Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **1**, January **2017**

Security Analysis of The Speech Scrambling Method Based on Imitation of A Super-Gaussian Signal

Dora M. Ballesteros L., Diego Renza, and Steven Camacho

Telecommunications Engineering
Universidad Militar Nueva Granada
Carrera 11 101-80, Bogotá-Colombia
dora.ballesteros@unimilitar.edu.co, diego.renza@unimilitar.edu.co, u1400943@unimilitar.edu.co

Received May, 2016; revised September, 2016

Abstract. *A scheme of speech scrambling based on imitation of a super Gaussian signal has recently been proposed. This scheme gives scrambled signals with zero trace of the original content and high scrambling degree. In this paper, the scheme is used in two scenarios: scrambling with private key method (symmetric cryptography), and scrambling with one private and one public key method (a hybrid between symmetric and asymmetric cryptography). Each scenario is analyzed in terms of security to determine if an intruder can reveal the secret content. In the first scenario three kinds of attacks are taken into account: known plaintext, brute force, and known ciphertext. In the second scenario only the brute force attack is considered, because only one information type is transmitted between Alice and Bob. According to theoretical analysis and experimental tests, it has been concluded that the scheme based on imitation of a Super Gaussian signal is highly secure, and that the original content cannot be revealed by a non-authorized user.*
Keywords: Speech scrambling, imitation; Security; Cryptanalysis.

1. **Introduction.** Scrambling is a useful tool for privacy protection of secret information like audio [1, 2, 3], image [4, 5] or video [6, 7, 8]. Its purpose is to modify the content of a signal before transmission in such a way that it is not intelligible. With the appropriate key, only the authorized user may be able to recover the secret content.

There are several approaches to the scrambling of speech signals. The core of a scrambling method is to transpose the samples (or spectral coefficients) of the secret message in new places according to a key. Key generation can obey pseudorandom processes, chaotic maps [9, 10, 11], or sequences obtained by artificial intelligence like cellular automata [3, 12], genetic algorithms or imitation of target signals [13, 14]. Generally, the main objective of a scrambling scheme is to achieve zero residual intelligibility, and some of the available methods have reached this aim; however, there are challenges that still remain. One of these challenges is resistance to cryptanalysis, because although recent advances in computing have allowed the development of more sophisticated scrambling schemes, cryptanalysis attacks are more complex, too.

Cryptanalysis corresponds to active attacks on the system to identify the key or the secret content without prior knowledge of the key. Here, if the scrambling system follows Kerckhoff's assumption, the security of the system relies only on the key because the method is public [15]. Typically, cryptanalysis encompasses *brute force* attack, *known plaintext* attack and *ciphertext* attack. The first one consists of testing all available keys of the *keyspace* until the secret content is revealed. In the second one, the attacker

has some pairs of *plaintext* and *ciphertext*, and then s/he discovers the key. In the last one, which is the most used in real cases, the attacker accesses the *ciphertext* but not the *plaintext* or the key and through mathematical operations s/he identifies the secret content [16, 17].

The best way to resist *brute force* attacks is through a huge *keyspace* (e.g. 10^{150} [18]) with equally likely keys. The higher the *keyspace*, the higher the amount of effort required to breach the system. The attempts of the designer consist of creating many equally likely keys with a large size. In the case of *known plain text* attacks, if the system works with dynamic mapping (i.e. the relationship of *plaintext* vs. *ciphertext* changes every time), it resists this kind of attack. In this case, the designer's attempt rests on making many equally likely relationships between the input and output signals. Finally, some methods of *ciphertext* attacks have been successful; for example, in the case of image scrambling, there have already been incidents of breached security using this attack [19, 20, 21]. In the case of speech scrambling, the secret content can be discovered through the manipulation of the spectrogram of the scrambled speech signal [22]. To do this, it is well known that the spectrogram of a scrambled speech signal has abrupt changes, but the spectrogram of a natural speech signal does not. Therefore, the spectrogram of the scrambled speech signal can be used like a puzzle; its pieces can be arranged in order to obtain a continuous behavior; the result is the spectrogram of the speech signal. The system can only overcome this kind of attack if there are several likely solutions to the puzzle problem.

According to the above, this paper presents a theoretical and experimental analysis of the security of the speech scrambling scheme based on imitation of a super Gaussian signal proposed in a previous work [13]. Since the studied scheme can be used in two different scenarios (as mentioned in the abstract), cryptanalysis is performed in each one. The paper is organized as follows. Section 2 presents a background of the studied scheme. It encompasses scrambling with *private key* (symmetric cryptography), and scrambling with one *private* and one *public key* (a hybrid between symmetric and asymmetric cryptography). Section 3 presents the security analysis of the two proposed scenarios in terms of typical cryptanalysis attacks. Section 4 illustrates some applications of the proposed scenarios, and finally, in Section 5 the work is concluded.

2. **Description of the Speech Scrambling Scheme under Study.**

The scheme under analysis uses imitation between a speech signal with intelligible content and a super Gaussian noise signal (i.e. a signal with a probability density function similar to a Gaussian signal but kurtosis higher than three). Imitation is successful because the probability density function (pdf) of speech signals is similar to Gaussian pdf with fatter tails pdf [23]. In [13] we found that kurtosis of speech signals is close to six and then a super Gaussian signal with this value of kurtosis is adequate to be imitated. Specifically, the Laplacian distribution is an example of a Gaussian function which satisfies the above condition and has proved to be a good choice to model speech signals [24, 25, 26, 27].

Our proposed scheme can work with either dynamic generation of Laplacian noise signals to obtain the scrambled speech signal (i.e. symmetric cryptography system) or with a fixed Laplacian noise signal to obtain a *public key* (i.e. a hybrid of symmetric and asymmetric cryptography). In the following section, the proposed scenarios are explained.

2.1. **First scenario: scrambling with *private key*.**

In this case, the speech signal (i.e. the secret message) imitates a Laplacian noise signal which is generated in situ by the system. *Alice* sends the scrambled signal and *private key* to *Bob* (S_c) through two different channels, as illustrated in Figure 1.

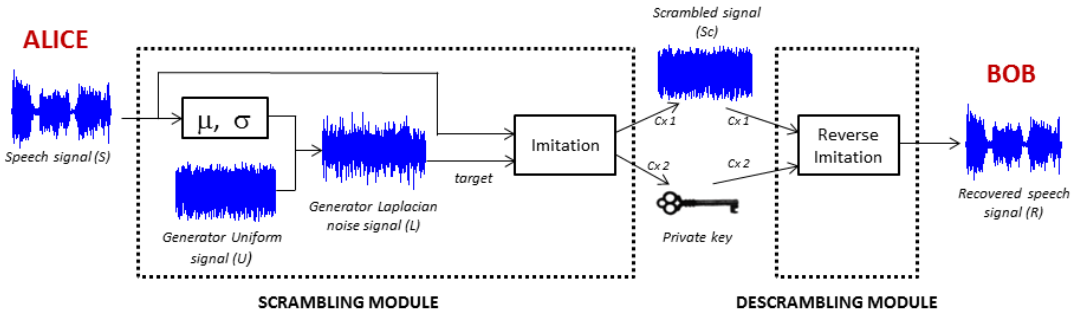

FIGURE 1. Proposed scheme with *private key* (first scenario).

The scrambling module contains two main parts: Laplacian noise generator and Imitation. Each block is explained as follows:

Laplacian noise generator: the input of this block is the speech signal and the output is the Laplacian noise signal, generated in situ. From the speech signal, the mean (μ) and the standard deviation (σ) are calculated. On the other hand, a discrete uniform signal (U) is generated in the interval (1/2, 1/2], with the same number of samples of the speech signal. Therefore, a Laplacian noise signal (L) is calculated according to Equation 1.

$$L = \mu - \frac{\sigma}{\sqrt{2}} sng(U) ln(1 - 2|U|) \tag{1}$$

The pdf of the Laplacian noise signal is modeled as Equation 2.

$$f(x) = \frac{1}{\sqrt{2\sigma}} e^{-\left(\frac{\sqrt{2}|x-\mu|}{\sigma}\right)} \tag{2}$$

Imitation: the input of this block is the Laplacian noise signal generated in situ, and the outputs are the scrambled speech signal (S_c) and the *private key*. Imitation is made possible by the ability to adapt the speech signals [28].

Since the *pdf* of the speech signal is very similar to the *pdf* of the Laplacian noise signal, they can imitate each other's behavior by means of a permutation process. The largest sample of the original signal is placed in the position of the largest sample of the target signal. Then, the second largest sample of the original signal is placed in the position of the second largest sample of the target signal. The above procedure is repeated until all samples of the original signal have been relocated.

We illustrate the imitation process with an example:

Let S a signal with data $S = [9 \quad 8 \quad 7 \quad 6 \quad 2 \quad 4 \quad 3 \quad 5 \quad 1 \quad 0]$ and L a signal with data $L = [10 \quad 18 \quad 0 \quad 14 \quad 6 \quad 16 \quad 8 \quad 12 \quad 2 \quad 4]$. You can note that the content of S and L are different, nevertheless, imitation is feasible. The first step is finding the largest elements of S and L, and their linear indices; in this case the largest elements are 9 and 18, respectively, and their linear indices are 1 and 2, respectively. Then, number 9 is placed in the 2^{nd} position of the signal S_c, and number 1 is placed in the 2^{nd} position of the *private key*. The second step is finding the second largest elements of S and L, which are 8 and 16, respectively. In addition, their linear indices are 2 and 6. Next, number 8

is placed in the 6^{th} position of the signal S_c, and number 2 is placed in the 6^{th} position of the *private key*. The above procedure is performed until the smallest data of the signal S is relocated. At the end, the signal S_c and the *private key* are:

$$S_c = [5 \quad 9 \quad 0 \quad 7 \quad 3 \quad 8 \quad 4 \quad 6 \quad 1 \quad 2]$$
$$private\ key = [8 \quad 1 \quad 10 \quad 3 \quad 7 \quad 2 \quad 6 \quad 4 \quad 9 \quad 5]$$

More detail about the imitation process is presented in [13]. It is worth noting that the signal S_c and the *private key* must be transmitted by two different channels.

The descrambling module contains only one block, explained as follows:

Reverse imitation: in this block the inputs are the scrambled signal S_c and the *private key*. The process consists of relocating data of S_c according to information of the *private key*. If *Bob* does not have the correct *private key*, he cannot recover the signal S.

To recover the original signal, the first values of S_c and the *private key* are read. Using the above example, number 5 is placed in the 8^{th} position of the recovered signal, R. Then, the second values of S_c and *private key* are read. Next, number 9 is placed in the 1^{st} position of R. This procedure is repeated for all data of S_c. At the end, the recovered signal is $R = [9 \quad 8 \quad 7 \quad 6 \quad 2 \quad 4 \quad 3 \quad 5 \quad 1 \quad 0]$. You can note that the R and S signals are equal; it means that scrambling based on imitation is a completely reversible method.

2.2. **Second scenario: scrambling with one *private* and *public key*.** In this scenario, the Laplacian noise signal is not generated in situ. Both *Alice* and *Bob* have the same *private key*, before starting communication (previously stored in the device). Only the *public key* is transmitted between them. Figure 2 illustrates this scenario.

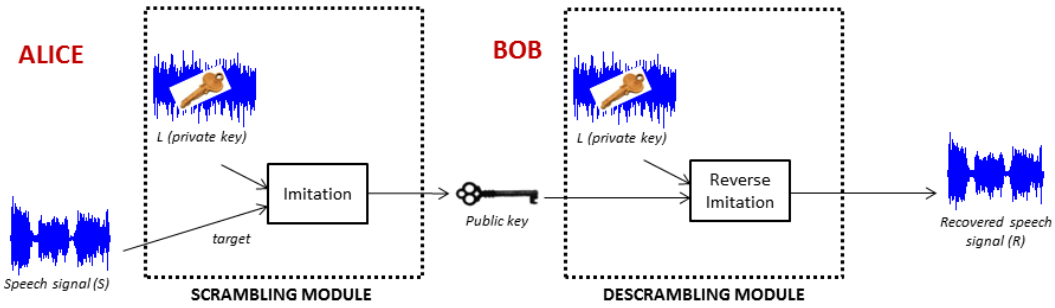

FIGURE 2. Proposed scheme with private target and *public key* (second scenario).

Unlike the first scenario, the vector that contains the mapping between the places of the original speech signal and the scrambled speech signal (i.e. the *key*) is public. It does not matter if this information is intercepted by a third party, since if s/he does not have the correct Laplacian noise signal (i.e. *private key*), s/he will not be able to reveal the secret content.

According to several tests, the speech signals have a mean close to zero and standard deviation close to 0.2. A Laplacian noise signal, which acts like the *private key*, is generated from the above values and it is stored in *Alice* and *Bob* devices. Then, when *Alice* wants to transmit a speech signal, this signal imitates the *private key*. Although the results of the imitation process are the scrambled signal and the *key*, only the *key* is transmitted by a public channel. When *Bob* receives the *public key*, he uses his *private key* (i.e. the fixed Laplacian noise signal) to recover the secret content.

3. **Security analysis of the speech scrambling scheme.** In this Section, the speech scrambling scheme proposed by Ballesteros et al. [13] is evaluated in two scenarios of secure communication. In the first case, it is supposed that the scrambled speech signal is transmitted by a public channel, and the key is transmitted by a private channel. In the second case, only the key is transmitted between the parties by a public/private channel.

3.1. **Security analysis of the first scenario.** Security analysis consists of evaluating if an unauthorized third party of the communication (i.e. *Eve*) can reveal the secret content from the knowledge of some data. It is focused on three kinds of attacks: *known plaintext*, *brute force*, and *known ciphertext*.

***Known plaintext* attack:** this attack is successful if the system works with a fixed mapping between the speech signal and the scrambled signal. In our case, even with the same speech signal, the scrambled signal changes each time the algorithm runs, because the target signal (i.e. Laplacian noise signal) is always new. Then, mapping between them is dynamic and *Eve* cannot determine the current *key* by detecting a past key.

In the following example, *Alice* has generated ten scrambled signals from the same secret message and she obtained ten keys. Then, there are ten mappings between the speech signal and the scrambled speech signals. The example can be followed at `https://www.mathworks.com/matlabcentral/fileexchange/59365-audio-descrambling`, a Matlab code (scenario1test1.m) allows the loading of ten scrambled signals and its keys, and deciphering the signal. Figure 3 shows the scrambled signals and Figure 4 the deciphered signal obtained by *Eve*.

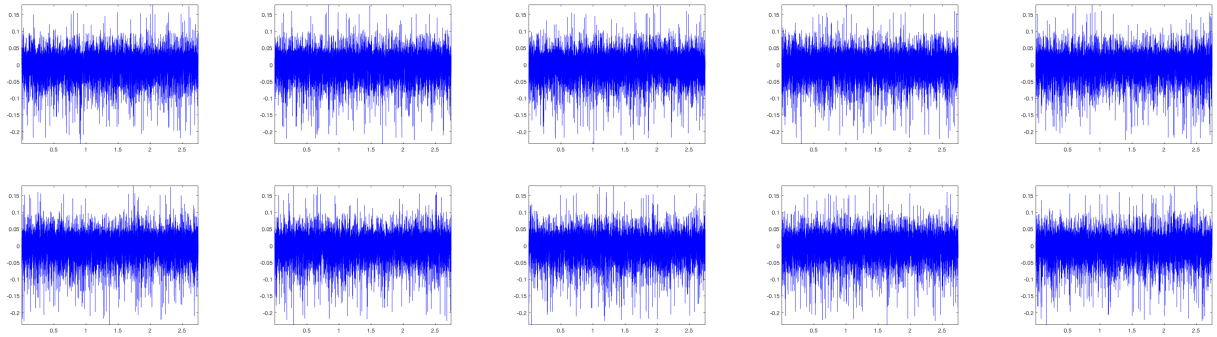

FIGURE 3. *Known plaintext* attack example: scrambled1 to scrambled10.

The reader can corroborate that although the scrambled signals and their keys are different from each other, the deciphered signals are the same. In each case, the deciphered signal corresponds to the same woman saying the phrase "Let me know if you have any concerns".

It means that whenever *Alice* runs the algorithm, even with the same speech signal, a different mapping, i.e. a different *key* is obtained. Then, if *Eve* intercepts both a speech signal and its scrambled signal, and determines its *key*, this mapping is not successful to decipher another scrambled signal.

***Brute force* attack:** in this attack, *Eve* tests all available keys and reveals the secret content. If the total number of available keys is small (i.e. the *keyspace* is small), the system is weak because time to find the correct key is short; otherwise, the system is strong in terms of *brute force* attack. In our scheme [13], length of the secret key is equal to the total number of samples (m) of the secret message and it contains the numbers 1

to m in disorderly places. Then, if m is the length of the speech signal, the *keyspace* is theoretically equal to $m!$. However, in a real case, it should be considered that Laplacian noise obeys to a pseudo-random generator and then some sequences may not exist. In practice, the result is less than $m!$. In spite of that, the keyspace is big enough to ensure a long time to test of available keys.

For example, if the speech signal is one-second and the frequency sampling is 8 KHz, the *keyspace* is 8000!. Therefore, even with a pseudo-random generator of the Laplacian noise signal, the total number of available keys is long (e.g. $3200! > 10^{9800}$). Then, *Eve* will take several years to test all available keys even with a cluster of super computers.

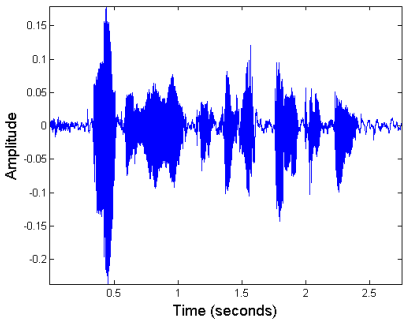

FIGURE 4. *Known plaintext* attack example: recovered secret message.

***Known ciphertext* attack:** in this attack *Eve* accesses the scrambled speech signal and has enough time and computation resources to obtain several keys. Since each key allows the acquisition of one deciphered signal, she tries to reject the wrong deciphered signals and select the correct one. However, this is not an easy task for *Eve*, because there are many deciphered signals with intelligible content and any of them could be the correct secret message.

In order to illustrate *Eve's* challenge, we are going to provide an example. Suppose *Eve* intercepts a scrambled speech signal (Figure 5). This signal looks and sounds like noise and a priori, it does not provide any clues about the secret message. Then, she obtains ten keys of the *keyspace* and she should reject the wrongly deciphered signals and identify the correct one. Figure 6 shows ten deciphered signals discovered by *Eve*.

The characteristics of the deciphered signals obtained by *Eve* are:

a. Five signals are in English and five are in Spanish.
b. They correspond to two speakers: one male and other female.
c. All deciphered signals have intelligible content. The following are the deciphered signals.
   From female speaker:
   Deciphered 1: please, turn off the light
   Deciphered 2: it's time to start
   Deciphered 3: ok, let's begin
   Deciphered 4: in the last lecture
   Deciphered 5: some examples
   From male speaker:
   Deciphered 6: apoyo administrativo
   Deciphered 7: boletín de investigación
   Deciphered 8: formatos del programa
   Deciphered 9: resultados preliminares

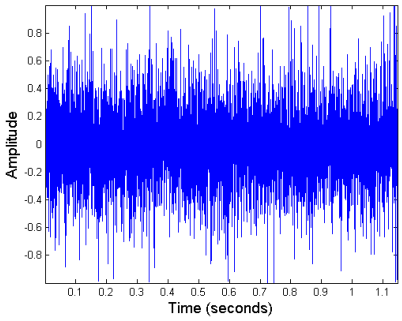

FIGURE 5. *Known ciphertext* attack example: scrambled speech signal.

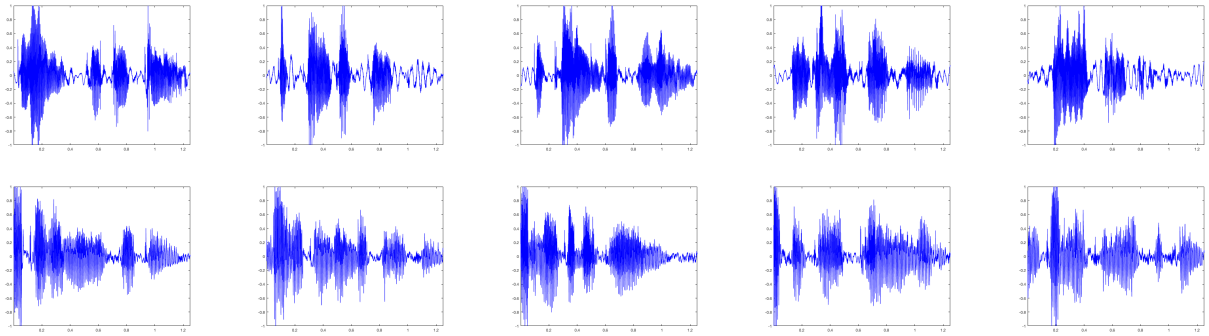

FIGURE 6. *Known ciphertext* attack example: deciphered signals with ten specific keys.

Deciphered 10: internacionalización

According to the above results, with a simple inspection (e.g. plotting or listening to the deciphered signals), *Eve* cannot identify which is the correct secret message. Therefore, she should use mathematical analysis to try to overcome her challenge.

A typical analysis in TSP (Time Scrambling Permutation) schemes, like our proposal, is the spectrogram of the speech signal. It is well known that the spectrogram of a natural speech signal has continuous shape, but the spectrogram of a scrambled speech signal has abrupt transitions in the border of sub-bands and time segments [22]. Then, *Eve* expects to reject wrongly deciphered signals according to the analysis of the spectrogram. However, to her surprise, all deciphered signals have spectrograms with natural behavior. Figure 7 shows the spectrogram of the scrambled speech signal and the spectrograms of the deciphered signals of Figure 6.

In a second attempt, *Eve* uses statistical analysis of the deciphered signals, specifically the histogram behavior. If the histogram behavior is similar to a super Gaussian distribution with smooth shape, she classifies the corresponding signal as a probable recovered speech signal; otherwise, the signal is classified as wrong. Nevertheless, since the scrambling method based on imitation does not change the histogram of the signal (it means the histogram of the scrambled speech signal is equal to that of the secret message), all histograms of the deciphered signals have the same behavior. And again, *Eve* cannot reject the deciphered signals because of their histograms. Figure 8 shows the histogram of the scrambled signal and the histogram of one deciphered signal (e.g. with $key1$).

For a third attempt, *Eve* analyzes the correlation between two adjacent samples (Figure 9). This evaluation has been very successful in determining the quality of scrambled

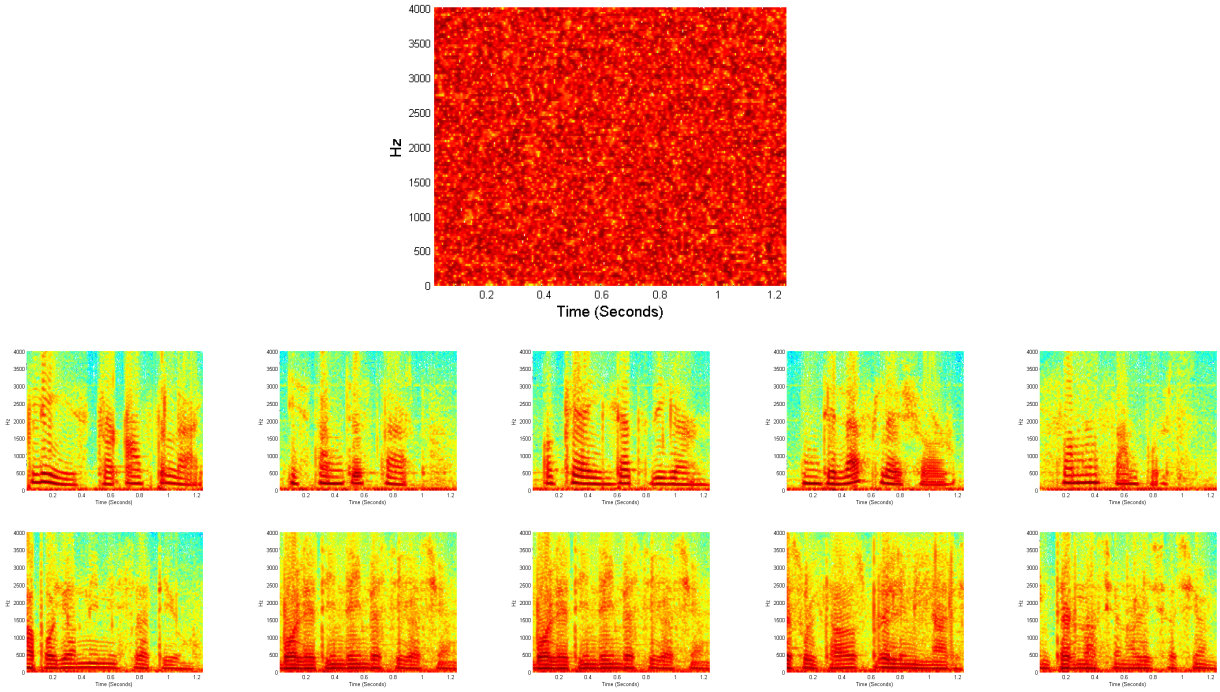

FIGURE 7. *Known ciphertext* attack example: spectrograms of the scrambled signal and ten deciphered signals.

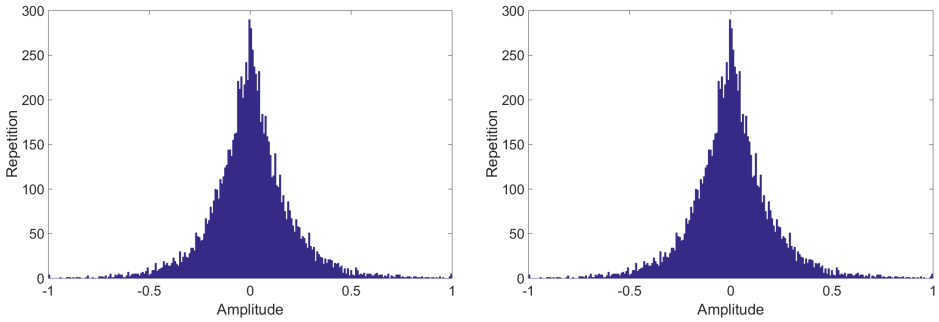

FIGURE 8. *Known ciphertext* attack example: histograms of the scrambled signal (left) and deciphered signal (right).

signals [29]. In the case of natural speech signals, the plot of the correlation of two adjacent samples is a set of dots around the identity line, in a similar way to the results of Figure 9. Accordingly, *Eve* cannot reject any of the deciphered signals with this attempt because in all cases the behavior corresponds to a natural speech signal.

In summary, the deciphered signals obtained with the ten selected keys are good candidates to be the secret message, and then, *Eve* cannot identify which is the correct one. Since the scheme of Ballesteros et al. [13] works with equally likely keys, many deciphered signals have natural behavior, and *Eve* cannot identify which is the correct secret message.

The files to test the above *known ciphertext* attacks can be accessed at `https://www.mathworks.com/matlabcentral/fileexchange/59365-audio-descrambling`. A Matlab code (scenario1test2.m) is available for each of the results. The code reads one scrambled signal and ten different keys, giving the results showed in Figures 5 to 9.

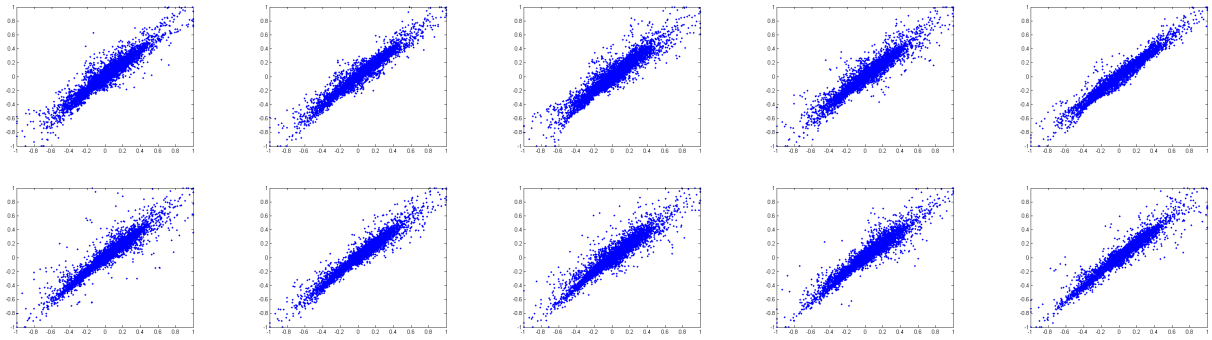

FIGURE 9. *Known ciphertext* attack example: correlation between two adjacent samples.

3.2. **Security analysis of the second scenario.** In this scenario, only the *public key* which contains the mapping between the places of the speech signal and the scrambled speech signal is sent by *Alice* to *Bob*. Although *Alice* can use a private channel, *Eve* may intercept the *public key*. *Eve*, with knowledge of the *public key*, should now try to reveal the secret message. She creates a Laplacian noise signal with the same length of the key, having zero mean and standard deviation of 0.2. Then, she uses the generated Laplacian noise signal to decipher the message. If the result does not have intelligible content, she tests with a new Laplacian noise signal until the result is an intelligible signal (i.e. *brute force* attack). Again, it is not an easy task for *Eve*, as we illustrate with the following example. It works with the same data used in the *known plaintext* attack. Currently, the first key is intercepted by *Eve*, and nine Laplacian signals are selected to try to decipher the secret message.

Figure 10 shows each of the deciphered signals obtained by *Eve*.

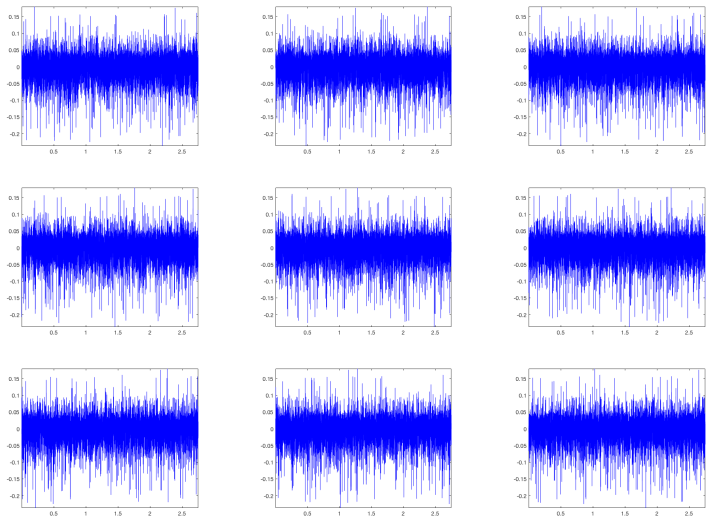

FIGURE 10. Results of deciphered signals with incorrect Laplacian noise signals.

Although *Eve* can find a Laplacian noise signal which gives her a deciphered signal with intelligible content, she has no way to be sure if the selected Laplacian signal is the correct one, because the *Laplacian space* is equal to the *keyspace* with a size of $m!$, and all Laplacian noise signals are equally likely.

In summary, even if *Eve* intercepts the *private key*, she cannot reveal the secret message by means of *brute force* attack. The other analysis of the first scenario (i.e. of *known*

plaintext, and *known ciphertext*) is not applied to this scenario, because the scrambled signal is not transmitted, only the key is.

The files to test the second scenario can be accessed at `https://www.mathworks.com/matlabcentral/fileexchange/59365-audio-descrambling`, a Matlab code (scenario2test1.m) is available. The code reads the key and nine Laplacian noise signals, giving the results showed in Figure 10.

4. **Applications of the current scrambling scheme.** Scrambling is a classic technique traditionally used to protect the information content. In the first scenario, the proposed scheme is useful for covert communication in which the secret message is transmitted in a disguised form. Since the aim of covert communication is to preserve the privacy of information, it is necessary for the system work with unconditional security, as in our scheme. According to theoretic analysis and experimental tests, even if *Eve* knows how the system works and intercepts the *private key* or the scrambled speech signal, she cannot reveal the secret content.

In the second scenario, the proposed scheme can also be used as cryptographic hash function in applications like authenticity. The aim of authenticity is determining if a file has been manipulated or not; or in other words evaluating the integrity of the file. Since it is extremely unlikely that the same *private key* can be obtained from the same fixed Laplacian noise signal and from two different speeches, this characteristic can be used to detect when a speech signal has been altered (even with modification of only few samples). For example, in the case of audio-forensics, a conversation record can be given as evidence, and in order to guarantee the chain of custody, it is necessary to ensure that its content is not altered. At the beginning, the original content is delivered (by *Alice*) jointly with the *public key* (i.e. the hash function). To verify the integrity of the file, the *public key* is calculated again and compared to the original one. If the speech file has not changed, the two *public keys* are equal. But, if *Alice* or another subject manipulates the speech file (even with a slight content-modification like mute attack of a small frame), the modified file will produce a different *public key* (hash function), because it is extremely unlikely to obtain the same hash function from two different speech signals with the same fixed Laplacian noise. Therefore, it is concluded that the signal has been manipulated. Only, if two private keys are exactly equal, the audio recording is considered as authentic.

The above applications are feasible because the current scrambling scheme satisfies the following conditions:

a. There is a unique relationship between the speech message and the scrambled message, for a specific Laplacian noise signal. Therefore, there is a unique identifying value (*key*) for each pair of signals.
b. For the first scenario, mapping process between the speech signal and the scrambled speech signal is dynamic, because the Laplacian noise signal is generated each time.
c. For the second scenario, only the correct Laplacian noise signal gives the correct deciphered message.
d. The lengths of the key, the speech signal, and the Laplacian noise signal are equal. Therefore, the *keyspace*, the *secret space* and the *Laplacian space* have the same size.
e. All keys are equally likely.
f. All Laplacian noise signals are equally likely.

5. **Conclusion.** The speech scrambling scheme proposed by Ballesteros, Renza and Camacho was analyzed in terms of security. This scheme can have two application scenarios: scrambling with *private key*, and scrambling with one *private* and one *public key*. Both

scenarios can be used for covert communication but the second one can be used for forensic authenticity, too.

Since the aim of covert communication is protect the privacy of content, its security testing was focused on *known plaintext*, *brute force*, and *known ciphertext* attacks. The analyzed scheme overcomes the *known plaintext* attack because the mapping between the places of the speech signal and the scrambled speech signal is dynamic. In terms of *brute force* attack, the *keyspace* is large enough to require several years to test all available keys. Finally, the *known ciphertext* attack was overcome because several deciphered signals may have natural behavior and none of them can be discarded through typical signal processing analysis.

On the other hand, in the second scenario only the *public key* is transmitted between two parts of communication. The Laplacian noise signal (i.e. target signal of the imitation process) is fixed in each device. If *Eve* intercepts the *public key* but she does not know the Laplacian noise signal of the device, she cannot reveal the secret content.

In Summary, in both proposed scenarios of the analyzed scrambling scheme, the system overcomes the security tests and therefore content is protected.

Acknowledgment. This work is supported by the "Universidad Militar Nueva Granada - Vicerrectoría de Investigaciones" under the grant IMP-ING-2136 de 2016.

REFERENCES

[1] W. Q. Yan, W. G. Fu, and M. S. Kankanhalli, Progressive audio scrambling in compressed domain, *Multimedia, IEEE Transactions on*, vol. 10, no. 6, pp. 960–968, 2008.

[2] L. Zeng, X. Zhang, L. Chen, Z. Fan, and Y. Wang, Scrambling-based speech encryption via compressed sensing, *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 1–12, 2012.

[3] N. Augustine, S. N. George, and P. Deepthi, Sparse representation based audio scrambling using cellular automata, in *Electronics, Computing and Communication Technologies (IEEE CONECCT), 2014 IEEE International Conference on*, pp. 1–5, IEEE, 2014.

[4] Z. Chun-yu, Z. Wen-xiang, and W. Shao-wei, Comparison of two kinds of image scrambling methods based on lsb steganalysis, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 4, 2015.

[5] L. Yuan, P. Korshunov, and T. Ebrahimi, Secure jpeg scrambling enabling privacy in photo sharing, in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 4, pp. 1–6, IEEE, 2015.

[6] H. Sohn, W. De Neve, and Y. M. Ro, Privacy protection in video surveillance systems: analysis of subband-adaptive scrambling in jpeg xr, *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 2, pp. 170–177, 2011.

[7] F. Dufaux and T. Ebrahimi, Scrambling for privacy protection in video surveillance systems, *Circuits and systems for video technology, IEEE Transactions on*, vol. 18, no. 8, pp. 1168–1174, 2008.

[8] F. Dai, D. Zhang, and J. Li, Encoder/decoder for privacy protection video with privacy region detection and scrambling, in *Advances in Multimedia Modeling*, pp. 525–527, Springer, 2013.

[9] S. M. Alwahbani and E. Bashier, Speech scrambling based on chaotic maps and one time pad, in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, pp. 128–133, IEEE, 2013.

[10] E. M. Elshamy, E.-S. M. El-Rabaie, O. S. Faragallah, O. A. Elshakankiry, F. E. A. El-Samie, H. S. El-Sayed, and S. El-Zoghdy, Efficient audio cryptosystem based on chaotic maps and double random phase encoding, *International Journal of Speech Technology*, vol. 18, no. 4, pp. 619–631, 2015.

[11] C. Guo, C.-C. Chang, and C.-Y. Sun, Chaotic maps-based mutual authentication and key agreement using smart cards for wireless communications, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 4, 2013.

[12] A. Madain, A. L. A. Dalhoum, H. Hiary, A. Ortega, and M. Alfonseca, Audio scrambling technique based on cellular automata, *Multimedia tools and applications*, vol. 71, no. 3, pp. 1803–1822, 2014.

[13] D. M. Ballesteros L, D. Renza, and S. Camacho, An unconditionally secure speech scrambling scheme based on an imitation process to a gaussian noise signal, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 2, 2016.

[14] D. M. Ballesteros L and J. M. Moreno A, Speech scrambling based on imitation of a target speech signal with non-confidential content, *Circuits, Systems, and Signal Processing*, vol. 33, pp. 3475–3498, 2014.

[15] F. A. Petitcolas, Kerckhoffs principle, in *Encyclopedia of cryptography and security*, pp. 675–675, Springer, 2011.

[16] F. Mirza, Block ciphers and cryptanalysis, *Royal Holloway University of London, Department of Mathematics, England*, 1998.

[17] N. S. Kulkarni, B. Raman, and I. Gupta, Multimedia encryption: a brief overview, in *Recent advances in multimedia signal processing and communications*, pp. 417–449, Springer, 2009.

[18] X. Jin, K. Guo, C. Song, X. Li, G. Zhao, J. Luo, Y. Li, Y. Chen, Y. Liu, and H. Wang, Private video foreground extraction through chaotic mapping based encryption in the cloud, in *MultiMedia Modeling*, pp. 562–573, Springer, 2016.

[19] C. Li and K.-T. Lo, Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, *Signal processing*, vol. 91, no. 4, pp. 949–954, 2011.

[20] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, On the security of permutation-only image encryption schemes, *Information Forensics and Security, IEEE Transactions on*, vol. 11, no. 2, pp. 235–246, 2016.

[21] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, *Signal Processing: Image Communication*, vol. 23, no. 3, pp. 212–223, 2008.

[22] H. Ghasemzadeh, H. Mehrara, and M. T. Khas, Cipher-text only attack on hopping window time domain scramblers, in *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*, pp. 194–199, IEEE, 2014.

[23] T. Lotter, C. Benien, and P. Vary, Multichannel speech enhancement using bayesian spectral amplitude estimation, in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1, pp. I–880, IEEE, 2003.

[24] J. W. Shin, J.-H. Chang, and N. S. Kim, Statistical modeling of speech signals based on generalized gamma distribution, *Signal Processing Letters, IEEE*, vol. 12, no. 3, pp. 258–261, 2005.

[25] S. Gazor and W. Zhang, Speech probability distribution, *Signal Processing Letters, IEEE*, vol. 10, no. 7, pp. 204–207, 2003.

[26] S. Vanambathina and T. K. Kumar, Speech enhancement by bayesian estimation of clean speech modeled as super gaussian given a priori knowledge of phase, *Speech Communication*, vol. 77, no. C, pp. 8–27, 2016.

[27] M. A. Akhaee, N. K. Kalantari, and F. Marvasti, Robust audio and speech watermarking using gaussian and laplacian modeling, *Signal processing*, vol. 90, no. 8, pp. 2487–2497, 2010.

[28] D. M. Ballesteros L and J. M. Moreno A, On the ability of adaptation of speech signals and data hiding, *Expert Systems with Applications*, vol. 39, no. 16, pp. 12574–12579, 2012.

[29] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption, *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 8, pp. 3303–3327, 2012.