Journal of Information Hiding and Multimedia Signal Processing
Ubiquitous International ISSN 2073-4212
Volume **9**, Number **4**, July **2018**

Evolutionary Algorithm for Speech Scrambling based on Asexual Reproduction

Dora M. Ballesteros L., Alejandro Sandoval and Diego Renza

Telecommunications Engineering
Universidad Militar Nueva Granada
Carrera 11 101-80, Bogotá-Colombia
dora.ballesteros@unimilitar.edu.co, u1400864@unimilitar.edu.co, diego.renza@unimilitar.edu.co

Received June, 2017; revised February, 2018

Abstract. *In this paper, it is proposed an evolutionary algorithm based on asexual reproduction to obtain scrambled signals with high disorder level (DS) and therefore very low residual intelligibility. The first parent is a copy of the original speech signal which is divided into several chromosomes (C) with genes of different length (G). The offspring is obtained by a transposition process of the chromosomes. The aim of reproduction is to obtain an offspring with high dissimilarity to the original speech signal, which is measured through the Squared Pearson Correlation Coefficient (SPCC) between the two signals. The fitness function corresponds to an SPCC lower than a fixed threshold. According to several tests it was found the offspring that satisfied the fitness function are scrambled signals without trace of its original content (i.e.* $DS \geq 0.25$*). Unlike other methods of speech scrambling, our proposed algorithm does not need an external key or the adjustment of initial conditions; i.e. the output signal obeys only the fulfilment of the fitness function.*

Keywords: Speech Scrambling, Evolutionary Algorithm, Asexual Reproduction, Transposition, Level of disorder.

---

1. **Introduction.** Speech signals are one of the most used signals in communication. With recent technologies, speech signals are widely transmitted over the internet and mobile telephony. Most of these communications belong to non-confidential messages and then users do not care about confidentiality. However, in other cases, information is classified as confidential and therefore it is necessary to preserve its privacy. In the context of audio protection, privacy of the content can be kept in two ways: tampering the content of the audio or hiding the content into a host signal. In the first case, solutions are known as scrambling [1]; in the second case, as steganography [2]. If the speech signal is scrambled and then the result is hidden into a host signal, the final system will work with at least two levels of security.

In steganography, the transmitted signal is known as the stego signal, which must be akin to the host signal and dissimilar to the secret (speech) message. The purpose of any steganography system is to preserve the secret content without generating suspiciousness about its existence. Classical methods of speech steganography are based on LSB (Least Significant Bit) substitution [3], shift and spread spectrum [4], frequency/wavelet masking [5] and Quantization Index Modulation [6]; non-classical methods include evolutionary algorithms [7]. All of them must guarantee the imperceptibility of the secret content.

In scrambling systems, an eavesdropper knows that the transmitted signal (i.e. the scrambled signal) has been manipulated, but he does not know the secret content. In the initial proposals, the speech signal was permuted in time or frequency domain based on pseudo-random sequences, which are known as the *key* of the system [8, 9, 10]. However, the security of these keys is easily broken with modern digital systems. Therefore, new proposals of speech scrambling are focused on key generation with the purpose of obtaining more secure systems based on chaotic maps [11, 12], cellular automata [13, 14, 15] or bio-inspired principles [16, 17, 18]. The disadvantage in key generation based on chaotic maps is the relationship between the scrambling degree and the initial conditions of the algorithm. Cellular automata can work directly over the speech signal [14] or over the compressive sensed audio signal [13, 15]. If the system works directly over the speech signal, the computational cost is less complex than in compressive sensed audio, but, the storage and transmission of initial state matrix is required. In the case of bio-inspired principles, the key is not an input of the system and it is created in-situ through an imitation process between the secret message and a target speech signal. Target signals can be non-sensitive speech signals [16], or noise signals [17, 18]. In both cases, the system is unconditionally secure according to Shannons theorem. As a disadvantage, in the first case the system needs a large database of non-sensitive speech signals, and in the second case it needs the creation in-situ of a super-Gaussian noise signal. In addition, the scrambling process obeys a sorting process which is a complex task for hardware devices. According to the above discussion, a solution for scrambling speech signals with high security is still a challenge, regardless of external keys and initial conditions, as well as simple operations that allow its hardware implementation.

On the other hand, evolutionary algorithms are well-known methods for searching that does not respond to a systematic procedure. The solution is only better compared to others, but, the best or optimal solution is not known. This kind of method is useful in problems where it is difficult to verify the optimal candidate or when "good" solution is enough. In terms of reproduction, this kind of methods can be sexual or asexual. Unlike sexual reproduction, in asexual reproduction there is only one parent and then the offspring is obtained from its material.

Taking into account the problem of relocating the samples of a speech signal without using an external key and without dependence on initial conditions adjustment, the use of an evolutionary algorithm is proposed as the core of the proposed scrambling method. Asexual reproduction was chosen because this kind of reproduction does not change the genetic content of the organism, which means, in our context, that histograms of the speech signal and the scrambled speech signal are the same. Consequently, the scrambling process is reversible and then the authorized person may recover the original content of the signal.

2. **Evolutionary Algorithms with Sexual and Asexual Reproduction.** Sexual reproduction is a well-known kind of reproduction used in evolutionary algorithms that satisfies the following conditions:

- It involves two individual organisms (parents). Every parent has several chromosomes everyone with some genes.
- The next generation (*offspring*) is obtained from the parents with at least one of the reproduction mechanisms (e.g. crossover, mutation, transposition).
- The *offspring* has material from each parent.
- The reproduction is an iterative process applied until a goal is reached, known as the *fitness function*.
- Each *offspring* is evaluated in terms of the *fitness function*.

- If the *fitness function* is not satisfied with the current *offspring*, a new *offspring* is obtained.
- The iterative process stops when the highest number of iterations has been reached or when the *fitness function* is satisfied.
- The *offspring* that satisfies the *fitness function* is not necessarily the best solution of the search space, but it is a "good" solution.

Although sexual reproduction is the classical way to obtain *offspring*, asexual reproduction has been used in the last years as an alternative solution in classical heuristic problems like optimization function of two variables, vehicle routing problem and job shop scheduling problem [19, 20, 21].

In the case of asexual reproduction, the following conditions are satisfied:

- There is only one progenitor. This parent is composed by chromosomes (C) and genes (G).
- The *offspring* is obtained with genetic material from only one progenitor.
- The most useful mechanism of reproduction is transposition, which is explained in Figure 1.

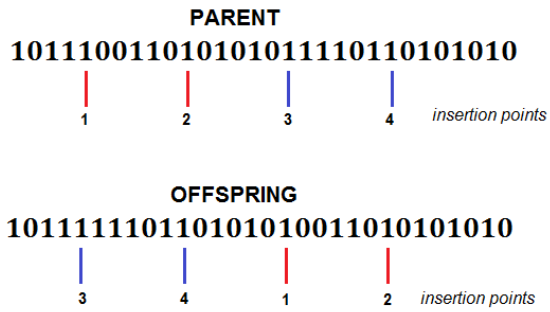


FIGURE 1. Example of the transposition mechanism.

In Figure 1, the parent is composed by thirty bits which are reorganized according to four insertion points. Insertion points 1 and 2 delimit the first block to be transposed, and insertion points 3 and 4 delimit the second one. Then, the first block is located at the position of the second block and vice versa. Therefore, in asexual reproduction, the *offspring* is obtained only from one parent. In this binary example, it is noticeable that both parent and *offspring* have the same quantity of bits 1 (i.e. 18 bits equal in this example).

Summarizing, the main difference between sexual and asexual reproduction is the way to obtain the *offspring*, because in asexual reproduction, to obtain a child, only one parent is required.

3. **Proposed Algorithm.** We propose an evolutionary algorithm that uses asexual reproduction for tampering the signal. The entire system is composed of two stages: the scrambling process (at the transmitter module) and the descrambling process (at the receiver module). With the purpose of recovering the secret message, a secret key generated by the system is transmitted between the two parts of the communication.

3.1. **Scrambling Stage.** The aim of this stage is to tamper the content of the secret message. At the end, the system has two outputs: the scrambled signal and the *key*. These outputs must be transmitted by two different channels in order to preserve the privacy of the secret content. For example, suppose that the *key* is transmitted by e-mail and the scrambled signal is published in a public web site. Although an eavesdropper intercepts the *key*, if he does not know which scrambled signal it is related to, then he cannot reveal the secret content. Conversely, having the scrambled signal without the *key*, the descrambling process is not feasible.

Algorithm 1 shows the pseudo-code of the proposed evolutionary algorithm.

Algorithm 1 Pseudo-code of the proposed scrambling scheme

Inputs: Speech signal (S), Total number of samples (m), Number of genes by chromosome (G) .
Outputs: Scrambled signal (Sa), Key.
1: **procedure** SCRAMBLING
2: Read S, m, G
// Calculate the number of chromosomes (C)
3: $C \leftarrow m/G$
// Calculate the highest number of iterations (N)
4: $N \leftarrow \lfloor C/2 \rfloor$
// Initialize first parent (P) and iteration (it)
5: $P \leftarrow S$
6: $it \leftarrow 1$
// Separate P in C chromosomes of length G
7: P *is a set of* $\{P(1), P(2), \ldots P(C)\}$
//Select two numbers in the range $\{1, 2, \ldots, C\} \notin Key$, and add them to the key
8: *a, b are the selected numbers*
9: $key(it) \leftarrow [a \;\; b]$
// Generate the offspring (Os)
10: *Os is obtained from P by transposing P(a) with P(b)*
// Evaluate the fitness function
11: $value1 \leftarrow SPCC(S, Os)$
12: **if** $it > N$ OR $value1 < 0.001$ **then**
13: $Sa \leftarrow Os$
14: **else**
15: $P \leftarrow Os$
16: $it \leftarrow it + 1$
17: Go to 7
18: **end if**
19: **end procedure**

The algorithm is explained, as follows:

1. Create the original parent: the first parent, P, of the iterative process is the same speech signal, S, which contains m samples.
2. Divide the parent into *chromosomes* and *genes*: the total number of *chromosomes* (C) is equal to dividing m by G.
3. Assign to every chromosome the number related to its place.
4. With a random generator select two numbers in the range of 1 to C. For example, if there are 100 chromosomes, the selected numbers must be in the range 1 to 100. These numbers are locked for future iterations. It means each chromosome can be selected only once. The above numbers are placed in the *key*, each one in one column of the *key*.
5. Transpose the selected chromosomes. For example, if the selected chromosomes are 2 and 80, the genes of the chromosome 80 will be placed in the position of the genes of the chromosome 2, and vice versa. The result is the *offspring*.
6. Evaluate the *fitness function* with the current *offspring*. In our proposal it is defined, according to:

$$fitnessfunction: \quad SPCC(S, Os) < 0.001 \tag{1}$$

Where S is the original speech signal and Os is the current *offspring*.

7. If the *fitness function* is satisfied, the current *offspring* is the scrambled signal; otherwise the steps 4 to 6 are repeated again. However, a stop condition related to the total number of iterations is included to prevent the system from working indefinitely.

If an *offspring* satisfies the *fitness function*, it means that the original content of the speech signal has been highly tampered, and therefore the iterative process stops. At the end of the iteration process, the last *offspring* corresponds to the scrambled speech signal. It is worth noting that there are many scrambled speech signals with non-intelligible content that could be obtained with the proposed evolutionary algorithm. Then, the "best" scrambled speech signal is not obtained, but the result is a signal without a trace of the original content.

In terms of *key* size, our proposed algorithm provides *keys* with different sizes. The total number of columns is fixed in two (i.e. the selected chromosomes by iteration), but the total number of rows depends on the number of iterations, which varies from experiment to experiment.

3.2. **Descrambling Stage.** The aim of this stage is to recover the secret message by using the scrambled speech signal and the *key*. If the user does not have the *key*, he cannot recover the secret message even if he knows the algorithm to scramble the speech signal. For an eavesdropper of communication, the size of the *key* is an unknown parameter, even with the same secret message because the selection of the pairs of chromosomes is a random process. Algorithm 2 shows the pseudo-code of the proposed descrambling scheme.

The steps to descramble the secret message are as follows:

1. Read the scrambled speech signal (Sa), the key, and the number of genes by chromosome (G).
2. In a similar way of the scrambling scheme, it is necessary to separate the scrambled speech signal into C chromosomes, each of G genes.
3. Transpose the chromosomes according to the *key*. If the *key* has N rows, there are N pair of chromosomes that must be transposed.
4. Repeat the previous step for every row of the *key*. At the end, the descrambled speech signal is obtained (Sr).

Algorithm 2 Pseudo-code of proposed descrambling scheme.

Inputs: Scrambled signal (Sa), Number of genes by chromosome (G), *Key*.
Outputs: Descrambled speech signal (Sr).

procedure DESCRAMBLING
  Read Sa, G, and the *key*
  // Initialize the descrambled speech signal (Sr)
  $Sr \leftarrow Sa$
  // Calculate the number of chromosomes (C)
  $C \leftarrow m/G$
  // Separate Sa in C chromosomes of length G
  Sa *is a set of* $\{Sa(1), Sa(2), \ldots Sa(C)\}$
  // Calculate the total number of transpositions (N)
  $N \leftarrow$ *number of rows in the key*
  Transpose the genes in the speech signal
  for $i = 1$ to N **do**
    $a \leftarrow key(i, 1)$
    $b \leftarrow key(i, 2)$
  // Exchange blocks $Sr(a)$ and $Sr(b)$
    Update Sr *by transposing* $Sr(a)$ *with* $Sr(b)$
  end for
end procedure

Our proposal is completely reversible in terms of the secret message; it means that the descrambled signal (Sr) is equal to the original speech signal (S).

3.3. **Example of the Scrambling Process.** Consider a secret message, S, with the following data:
$S = [4,\ 5,\ 7,\ 8,\ 9,\ 7,\ 8,\ 5,\ 6,\ 3,\ 2,\ -1,\ -2,\ -5,\ -4]$.

a. The first step is to assign the speech signal to the first parent of the algorithm. Then, if P is the first parent,
$P = S = [4,\ 5,\ 7,\ 8,\ 9,\ 7,\ 6,\ 4,\ 3,\ 3,\ 2,\ -1,\ -2,\ -5,\ -4]$.
b. The second step is to obtain the number of chromosomes (C) of the parent. Divide the parent in C parts. For the example, the value of C is five.
c. The third step is to select a pair of chromosomes. Since the total number of chromosomes is five, the selected numbers are among 1 to 5; for example, 2 and 5. These values are placed into the key.
d. In the fourth step, the chromosomes 2 and 5 are placed in the new positions. The result is the first *offspring*.
e. Finally, the *fitness function* is evaluated. If the criterion is satisfied the process finishes; otherwise, the new parent is the most recent *offspring* and the steps (c) to (d) are repeated again.

Figure 2 shows the results of the above steps.

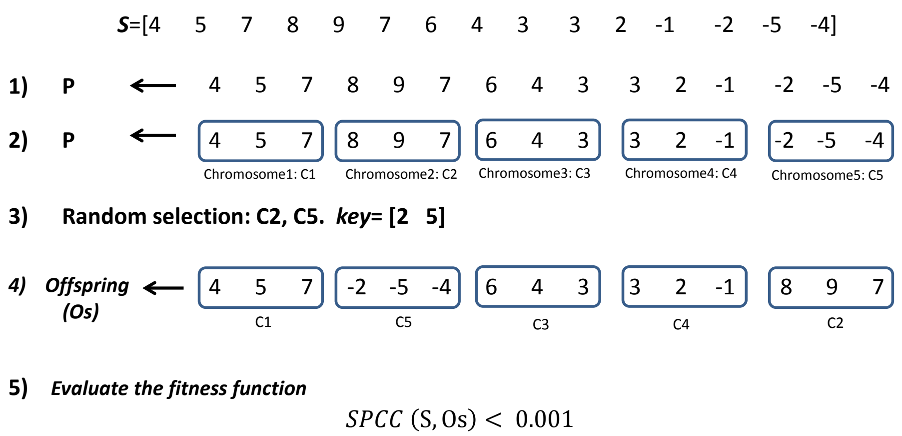


Figure 2. Example of the scrambling process.

To recover the original message, the receiver needs to know the *key*, the G size, and the scrambled speech signal.

4. **Qualitative Comparison with other Methods.** In this section a qualitative comparison of our method with other methods is carried out (Table 1). A specific comparison in terms of the accuracy of the tampering process is presented in Section 5.

It should be noted that in terms of mathematical operations, our proposal uses simpler operations compared to related methods. Also, in terms of initial conditions, our proposal has very low dependency on initial conditions. In contrast, in some works the quality of the scrambled signal strongly depends on the adjustment of initial conditions [14] and the fulfilment of requirements [16]. In addition, one of the advantages of our proposal is that the *key* size is not fixed and it varies inter signals and intra signals. This gives an additional security level for the system.

Table 1. Comparison of speech scrambling methods.

Reference	[14]	[16]	[17, 18]	Our
Technique	Cellular automata	Based on imitation of a target speech signal	Based on imitation of a super-Gaussian signal	Evolutionary algorithm, asexual reproduction
Initial conditions	NOG (number of generations), neighbourhood types, transition function and rule	Database of target speech signals	Seed of the noise generator	Seed of the random generator of indices
Mathematical Operations	Generation of indices, mapping	Sorting of the signals, mapping	Generation of noise signal, sorting of the signals, mapping	Random generation of indices, interchange of genes
Output	Noise signal	Adapted signal very similar to the target signal	Gaussian noise signal	Noise signal
Key	It is static (i.e. the same result for the same input conditions).	It is static (i.e. fixed length with the same result every time)	It is partially dynamic (i.e. fixed length with different result for every time)	It is dynamic (i.e. variable length and different result every time)

5. **Implementation and Evaluation of the Method.** The purpose of this phase is to validate the proposed scheme in terms of the accuracy of the tampering process to obtain scrambled signals without a trace of the original content. We work with a database with the following characteristics:

- Number of recordings: 200
- Gender of speaker: female and male
- Length of the recordings: five seconds
- Language: English

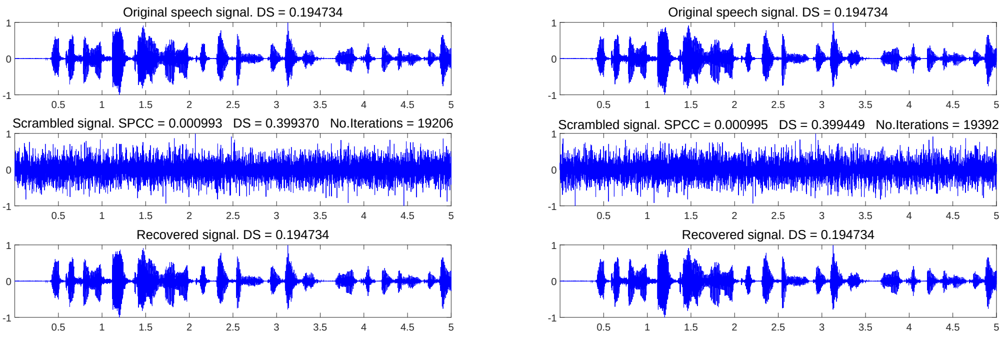
Our method is compared with [17] in order to evaluate its strengths and weaknesses. The reference method is selected because it has demonstrated high performance in terms of residual intelligibility and security. For the reference method, 200 scrambled signals are obtained. For our method, we have fixed the value of G in four values: 1, 2, 5, and 10. For every value of G, 200 scrambled signals are obtained. In total, the validation phase is composed by 1000 scrambled signals.

5.1. **Evaluation Measurements.** To measure the accuracy of the tampering process, we have selected the parameter DS (Disorder Level of the scrambled signal), which is calculated through the Equation 2.

$$DS = \frac{\sum_{i=2}^{m-1} \sqrt{|Os_i - Os_{i+1}| + |Os_i - Os_{i-1}|}}{m-2} \tag{2}$$

Where Os_i is the $i-th$ sample of the *offspring* audio signal, and m is the total number of samples in this signal. Since the central sample is compared with its right and left neighbour, the total number of central samples is $m-2$. DS is a scalar and it is a measurement of the level of disorder of a signal; its highest value is $\sqrt{Vpp}$ where Vpp is the peak-to-peak amplitude of the signal. If the central gene (or sample of the signal) is similar in value to its neighbours, then, DS is low (i.e. close to 0). If the signal is in the range $[-1\ 1]$ then DS is in the range $[0\ 2]$. The higher the value of DS, the lower the intelligibility of the signal is.

5.2. **Preliminary Results.** With the purpose of illustrating the performance of our scheme, we present two examples (Figure 3) from a real speech signal using the proposed evolutionary algorithm with asexual reproduction, using $G = 1$. The algorithm runs twice with the same input signal, in order to evaluate the repeatability of the algorithm.



(a) Example 1. (b) Example 2.

FIGURE 3. Examples of two scrambled signals from the same original speech signal.

The speech signal in Figure 3 has 5-seconds and a frequency sampling of 8 kHz (i.e. $m = 40k$). Then, the maximum number of iterations, N, is 20 K (i.e. $N = m/(2 \times G)$).

According to the preliminary results (Figure 3), the proposed evolutionary algorithm with asexual reproduction for speech scrambling allows to tamper the speech signal without a trace of its original content. In both cases, the number of iterations to reach the fitness function is lower than 20.000, and the DS is very similar between them. It is worth noting that the number of iterations varies between experiments and therefore, the size and content of their keys is not the same.

5.3. **Results and Discussion.** Several tests were carried out (1000 in total) with the purpose of validating the quality of the scrambled signal and the percentage of re-located samples. Two hundred speech signals were selected to be scrambled using two methods: ours with four different values of G (i.e. 1, 2, 5 and 10) and a reference method corresponding to the proposal in [17]. In each case, 200 simulations were carried out.

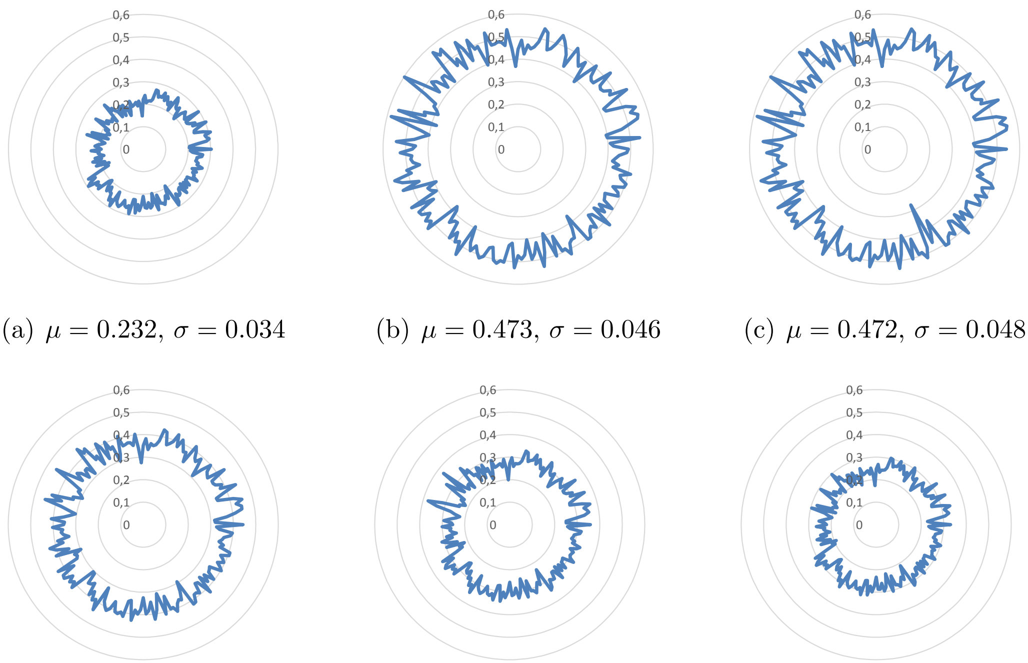
5.3.1. *Quality of the Scrambled Signal.* The aim of these tests is to calculate the value of DS of the scrambled signals. The more distant the value is to zero, the lower the trace is of the original content; otherwise, the signal keeps sounds of the original speech signal.

Having as purpose summarizing the results, each set (200 hundred simulations) is drawn by means of a radar chart. The angular separation between data corresponds to 360/200 degrees. Figure 4 shows the results as follows: Figure 4(a) corresponds to the radar plot of the DS values for the original speech signal; Figure 4(b) corresponds to the DS values for the reference method, Figures 4(c) to 4(f) are the radar plots of the DS values for the proposed method, with G equal to 1, 2, 5 and 10, respectively. In the plots, μ represents the average of the set; σ represents its standard deviation.

According to the results of Figure 4, the best results (i.e. higher values of DS) correspond to the reference method and our proposal with $G = 1$. It is worth noting that the average and the standard deviation are very similar in both cases. On the other hand, in our proposal if G increases, the quality of the scrambled signal decreases.

5.3.2. *Percentage of Re-located Samples.* The second part of the validation stage corresponds to the number of re-located samples as a percentage of the total number of samples of the signal. The objective is to calculate the percentage of samples that are re-located, but not the execution time. In terms of efficiency, better results are reached if the number of relocations is lower for the same quality of the scrambled signal. In a similar way to Section 5.3.1, radar plots are used in order to summarize the results. Again, each angular separation is equal to 360/200 degrees. Figure 5 shows the results of the 1000 tests.

According to Figure 5, the average of re-located samples is very similar among the results of our proposal. This value ($\approx 97\%$) is lower than the one obtained with the reference method (i.e. $\approx 100\%$). Consequently, there are two conclusions of these results: firstly, with our proposal the percentage of re-located samples does not depend on the value of G (verified for the range 1 to 10); secondly, the quantity of re-located samples in our proposal is lower than in the reference method, even for the same value of desired DS.



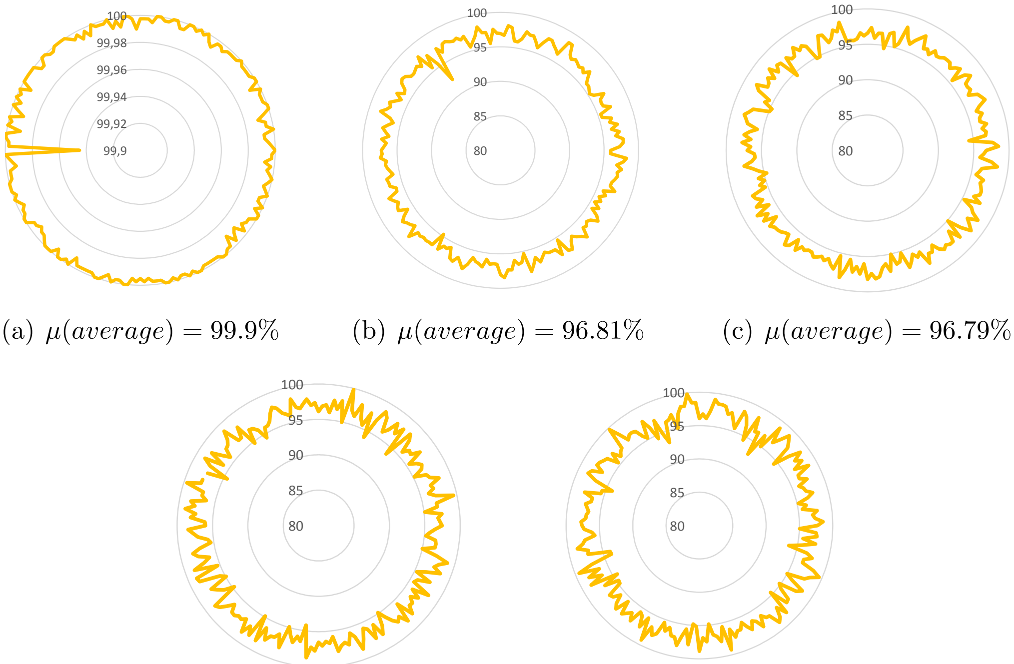
(a) $\mu = 0.232$, $\sigma = 0.034$ (b) $\mu = 0.473$, $\sigma = 0.046$ (c) $\mu = 0.472$, $\sigma = 0.048$

(d) $\mu = 0.374$, $\sigma = 0.037$ (e) $\mu = 0.289$, $\sigma = 0.035$ (f) $\mu = 0.260$, $\sigma = 0.033$

FIGURE 4. Results in terms of DS: a) Original speech signal, b) Reference method, c-f) Our proposal with G=1, 2, 5 and 10, respectively.

5.3.3. *Quality of the Descrambled Signal.* The proposed method of speech scrambling is completely reversible. It means, the descrambled signal is equal (bit by bit) to the original speech signal. An example of the descrambling signal was shown in Section 5.2.

6. **Conclusions.** In this work, we develop a speech scrambling method that uses an evolutionary algorithm with asexual reproduction, to obtain an *offspring* which is very dissimilar to the secret message in terms of its content. It was demonstrated that with the proposed *fitness function* (i.e. value of $SPCC$ lower than 0.001), the reproduction type (i.e. asexual) and the selected reproduction mechanism (i.e. transposition), the method is able to find an *offspring* that tampers the content of the original speech signal. The scrambled speech signal is the first *offspring* that satisfies the *fitness function* and its



(a) $\mu(average) = 99.9\%$ (b) $\mu(average) = 96.81\%$ (c) $\mu(average) = 96.79\%$

(d) $\mu(average) = 96.93\%$ (e) $\mu(average) = 96.88\%$

FIGURE 5. Results in terms of percentage of re-located samples: a) Reference method, b-e) Our proposal with G=1, 2, 5 and 10, respectively.

characteristics are: it looks like a noise signal; its value of DS (level of disorder) is higher than the obtained from the original speech signal. Further, the process is reversible and therefore the recovered speech signal is equal to the original one.

In terms of the *key*, we work with an adaptive key-generator which responds to the conditions of the evolutionary algorithm with asexual reproduction. Its size and values are unpredictable and they change between simulations. This property of the key is useful in terms of security (very poor predictability of the key). In terms of re-located samples, our proposal needs a lower number of transpositions (permutations) than other related methods for the same desired value of DS.

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

REFERENCES

[1] B. G. Lee and S. C. Kim, *Scrambling techniques for digital transmission*. Springer Science & Business Media, 2012.

[2] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2012, no. 1, p. 25, 2012.

[3] J. Liu, K. Zhou, and H. Tian, "Least-significant-digit steganography in low bitrate speech," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 1133–1137, IEEE, 2012.

[4] T. Rabie and D. Guerchi, "Spectral magnitude speech steganography," *International Journal of Computer Applications*, vol. 116, no. 5, 2015.

[5] D. M. Ballesteros L and J. M. Moreno A, "Highly transparent steganography model of speech signals using efficient wavelet masking," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9141–9149, 2012.

[6] H. Tian, J. Liu, and S. Li, "Improving security of quantization-index-modulation steganography in low bit-rate speech streams," *Multimedia systems*, vol. 20, no. 2, pp. 143–154, 2014.

[7] K. Bhowal, D. Bhattacharyya, A. J. Pal, and T.-H. Kim, "A ga based audio steganography with enhanced security," *Telecommunication Systems*, vol. 52, no. 4, pp. 2197–2204, 2013.

[8] A. Matsunaga, K. Koga, and M. Ohkawa, "An analog speech scrambling system using the fft technique with high-level security," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 540–547, 1989.

[9] F. Ma, J. Cheng, and Y. Wang, "Wavelet transform-based analogue speech scrambling scheme," *Electronics Letters*, vol. 32, no. 8, pp. 719–721, 1996.

[10] R. W. Woo and C. Leung, "A new key generation method for frequency-domain speech scramblers," *IEEE transactions on communications*, vol. 45, no. 7, pp. 749–752, 1997.

[11] S. B. Sadkhan and H. Ali, "A proposed speech scrambling based on hybrid chaotic key generators," in *Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on*, pp. 1–6, IEEE, 2016.

[12] R. S. Mohammed and S. B. Sadkhan, "Speech scrambler based on proposed random chaotic maps," in *Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on*, pp. 1–6, IEEE, 2016.

[13] S. N. George, N. Augustine, and D. P. Pattathil, "Audio security through compressive sampling and cellular automata," *Multimedia Tools and Applications*, vol. 74, no. 23, pp. 10393–10417, 2015.

[14] 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.

[15] 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.

[16] 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.

[17] 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," *J. Inf. Hiding Multimedia Sig. Process*, vol. 7, no. 2, pp. 233–242, 2016.

[18] D. M. Ballesteros L, D. Renza, and S. Camacho, "Security analysis of the speech scrambling method based on imitation of a super-gaussian signal," *J. Inf. Hiding Multimedia Sig. Process*, vol. 8, no. 1, pp. 156–167, 2017.

[19] J. Cantó, S. Curiel, and E. Martínez-Gómez, "A simple algorithm for optimization and model fitting: Aga (asexual genetic algorithm)," *Astronomy & Astrophysics*, vol. 501, no. 3, pp. 1259–1268, 2009.

[20] M. Amirghasemi and R. Zamani, "An effective asexual genetic algorithm for solving the job shop scheduling problem," *Computers & Industrial Engineering*, vol. 83, pp. 123–138, 2015.

[21] A. Tretyakova and F. Seredynski, "A novel genetic algorithm with asexual reproduction for the maximum lifetime coverage problem in wireless sensor networks," in *The Third International Conference on Advanced Communications and Computation (INFOCOMP-2013)*, pp. 87–93, 2013.