

AKADÉMIAI KIADÓ

Pollack Periodica •
An International Journal
for Engineering and
Information Sciences

18 (2023) 1, 161-166

DOI:

[10.1556/606.2022.00695](https://doi.org/10.1556/606.2022.00695)

© 2022 Akadémiai Kiadó, Budapest

ORIGINAL RESEARCH
PAPER



Time domain speech scrambler based on particle swarm optimization

Zahraa A. Hasan^{1*}, Suha M. Hadi¹ and Waleed A. Mahmoud²

¹ Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq

² Department of Electronic and Communication, College of Engineering, Uruk University, Baghdad, Iraq

Received: July 10, 2022 • Revised manuscript received: August 14, 2022 • Accepted: August 16, 2022

Published online: October 25, 2022

ABSTRACT

Speech scrambler is used to transform clear speech into an unintelligible signal to prevent eavesdropping. The speech scrambling algorithm involves the permutation of speech segments depending on a specific permutation matrix which may be fixed or dynamic during encryption. A fixed permutation matrix is easy to break and has given high residual intelligibility in the scrambled signal. The proposed scheme used a particle swarm optimization algorithm to generate a dynamic permutation array that can attain a high degree of security. The outcome of the scrambled speech signal does not have any residual intelligibility, and the quality of the descrambled speech is extremely satisfying, with zero mean squared error.

KEYWORDS

speech scrambling, particle swarm optimization, speech segmentation, analog scrambling

1. INTRODUCTION

The importance of secure speech communication in commercial, civil, and military communication systems cannot be overstated. As a result, speech scrambling has drawn a great research interest in speech security. Speech scrambling is achieved by altering the speech signals to make them incomprehensible to eavesdroppers [1]. Speech scrambling methods may be classified as analog and digital. In analog scramblers, the permutation of speech segments in frequency, time, or time-frequency domains, as well as the permutation of coefficients for each speech segment, is used. These scramblers are popular and widely used. However, these time-frequency algorithms are not providing enough security against cryptanalysis since, in these scramblers, the permuted elements are not large enough to provide a sufficient number of variant permutations because of processing delays and hardware limitations [2].

An analog voice privacy system was invented in 1986, it was the first digital time-frequency scrambler. The value of the residual intelligibility of the proposed system was satisfactory [3]. While in 1989, a full-duplex analog speech scrambling system was proposed using the Fast Fourier Transform (FFT) technique. The scrambling algorithm was based on the rearrangement of the FFT coefficients. According to the simulation's findings, the speech that was scrambled did not retain any intelligibility, while the quality of the unscrambled speech was satisfactory [4]. Later in 2017, a random permutation was presented by Jayakumari using chaotic mapping and pseudo-random binary scrambling [5]. In 2020, Raheema, Sadkhan-Smieeee, and Satar developed an improved method for scrambling speech based on chaotic signals [6]. After a few years, in 2021, Zghair and Sadkhan used the key encrypted by Quadrature Amplitude Modulation Orthogonal Frequency Division Multiplexing (QAM OFDM) to enhance the performance of Bits Error Rate (BER) at the receiving side. This

*Corresponding author.
E-mail: phd202030560@iips.icci.edu.iq

makes the BER robust against the effect of the Additive White Gaussian Noise (AWGN) [7]. Although the permutation of segments differs from one scrambler to another, it is based on either a fixed or dynamic permutation matrix. So, the permutation matrix plays an important role in the efficiency and security of a scrambler. In the proposed scheme, the particle swarm optimization algorithm generates the optimum permutation array that attains the highest residual intelligibility in the scrambled speech. This invention aims to improve scrambling techniques that offer high security.

2. SPEECH SCRAMBLING

Speech scrambling algorithms are gaining universal attention to increasing protection in military and civil applications. Scrambling comes from modifying the signal at the transmitting end while the receiver descrambles to recover the signal. Thus, listeners of the scrambled signal in the transmission channel would only hear a noisy garbled version of the original audio signal. The way the signal is scrambled is specified by the designer. Thus a receiver with the specific descrambling code will be able to successfully unscramble the signal [5].

Earlier speech scrambling methods are classified as analog scrambling and digital scrambling.

In “analog” scrambling, signal transmission is carried out digitally, so it is the only real analog operation. At first, incoming speech signals are digitized, processed by an algorithm, converted, and transmitted to the receiver. Then, they are digitized again, inversely processed, and reconverted to an analog signal for reconstruction [8]. The primary step in digital encryption is the process of digitizing the input speech signal that has been received. Following this step, the digitized signal is compressed to produce a bit stream, followed by the encryption and transmission of this bit stream through the transmission channel. Although digital encryption is more secure than its analog counterpart, its implementation is more difficult [9]. In analog scrambling, no speech compression or modem is required for transmission [10]. The original signal is changed by speech security systems because the algorithms used to code the signal are designed so that the coded signal is not similar to the original signal. The encoding algorithm relies on a particular code, which is also referred to as a key. When different keys are used, different coded signals are obtained as a result. Following that, the signal that has been encoded is sent through the communication channel. A decoding algorithm based on the specified key is used at the receiver to recover the modified coded signal. The transmission key is used to generate the inverse, which is the decoding key. In the end, the result is the decoded recovered signal, comparable to the initially recovered signal [11]. The frequency-domain and time-domain methods are common approaches to analog scrambling.

3. PARTICLE SWARM OPTIMIZATION ALGORITHM

In 1995, Kennedy and Eberhart put forward a heuristic global optimization method called particle swarm optimization [12]. It is based on swarm intelligence and developed from the research movement behavior of birds flock. During the food search process, the birds either go together or scatter before locating where they can find the food. Moreover, there is always a bird in the swarm that can track and smell the food very well while the other birds are searching for food from one location to another, that is, the bird has the better food resource information since it is perceptible of the place where the food can be found. During food searching, the birds share good information, and because of that, the birds will flock to the location where food is placed. In the particle swarm optimization algorithm, the solution swarm is compared to the bird swarm. The solution swarm is developed when the bird moves from one location to another, and good information is considered the optimist solution [13].

The representation of the optimization problem is by a vector $\mathbf{X} = [x_1, x_2, x_3, \dots, x_n]$ that is based on the formulation of the fitness function $f(\mathbf{X})$. Vector $\mathbf{X}$ is a position vector and a variable model. It has n dimensions, where n is the number of variables that can be found in a given problem. Moreover, $f(\mathbf{X})$ is also called the objective function that can assess how bad or good a vector $\mathbf{X}$ is; the evaluation is performed through multiple survival criteria [14].

Each swarm consists of P particles, P is the number of particles in a swarm, and in each I particle, there is a position vector $\mathbf{X}_i^t = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}]^T$ and a velocity vector $\mathbf{V}_i^t = [v_{i1}, v_{i2}, v_{i3}, \dots, v_{in}]^T$ at a t iteration. The above vectors are updated through the j dimension based on the following equations:

$$\mathbf{V}_{ij}^{t+1} = \omega \mathbf{V}_{ij}^t + c_1 r_1^t (\mathbf{pbest}_{ij} - \mathbf{X}_{ij}^t) + c_2 r_2^t (\mathbf{gbest}_j - \mathbf{X}_{ij}^t), \quad (1)$$

and

$$\mathbf{X}_{ij}^{t+1} = \mathbf{X}_{ij}^t + \mathbf{V}_{ij}^{t+1}, \quad (2)$$

where $i = 1, 2, \dots, P$ and $j = 1, 2, \dots, n$.

There are three terms in Eq. (1) that will be further discussed. Meanwhile, Eq. (2) updates the position of the particle.

The parameter ω is a constant known as the inertia weight; it is a positive constant value used to balance the global search when higher values are specified; it is also known as the exploration weigh [14]. In the velocity update equation, the first term is the product of the inertia mass and the previous velocity of the particle, denoting the incorporation of a particle's previous motion into the current one. Consequently, if $\omega = 1$, the motion of the particle will continue in the same direction, indicating that the particle is completely influenced by its previous motion. On the other



hand, if $0 \leq \omega < 1$, the influence is reduced, so a particle will be directed to other regions in the searching domain. Then, the swarm explores more areas in the search domain as the reduction of inertia weight parameter, which means that the chances of exploring a global optimum is increased. While using lower ω values the simulations will be more time-consuming [14].

The second term of Eq. (1) represents the difference between the particle's best position ($\mathbf{pbest}_{ij}$) and its current position ($\mathbf{X}_{ij}^t$). The idea is that as the particle gets more distant from the ($\mathbf{pbest}_{ij}$) position, the difference ($\mathbf{pbest}_{ij} - \mathbf{X}_{ij}^t$) must increase. The positive constant parameter c_1 is the cognition parameter, and it weights the importance of the particle's own experience. While r_1 is a random value parameter with range, which plays a very important role, as it avoids early convergences, which increases the most likely global optima [14–16].

The third and final term is the social learning term. Because of it, every particle is able to share the optimum point achieved regardless of which particle has found it ($\mathbf{gbest}_j$). The difference ($\mathbf{gbest}_j - \mathbf{X}_{ij}^t$) attracts the particles to the optimum point until found at some t iteration. Similarly, c_2 is the social learning parameter, which weights the importance of swarm's global learning. At last, r_2 plays the same role as r_1 [14].

4. THE PROPOSED SCRAMBLING SCHEME

Various encryption schemes are proposed to protect and secure the audio data; one of them is the proposed system. In this section, a speech scrambling system using Particle Swarm Optimization (PSO) is introduced which is implemented using Matlab.

To solve a practical optimization problem, the number of particles is often set between 10 and 50. At first, the particle positions are initialized randomly, while the velocities are set to 0 or very small random values. The inertia weight parameter w usually decreased from around 0.9 to around

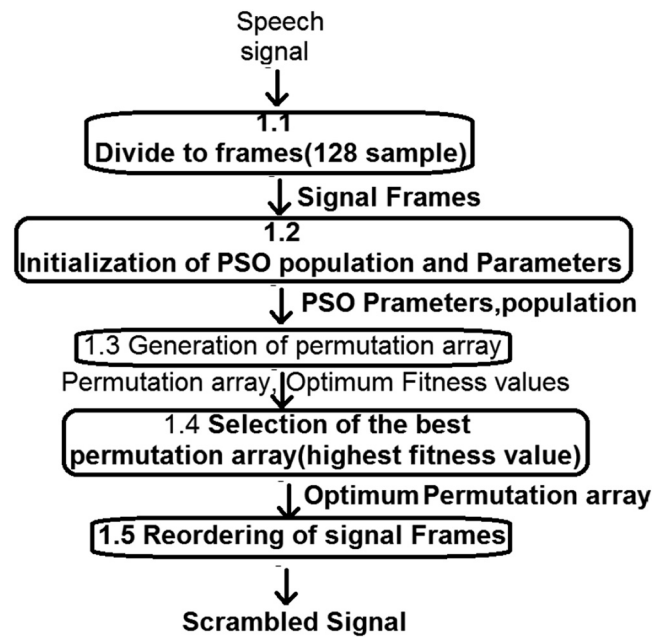


Fig. 1. Block diagram of the proposed scheme

0.4 during computation and the acceleration coefficients c_1 and c_2 are usually between 0 and 4.

The proposed system is illustrated by the block diagram of the processing model in Fig. 1.

Initially, the sound signal is inserted and read in the form of audio samples. The first step is the preprocessing method which divides the samples into frames for each consisting of 128 samples. Then the second step is the initialization of PSO parameters and weights. After that, the initial array is generated in step three, and the optimum array with the highest error is selected in step four. Finally, the audio frames are scrambled and permuted based on the optimum array generated from the PSO algorithm. The output of the system is the final scrambled audio signal. The following steps were followed in the proposed algorithm (Algorithm 1):

ALGORITHM 1

1. Read source speech signal to construct a one dimensional array of samples (Wave-Samples (No. of samples))
2. Calculate number of frames using the following equation

$$n_Frames = \text{No. of wave Samples} / f_size \dots\dots 3$$
 where, f_size is the number of samples in each frame (128).
3. Define original-matrix of size (n-Frames $\times$ 128) to split the signal into frames where each frame has an equal sample of 128 sample.


```

temp=0
for i=1 : n_Frames
    Original-matrix (i,:)= Wave-Samples(temp + 1:temp + f_size);
    temp=temp + f_size;
end
      
```

```

4. Construct a matrix of size (400×f_size) which is assigned to original_frames (400×128), since
   every 400 frames will be processed at a time.
   if (n_Frames>400)
       for i=1 : 400
           original_frames (i,:)= Original-matrix (i,:)
       end
       n_frames=400;
   end

5. Divide the above matrix original_frames () into 4 parts, then apply PSO on each part to generate
   the best permutation indexing array by the following steps:
   a. Initialize the parameters of PSO, population, velocity, position of the particle,  $c_1$ ,  $c_2$ ,  $r_1$  and  $r_2$  ;
   b. Generate randomly an array of indexing of size equal to number of frames divided by four which
   assigned to INDX(i).
   c. Scramble the frames depending on the generated indexing array from the above step for:
       For i=1 to n_frames /4
           Scrambled_frames ( INDX(i) , :) = original_frames( i , :)
       End
   Ps: The symbol (:) in Matlab means the entire row, therefor the rows of the matrix are permuted not
   the columns;
   d. Calculate the fitness value according to the objective function which is minimizing the PSNR
   value between original and scrambled speech;
   e. Follow (1) and (2) to update the position and velocity of the particle;
   f. Recalculate fitness values;
   g. If the maximum number of iterations is reached, the algorithm terminates and the array with best
   fitness value is set to Best_INDX() and used in the scrambling process ; otherwise, go to step 5.
   Evaluation of the results depends on some evaluation metrics Peak Signal-to-Noise Ratio (PSNR) and
   the mean squared error (MSE).
   6. Permute the original frames based on the best permutation indexing array Best_INDX() generated
   by PSO algorithm
       For i=1 to number of frames
           Scrambled_frames ( Best_INDX(i) , :) = original_frames( i , :)
       End
End

```

5. EVALUATION OF THE RESULTS

During the conducted tests in this research, several audio samples have been utilized as test materials in order to study the performance of the developed scheme taking into account the following audio parameters: audio sample resolution, sampling rate, and recording time. Table 1 lists the attributes of these samples, while Fig. 2 presents these samples' waveform patterns. It is important to mention that the audio sample of type stereo has two channels then; only one of these channels will be tested.

Figure 3 shows the effect of the proposed method on the waveform. Figure 4 shows the correlation between original and scrambled signals with and without PSO. Figure 5 shows the correlation between original and recovered signals with and without PSO. Finally, while Table 2 shows the calculated

Table 1. The attributes of the audio test samples

File name	No. of channel	Sample rate	Total samples	Duration	Bit per sample	No. of frame
Handel	Mono	8,192	73,113	8.924	16	571
FIRE	Mono	11,000	11,950	1.086	16	93
HAEMMER1	Stereo	11,025	90,937	8.248	16	710
HALDOING	Mono	11,025	21,944	1.990	16	171
GIVBREAK	Mono	11,025	19,151	1.737	16	149

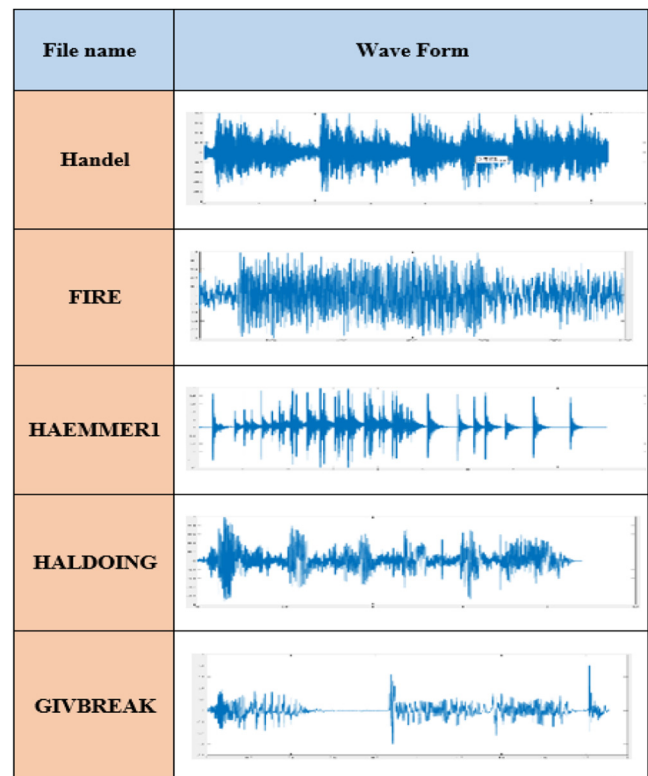


Fig. 2. The waveform patterns of audio samples

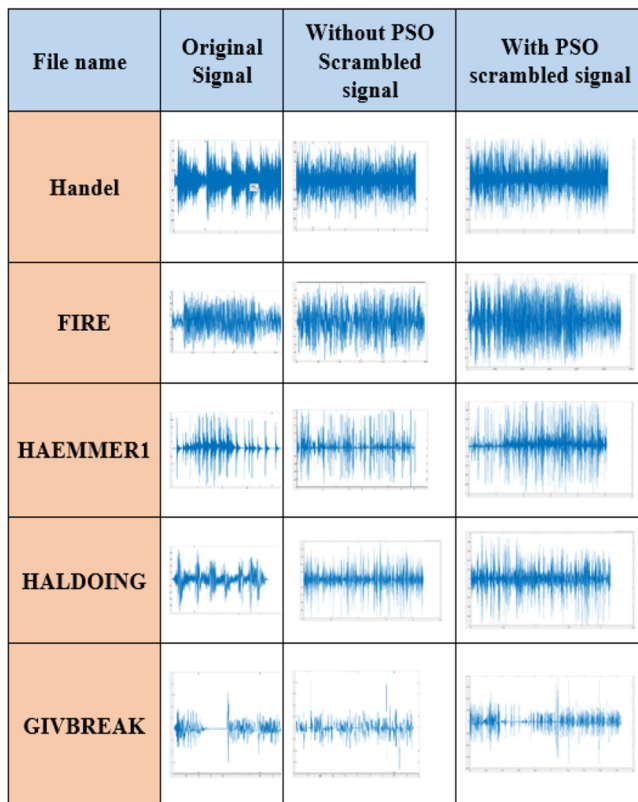


Fig. 3. Effect of PSO on the waveform of samples after scrambling

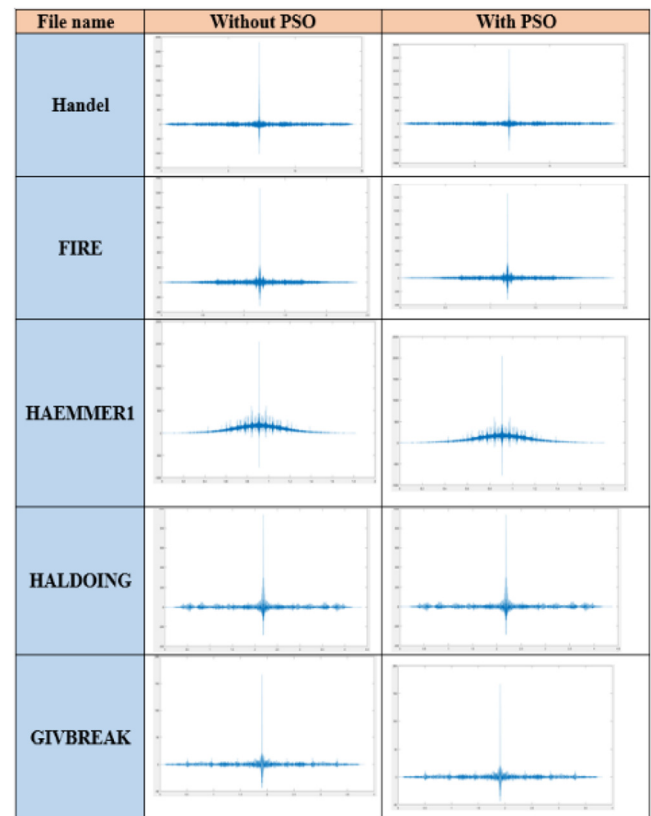


Fig. 5. Correlation between original and recovered signal

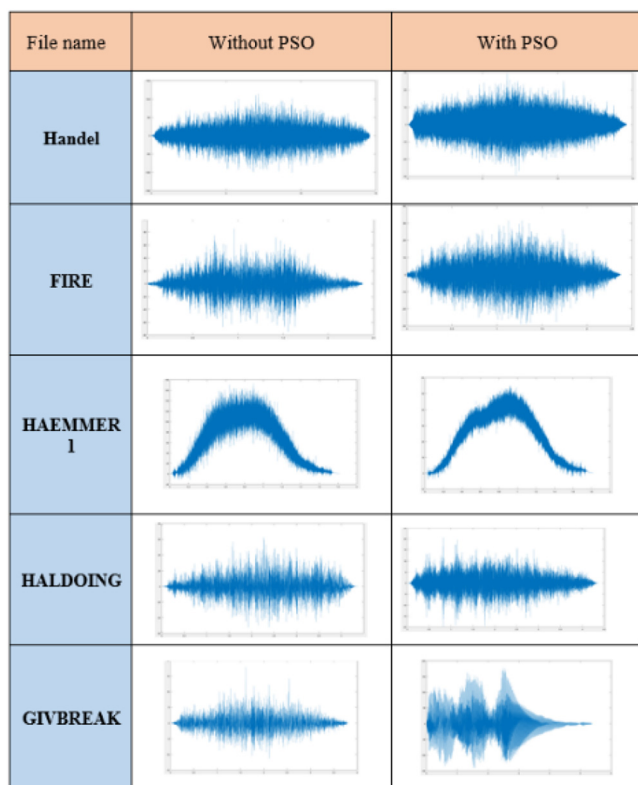


Fig. 4. Correlation between original and scrambled signal

segmental peak signal to noise ratio distance measure between original and scrambled speech for both models, it also shows the estimated time taken by each sample.

It is noticeable in Fig. 3 that the signal is scattered more with PSO than without it. As for the correlation in general, if the signal clearly exhibits a peak, this means that the original signal matches with the scrambled signal, so in Fig. 4, the correlated signal by PSO has no peak and is wider than the traditional signal, meaning the signal generated by PSO is more different from the original one than the traditional scheme. Finally, Fig. 5 shows that the correlated signal between original and recovered signals in both schemes indicates a large peak which means that the original signal matches the recovered signal. Table 2 shows that the proposed scheme gives a higher value of Mean Squared Error with no change in the Estimated Time.

6. CONCLUSIONS

Using PSO in the permutation process gives a dynamic permutation array each time using the system for a higher level of security and better PSNR values compared to the conventional scheme. Moreover, the ET remains the same in both schemes. The testing results of the proposed system's performance using three measuring evaluations (MSE, PSNR, and ET) show that our proposed system has potential, as shown by the results of the evaluations.

Table 2. Effect of PSO on PSNR, MSE, and ET

File name	Without PSO			With PSO		
	MSE	PSNR	Time	MSE	PSNR	Time
Handel	0.075	81.29	1.56	0.07	80.75	2.01
FIRE	0.214	70.85	0.93	0.23	69.84	0.413
HAEMMER1	0.042	87.07	2.58	0.03	86.51	2.71
GIVBREAK	0.017	95.69	1.06	0.02	94.39	0.89
HALDOING	0.085	80.06	1.00	0.08	80.04	0.814

It has been observed that the signal that was scrambled by using PSO has no correlation with the signal that was being scrambled originally, and it is scrambled very much. The waveform of the scrambled signal is erratic and highly distorted, which makes the scrambled speech less intelligible to those who can still make it out.

REFERENCES

- [1] S. B. Sadkhan and N. A. Abbas, "Performance evaluation of speech scrambling methods based on statistical approach," *Atti della Fondazione Giorgio Ronchi*, vol. 66, no. 5, pp. 601–614, 2011.
- [2] H. Kohad, V. R. Ingle, and M. A. Gaikwad, "An overview of speech encryption techniques," *Int. J. Eng. Res. Dev.*, vol. 3, no. 4, pp. 29–32, 2012.
- [3] S. Rajanarayanan and A. Pushparaghavan, "Recent developments in signal encryption – A critical survey," *Int. J. Scientific Res. Pub.*, vol. 2, no. 6, pp. 1–7, 2012.
- [4] R. Huang, K. H. Rhee, and S. Uchida, "A parallel image encryption method based on compressive sensing," *Multimedia Tools Appl.*, vol. 72, pp. 71–93, 2012.
- [5] G. Dhanya and J. Jayakumari, "Speech scrambling based on chaotic mapping and random permutation for modern mobile communication systems," *Aptikom J. Computer Sci. Inf. Tech.*, vol. 2, no. 1, pp. 20–25, 2017.
- [6] A. M. Raheema, S. B. Sadkhan, and S. M. A. Satar, "Performance enhancement of speech scrambling techniques based on many chaotic signals," in *2020 International Conference on Computer Science and Software Engineering*, Duhok, Iraq, April 16–18, 2020, pp. 308–313.
- [7] H. K. Zghair, S. A. Mehdi, and S. B. Sadkhan, "Speech scrambler based on discrete cosine transform and novel seven-dimension hyper chaotic system," *J. Phys. Conf. Ser.*, vol. 1804, 2021, Paper no. 012048.
- [8] J. F. de Andrade, M. L. R. de Campos, and J. A. Apolinario, "Speech privacy for modern mobile communication systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, NV, USA, March 31–April 4, 2008, pp. 1777–1780.
- [9] A. Srinivasan and P. A. Selvan, "A review of analog audio scrambling methods for residual intelligibility," *Innovative Syst. Des. Eng.*, vol. 3, no. 7, pp. 22–38, 2012.
- [10] D. M. L. Ballesteros and J. M. A. Moreno, "Speech scrambling based on imitation of a target speech signal with non-confidential content," *Circuits Syst. Signal Process.*, vol. 33, pp. 3475–3498, 2014.
- [11] S. B. Sadkhan and N. A. Abbas, "Performance evaluation of speech scrambling methods based on statistical approach," *Atti della Fondazione Giorgio Ronchi*, vol. 66, no. 5, pp. 601–614, 2011.
- [12] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Probl. Eng.*, vol. 2015, 2015, Paper no. 931256.
- [13] Y. Zhang, S. Wang, Z. Dong, P. Phillip, G. Ji, and J. Yang, "Pathological brain detection in magnetic resonance imaging scanning by wavelet entropy and hybridization of biogeography-based optimization and particle swarm optimization," *Prog. Electromagnetics Res.*, vol. 152, pp. 41–58, 2015.
- [14] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaziz, "Particle swarm optimization: technique, system and challenges," *Int. J. Computer Appl.*, vol. 14, no. 1, pp. 19–27, 2011.
- [15] G. H. Nasir and K. Jármai, "Kinematic-based structural optimization of robots," *Pollack Period.*, vol. 14, no. 3, pp. 213–222, 2019.
- [16] G. H. Nasir, S. Alsamia, and K. Jármai, "Fertilization optimization algorithm on CEC2015 and large scale problems," *Pollack Period.*, vol. 17, no. 1, pp. 24–29, 2022.