrossRef]
14. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
15. Robinson, D.W. Entropy and uncertainty. *Entropy* **2008**, *10*, 493–506. [CrossRef]
16. Ballesteros, D.M.; Sandoval, A.; Renza, D. Evolutionary Algorithm for Speech Scrambling based on Asexual Reproduction. *J. Inf. Hiding Multimedia Signal Process.* **2018**, *9*, 796–806.
17. Wang, J. Pearson Correlation Coefficient. In *Encyclopedia of Systems Biology*; Springer: New York, NY, USA, 2013; p. 1671.
18. Mosa, E.; Messiha, N.W.; Zahran, O.; El-Samie, F.E.A. Chaotic encryption of speech signals. *Int. J. Speech Technol.* **2011**, *14*, 285. [CrossRef]
19. Farsana, F.; Gopakumar, K. A novel approach for speech encryption: Zaslavsky map as pseudo random number generator. *Procedia Comput. Sci.* **2016**, *93*, 816–823. [CrossRef]
20. Lima, J.B.; da Silva Neto, E.F. Audio encryption based on the cosine number transform. *Multimedia Tools Appl.* **2015**, *75*, 8403–8418. [CrossRef]