Speech Scrambling based on Particle Swarm Optimisation

Zeina Hassan Razaq
College of Education, University of Kufa, Najaf, Iraq. E-mail: zienah.alhadad@uokufa.edu.iq

Received July 10, 2021; Accepted November 12, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19005

Abstract

Securing any communication system where important data may be transmitted through the channel is a very crucial issue. One of the good solutions in providing security for the speech is to use speech scrambling techniques. The chaotic system used in security has properties that make it a good choice for scrambling speech signal and the optimisation algorithm can provide a perfect performance when used to enhance the hybrid of more than one method. In this paper, we suggest a system that uses an optimisation method, namely, particle swarm optimisation. The evaluation measures prove that the output of the optimisation method has better performance among the methods used in the comparison, including chaotic maps and hybrid chaotic maps.

Keywords

Arnold Cat Map, Arnold–Lucas, Descrambling, Fibonacci Map, PSO, Scrambling.

Introduction

Nowadays, the security of any communication system has become very crucial. Therefore, we must protect transmitted information against any eavesdropper attempts. Our focus in the present paper is the security of the speech. Many techniques are used to secure speech, such as speech scrambling, which permutes speech samples to eliminate residual intelligibility of transmitted speech. The chaotic system has good randomness, making it a good choice in generating the key to scrambling. Moreover, orthogonal transformation is used to reduce the intelligibility of the speech signal. Optimisation techniques are used to enhance the hybridisation of the methods.

An optimisation method is an important tool that searches for the best solution with more fitness criteria between many workable solutions.

Particle swarm optimisation (PSO) is an optimisation technique that is a population-based stochastic technique inspired by fish schools and bird flocks.

Speech scrambling techniques have been developed in many fields. Many methods designed to satisfy this state depend on one of the accepted methods for mixing text in an unpredicted manner and retrieving the original file on the other end at the receiver side to protect audio files from eavesdropping or prevent any third party from understanding secure information transmitted through an insecure channel (Sath and Ramik 2017).

The algorithm of speech scrambling typically acts as a permutation for the segment of the speech in the time domain, frequency and time-frequency domain or the permutation of the transform order coefficient of the blocks of speech (Cristina et al., 2016). Scrambling methods have also been used to convert the pure speech signal into an unintelligible form to avoid eavesdroppers (Sattar, 2012).

Chaotic Generators

Chaos is a pseudo-random process provided in a system with dynamical and nonlinear properties. Chaos in nature is non-periodic, non-convergent and highly susceptible to the initial states.

The chaotic theory, which was proposed by Boltzmann, is considered to be a new nominee for cryptography. This theory returns to the characteristics and advantages in data encryption (Saad et al., 2014). Chaotic systems have many properties, including sensitivity, which depends on the initial condition, continuous broadband spectrum and ergodicity. The word 'ergodic' has Greek origins and is composed of two words (Hussein, 2017), namely, 'ergon' meaning work and 'odos' meaning path. The main properties (diffusion and confusion) must be present in a good encryption system. Two types of non-regular natural behaviour in a real system are chaos and noise (Saad et al., 2014). Accurate knowledge of parameters and initial conditions of the chaotic system leads to the return of the raw data after encryption. Using chaos theory with cryptography returns to the world Shannon (Hussein, 2017).

1) Types of Chaotic Maps

Many chaotic maps are used in cryptography. In this section, we present several types of chaos maps used in the proposed system.

- **Arnold Cat Map**

Arnold Cat Map (ACM) is a simple product of several principles of chaos (Minati, 2012). The change in this map is used to raise security and is the 2D map presented in Equation (1):

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} k+1 & k \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{N}, (1)$$

Where x_n and y_n are the indices in a square matrix (N × N), n = 1, 2, 3,…, N-1, x_{n+1}, y_{n+1} is the index mapping of the transformation, k = {1, 2, 3,....} and N is length of the square array. This map is used in the proposed system.

- **Fibonacci Transform**

The Fibonacci sequence n is a series of integer numbers presented in the following recurrence:

$$F_n = \begin{cases} 0, \text{if } n = 1 \\ 1, \text{if } n = 2 \\ F_{n-1} + F_{n-2} \text{ otherwise} \end{cases} \quad (2)$$

The formed series represents the values: 0, 1,1,2,3,5,8,…. The 2×2 matrix is represented by four consecutive values of the Fibonacci series. This matrix is unimodular and will be considered as a matrix scrambler. The generalised form of the Fibonacci Transform is presented in (Minati et al., 2012).

Definition: The generalised mapping of Fibonacci is the mapping.

F: T2 → T2 such that

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} F_i & F_{i+1} \\ F_{i+2} & F_{i+2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (3)$$

- **Lucas Series**

The Lucas series is considered a special state of the Fibonacci series and is defined as follows:

$$L_n = \begin{cases} 2, \text{if } n = 0 \\ 1, \text{if } n = 1 \\ L_{n-1} + L_{n-2} \text{ otherwise} \end{cases}, \quad (4)$$

Where n is the index and L is the Lucas value. The series forms the numbers: 2, 1, 3, 4, 7, 11, 18, 29,… (Minati et al., 2012).

- **Arnold–Lucas Transform**

The map comprises the first line from the Arnold map and the second line from the Lucas maps (Nidaa et al., 2018) as represented in the following:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \\ L_1 & L_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod N \qquad (5)$$

PSO

Global optimisation is part of applied mathematics and numerical analysis, which depends on good optimisation. The idea from the general optimisation is to discover from the possible element x the best element of the X set depending on a group of standards $F = \{f_1, f_2 \ldots f_n\}$. The standards are presented as mathematical functions, which are called objective functions (Thomas et al., 2009).

PSO is a heuristic optimisation method, which is inspired by the intelligence of swarm and is dependent on research on the behaviour of the movement of birds and fish (Toshinori et al., 2008).

Before the birds determine where to find food, their behaviour will be scattering, converging or going together. As birds search for food between places, one bird can smell food well. Thus, birds are aware of where to find food with better information about food resources.

They, especially, the good one, transmit information at any time during food searching between places; they eventually fly to the food location (Muhammad, 2014).

In terms of the PSO algorithm, the swarm checks with the bird flock. The bird transmission from one point to another is equal to developing a solution in the swarm. Good information represents the best solution. Food resources are the most optimistic solution throughout the cycle. The most optimistic solution could be found in the PSO algorithm through the individual's cooperation for each one (Rania et al., 2005).

The position of the most optimistic value influences the swarm position of each particle through the particle's movement (the experiment of the individual) and the location of the most optimistic particle in the particle's vicinity (that is near the experiment) (Jun et al., 2012). PSO is setting with the collection of particles (solutions) selected randomly and looks to optimise through generation updating.

The updating of each position of particle performed by the three ‘best’ values explained in the following (Jun et al., 2012).

A) The optimal solution you have carried so far (also save the fitness value), which is called the p-best (Qinghai et al., 2010).

B) The second-best global value is g-best which is found by the optimiser of the particle swarm (Qinghai et al., 2010).
C) The third best value is the best local (l-best) if the particle participates in the population within topological neighbours. The neighbour that is the best one allows parallel reconnaissance of the search area and reduces PSO's fall exposure to local minimums but slows the speed of convergence (Qinghai et al., 2010).

The particles try to adjust current locations and speeds depending on the distance between the current location of each swarm and p-best and distance between the current location and g-best of the swarm (Jun, 2012).

$$v_{n+1} = v_n + c_1\, \text{rand1}() * \left(p_{best,n} - x_n \right) + c_2\, \text{rand2}() * \left(g_{best,n} - x_n \right) \quad (6)$$

$$x_{[n+1]} = x_{(n)} + v_{n+1} \quad (7)$$

Where
$x_{(n+1)}$: is particle position at (n+1)th iteration
$x_{(n)}$: is particle position at (n)th iteration
v_{n+1}: is particle velocity at (n+1) th iteration
v_n: is particle velocity at nth iteration
c_1, c_2: are factors of acceleration that are related to g-best and l-best, respectively.
rand1( ), rand2( ): are random numbers in range 0 to 1
g-best: the swarm's g-best position (the best solution reached in the whole swarm)
p-best: the p-best position of the particle (the best solution reached for the current particle only).

Evaluation Measurements

The most crucial measures generally used for evaluating the techniques of speech scrambling are as follows:

1. The strength of the scrambler is to output the scrambled signal with low clarity.
2. The range to which encryption or decryption affects the recovered speech quality in the receiver.
3. The immunity of the scrambler or encryption system to cryptanalytic attack (Hemlata et al., 2012), (Ghassan et al., 2014).

Testing uses a set of measures for calculating residual intelligibility and quality of speech. We use many types of measures including correlation coefficients (CC) and signal to noise ratio (SNR). They are clarified in detail as follows:

1) CC

CC considers statistical measures that test the encryption quality of an encryption system. This analysis evaluates the relationship between two examples of value in the range +1 to 1. A correlation value that reaches zero means the weakest relationship between the two samples and disability of attackers to predict the secret key. CC is computed as follows:

$$E(x) = \frac{1}{T}\sum_{i=1}^{T} x_i \quad (8)$$

$$D(x) = \frac{1}{T_s}\sum_{i=1}^{Ts}\left(x_i - E(x_i)\right)^2 \quad (9)$$

$$\text{cov}(x,y) = \frac{1}{T}\sum_{i=1}^{Ts}\left(x_i - E(x_i)\right)\left(y_i - E(y_i)\right) \quad (10)$$

$$r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)D(y)}} \quad (11)$$

2) SNR

SNR tests remaining intelligibility and quality of the encrypted signal. In general, the encoded signal with a low SNR signal indicates a higher noise level than the original speech signal, whereas a high SNR means high quality of the decrypted signal (Farsana,2016), (Emad, 2011); SNR is calculated (Nikolaos, 2008) as follows:

$$SNR = 10\log_{10}\frac{\sum_{i=1}^{T} I^2(i)}{\sum_{i=1}^{Ts}(I(i)-D(i))^2}, \quad (12)$$

Where D (i) is the decrypted speech value, and I (i) is the value of original speech.

3) Log Spectral Distance

This measure is closely related to self-assessment of vocal differences because the loudness of the perceived sound of a sign is logarithmic in nature (Hussein, 2017). The logarithmic difference between the S (w) original speech signal power spectrum and the power spectrum S'(w) can be determined for the corresponding frame of the encrypted/restored signal by the following:

$$V(w) = logS(w) - logS'(w),\ w = 0, 1, \ldots N-1 \ldots. \quad (13)$$

The d_log between the densities S and S' is presented in

$$d\log = (S,S')^p = \frac{1}{N}\sum_{i=0}^{N-1}|V(W)|^p, w = 0,1,2,\ldots N-1 \quad (14)$$

The p determines the distance measure order.

Proposed System

The proposed system consists of two sections, namely, scrambling and descrambling PSO with chaotic maps.

The standard algorithm of PSO contains several parameters that must be determined before using the algorithm. Examples include accelerated constants, inertia weight, maximum speed and a number of swarm particles. Gaussian distribution can improve the capability of the convergence for PSO without having to adjust these parameters. The only parameter defined by the user is the number of the particles.

The steps of PSO are presented in Algorithm 1.

Algorithm 1: Algorithm of PSO

Step 1: Set parameter w, c_1 and c_2 of PSO.
Step 2: Initialise population of particles having positions X and velocities V.
Step 3: Set iteration k=1.
Step 4: Calculate fitness of particles $F^k_i = f(X^k_i)$, and find the index of the best particles b.
Step 5: Select $Pbest^k_i = X^k_i$ and $Gbest^k = X_b{}^k$.
Step 6: Update velocity and position of particles in Equations (6) and (7).
Step 7: Evaluate fitness $F_i{}^{k+1}=f(X_i{}^{k+1})$ and find the index of the best b1.
Step 8: Update Pbest of population.
If $F_i{}^{k+1} < F_i{}^k$, then $Pbest_i{}^{k+1} = X_i{}^{k+1}$; else $Pbest_i{}^{k+1}= Pbest_i{}^k$.
Step 9: Update Gbest of population.
If $F_{b1}{}^{k+1} < F_b{}^k$, then $Gbest^{k+1} = Pbest_{b1}{}^{k+1}$ and set b=b1; else, $Gbest^{k+1}=Gbest^k$.
Step 10: If k < maxr, then k=k+1; go to Step 6; else, go to Step 11.
Step 11: Print optimum solution as $Gbest^k$.

1) Scrambling Using PSO with Chaotic Maps

The speech vector is input to two chaotic maps, performs the PSO on these two maps as initial input and selects the best particle according to its fitness after N rounds. The particle represents the hybrid of the two input maps. That particle represents the key used in scrambling.

The two maps entered to the PSO may be: (ACM and Fibonacci map), (ACM and Lucas map) and (Lucas map and Fibonacci map).

The algorithm of PSO with chaotic maps is illustrated in Algorithm 2.

Algorithm 2: Algorithm of Scrambling using PSO with Chaotic Maps

Step 1: Convert the speech vector to the 2D matrix.
Step 2: Perform (ACM and Fibonacci map) using Equation (1) and (3)/(ACM and Lucas map) using Equation (1) and (5)/(Fibonacci map and Lucas map) using Equation (3) and (5).
Step 3: Convert the two matrices back to the 1D array.
Step 4: Perform PSO using Algorithm 1.
Step 5: Select the best solution as the scrambling key.

The speech is converted to a 2D matrix, and two chaotic maps are performed from the states (ACM and Fibonacci map), (ACM and Lucas map) or (Fibonacci map and Lucas map).

The output enters the PSO, and the better particle will be used as the scrambling key.

2) Descrambling using PSO with Chaotic Maps

The descrambling algorithm on the second side (the receiver side) is used to recover the original clear speech signal.

Descrambling using this method is explained in the Algorithm (3).

Algorithm 3: Algorithm of Descrambling using PSO with Chaotic Maps

Step1: Perform the key used in scrambling.
Step2: Check for additional zeros at the end of the descrambled vector and eliminate them.
Step3: Save the final clear speech vector.

The key to scrambling will be secretly sent to the receiver side. This key will be applied to the scrambled signal, and the resulting descrambled vector will be longer than the original length of the clear speech signal. Thus, the additional zeros at the end of the descrambled vector are removed from the descrambled signal.

Results and Discussion

In this part, we attempt to find a better performance using the optimisation algorithm, which we decide according to its outputs when we use the best hybrid. We use PSO as the optimisation algorithm.

The speech data were scrambled using two chaotic maps (ACM and Fibonacci map, ACM and Lucas map, or Fibonacci map and Lucas map). These scrambled vectors were considered as initial input to the PSO algorithm, where position X, which represents the scrambled vectors, is updated in each iteration and looped until the stop condition is satisfied. Finally, the best overall particles will be used as scrambled speech data. Table (1) presents the SNR of the three states used in the study with PSO and compares the results with the traditional hybrid performed on the same states.

In Table 1, The hybrid (ACM, Fibonacci map) performed by PSO provided the best SNR that reached 1.02303161, the best value of SNR for (ACM, Lucas map) was 1.02769, and the least value for the state (Fibonacci map, Lucas map) was 1.02637075.

Table 1 SNR for PSO with Chaotic Maps_ Scrambling Process

	Traditional Hybrid			**PSO**		
File	ACM, FIB	ACM, Lucas	FIB, Lucas	ACM, FIB	ACM, Lucas	**FIB, Lucas**
1	1.36687	1.30228	1.29318	1.2176841	1.2461	**1.20166**
2	1.1056	1.1308	1.05653	1.102679897	1.12891692	**1.02637075**
3	1.2631	1.2369	1.23037	1.2321827	1.217917	**1.2016137**
4	1.13015	1.14283	1.15204	1.040784	1.0299882	**1.0987025**
5	1.28547	1.27743	1.28132	1.03521296	1.03371495	**1.137925**
6	1.26272	1.29434	1.25702	1.1972921	1.160729	**1.21985**
7	1.13101	1.32885	1.3083	1.3080396	1.302477	**1.253063**
8	1.2701	1.26309	1.20699	1.1689848	1.16984	**1.19099**
9	1.0788	1.08548	1.08321	1.02303161	1.02769	**1.060931**
10	1.4364	1.4395	1.45554	1.2684678	1.306225	**1.298911**

The log spectral distance for the three states using traditional hybrid and PSO were also evaluated. Table 2 shows the log spectral distance of the three states and compares the results using the PSO with the traditional hybrid performed on the same states.

In Table 2, the maximum value of log spectral distance was provided by using PSO, which performed the same states explained above. The hybrid (ACM and Fibonacci map) performed by PSO yielded the best log spectral distance, which reached 2.78324381, the best value of log spectral distance for (ACM and Lucas map) was 2.75875938, and the best value for the state (Fibonacci map and Lucas map) was 2.78308339.

Table 2 Log Spectral Distance for PSO with Chaotic maps_ Scrambling Process

File	Traditional Hybrid			PSO		
	ACM, FIB	ACM, Lucas	FIB, Lucas	ACM, FIB	ACM, Lucas	**FIB, Lucas**
1	1.68962	1.579991	1.58208	2.78324381	2.7498236	**2.78308339**
2	1.82278	1.710579	1.69824	2.72805026	2.71754474	**2.72361713**
3	1.717869	1.60083	1.60332	2.76011227	2.73643901	**2.74446003**
4	1.719774	1.5606785	1.50457	2.76254893	2.75776718	**1.79547066**
5	1.67289	1.583095	1.576355	2.7638735	2.74775045	**2.76127768**
6	1.48281	1.33992	1.35622	2.77634634	2.75875938	**2.774934**
7	1.67540	1.59526	1.58957	2.68954268	2.67173473	**2.66824702**
8	1.661933	1.568762	1.57902	2.72345515	2.71570371	**2.72186061**
9	1.71327	1.611463	1.62057	2.70137496	2.69536683	**2.71121156**
10	1.60296	1.500998	1.50526	2.71986811	2.70432429	**2.71099762**

The correlation between the three states using traditional hybrid and PSO were also evaluated. Table 3 shows the correlation between the three states and compares the results using PSO with the traditional hybrid performed on the same states.

In Table 3, the best value of correlation was provided by using PSO that performed the same states explained above. The hybrid (ACM and Fibonacci map) performed by PSO provided the best correlation that reached 0.00031662, the best value of correlation for (ACM and Lucas map) was 0.00131673 and the best value for the state (Fibonacci map and Lucas map) was −0.00064.

Table 3 Correlation for PSO with Chaotic maps_ Scrambling Process

File	Traditional Hybrid			PSO		
	ACM, FIB	ACM, Lucas	FIB, Lucas	ACM, FIB	ACM, Lucas	FIB, Lucas
1	0.02491	0.01034	0.0082	−0.0020124	0.00908	0.006996
2	0.009298	0.0898	0.00272	0.00278263	0.00298	0.0209824
3	0.0152	0.00882	0.00819	0.00328996	0.00199	0.007826
4	−0.00452	−0.00693	0.01111	−0.001108088	0.00288	−0.001099
5	0.005132	−0.0054	0.00418	0.001011788	0.004988	−0.00064
6	0.0098	0.0822	0.00233	0.00176529	0.080719	−0.00191802
7	0.0583	0.0027	0.00236	0.00031662	0.00131673	−0.00180291
8	0.0919	0.0756	−0.00394	−0.00215956	−0.0019860	0.0029456
9	−0.00116	0.0064	0.00161	−0.001188302	−0.005096	0.0036290
10	0.01648	−0.03689	0.1933	−0.01185082	0.001868	0.001628

The measures used in the descrambling process are explained in Tables 4, 5 and 6. The three measures computed for descrambling measured the quality of the proposed algorithm of the proposed system for retrieving original speech.

In Table 4, the hybrid (ACM and Fibonacci map) performed by PSO provided the best SNR, which reached 86.645982, the best value of SNR for (ACM and Lucas map) was 86.65012 and the best value for the state (Fibonacci map and Lucas map) was 86.645621.

Table 4 SNR for PSO with Chaotic Maps _Descrambling Process

File	Traditional Hybrid			PSO		
	ACM, FIB	ACM, Lucas	FIB, Lucas	ACM, FIB	ACM, Lucas	**FIB, Lucas**
1	86.521391	86.5210914316861	86.5416913449658	86.5397966	86.556342	**86.556721**
2	86.525141430	86.5038430628082	86.5244429760878	86.645982	86.558988	**86.547812**
3	86.465685	86.4553855177839	86.4759854310635	86.467210	86.490215	**86.489019**
4	86.4406802	86.4303803081874	86.450980221467	86.450911	86.458921	**86.479019**
5	86.4650199	86.454719984193	86.4753198974726	86.4651	86.47312	**86.490324**
6	86.438686	86.4165617187889	86.4371616320685	86.4390	86.490381	**86.458321**
7	86.491839	86.48153907798	86.5021389912658	86.49219	86.48019	**86.512844**
8	86.469577481	86.4474481894146	86.4568048102694	86.46925	86.45099	**86.456602**
9	86.3411397	86.3308398085933	86.3514397218729	86.345127	86.35980	**86.409353**
10	86.6429087	86.6187873751594	86.63387288439	86.64563	86.65012	**86.645621**

The log spectral distance for the three states using traditional hybrid and PSO were also evaluated for descrambling. Table 5 shows the log spectral distance of the three states and compares the results using PSO with the traditional hybrid performed on the same states for descrambling. The table shows that the best value of log spectral distance was the least value.

In Table 5, the best value of the log spectral distance was provided by using the PSO that performed the same states explained above. The hybrid (ACM and Fibonacci map) performed by PSO provided the best log spectral distance that reached 0, the best value of log spectral distance for (ACM, Lucas map) was 0, and the best value for the state (Fibonacci map, Lucas map) was 0.

Table 5 Log Spectral Distance for PSO with Chaotic maps_ Descrambling Process

File	Traditional Hybrid			PSO		
	ACM, FIB	ACM, Local	FIB, Lucas	ACM, FIB	ACM, Lucas	**FIB, Lucas**
1	0	0.00000032	0.00000064	0	0.00000019	**0.00000009**
2	0	0.00000055	0.00000109	0	0.00000010	**0.00000106**
3	0.000000026	0.00000653	0.00001244	0	0	**0.00000104**
4	0	0.00000036	0.0000072	0	0	**0**
5	0	0.00000031	0.00000062	0	0.00000009	**0**
6	0	0.000000044	0.00000089	0.00000058	0	**0**
7	0	0.00000605	0.00001182	0	0	**0**
8	0.00000001	0.000000148	0.00000095	0	0.000000089	**0**
9	0	0.000000189	0.00000177	0	0.0000000010	**0**
10	0.000000061	0.000000132	0.00000063	0	0.000000002	**0**

The correlation between the three states using traditional hybrid and PSO were also evaluated for descrambling. Table 6 shows the correlation between the three states and compares the results using the PSO with the traditional hybrid performed on the same states where the best value of correlation in the descrambling process was the value nearest to 1.

In Table 6, the best value of correlation was provided using PSO that performed the same states explained above. The hybrid (ACM and Fibonacci map), (ACM and Lucas map) and (Fibonacci map and Lucas map) was 1.

Table 6 Correlation for PSO with Chaotic maps_ Descrambling Process

File	Traditional Hybrid			PSO		
	ACM, FIB	ACM, Lucas	FIB, Lucas	ACM, FIB	ACM, Lucas	**FIB, Lucas**
1	1	1	1	1	1	**1**
2	1	1	1	1	1	**1**
3	1	1	1	1	1	**1**
4	1	1	1	1	1	**1**
5	1	1	1	1	1	**1**
6	1	1	1	1	1	**1**
7	1	1	1	1	1	**1**
8	1	1	1	1	1	**1**
9	1	1	1	1	1	**1**
10	1	1	1	1	1	**1**

For the three measures used for comparison, the hybrid using the traditional methods and the hybrid using PSO was used for comparison between the three states in traditional methods and the three states using the PSO.

In the hybrid using PSO, the best state was (ACM and Fibonacci map) according to the measures that reached 15 states in scrambling and 22 states in descrambling.

Conclusions

In our proposed system, we focus on the use of the PSO in enhancing the performance of the methods used in speech scrambling. We find that in PSO with chaotic maps, the best hybrid is provided by ACM and Fibonacci map.

References

Sathiyamurthi, P., & Ramakrishnan, S. (2017). Speech encryption using chaotic shift keying for secured speech communication. *EURASIP Journal on Audio, Speech, and Music Processing*, *2017*(1), 1-11.

Duta, C.L., Gheorghe, L., & Tapus, N. (2016). Performance Analysis of Real Time Implementations of Voice Encryption Algorithms using Blackfin Processors. *In International Conference on Information Systems Security and Privacy*, *2*, 157-166.

Sadkhan, S., & Abbas, N. (2012). Speech scrambling based on wavelet transform. *Advances in wavelet theory and their applications in engineering physics and technology*, 41-58.

Al Saad, S.N., & Hato, E. (2014). A speech encryption based on chaotic maps. *International Journal of Computer Applications*, *93*(4), 19-28.

Ismael, H.A. (2017). *A proposed Speech Scrambling Based on Multi Chaotic Maps as Key Generators* (Doctoral dissertation, Master dissertation, University of Babylon College of Information Technology Department of Software, Iraq, 2107).

Mishra, M., Mishra, P., Adhikary, M.C., & Kumar, S. (2012). Image encryption using Fibonacci-Lucas transformation. *International Journal on Cryptography and Information Security (IJCIS)*, *2*(3).

Abbas, N.A., & Razaq, Z.H. (2018). Speech Scrambling Based on Arnold-Lucas Mapping. *In International Conference on Advanced Science and Engineering (ICOASE)*, 290-295.

Weise, T. (2009). Global optimization algorithms-theory and application. *Self-Published Thomas Weise*.

Munakata, T. (2008). *Fundamentals of the new artificial intelligence: neural, evolutionary, fuzzy and more*. Springer Science & Business Media.

Sohail, M.S., Saeed, M.O.B., Rizvi, S.Z., Shoaib, M., & Sheikh, A.U.H. (2014). Low-complexity particle swarm optimization for time-critical applications. *arXiv preprint arXiv:1401.0546*.

Hassan, R., Cohanim, B., De Weck, O., & Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. *In 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*.

Sun, J., Wu, X., Palade, V., Fang, W., Lai, C. H., & Xu, W. (2012). Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences*, *193*, 81-103.

Bai, Q. (2010). Analysis of particle swarm optimization algorithm. *Computer and information science*, *3*(1), 180-184.

Kohad, H., Ingle, V.R., & Gaikwad, M.A. (2012). An overview of speech encryption techniques. *International journal of Engineering research and development*, *3*(4), 29-32.

Ghassan, M.H. (2014). Speech Scrambling Using Multi-Stage Permutation with Filter Output. *Iraqi Journal of Information Technology*, *6*(2), 90-101.

Farsana, F.J., & Gopakumar, K. (2016). A novel approach for speech encryption: Zaslavsky map as pseudo random number generator. *Procedia computer science*, *93*, 816-823.

Mosa, E., Messiha, N.W., Zahran, O., & Abd El-Samie, F.E. (2011). Chaotic encryption of speech signals. *International Journal of Speech Technology*, *14*(4), 285-296.

Doukas, N., & Karadimas, N.V. (2008). A blind source separation based cryptography scheme for mobile military communication applications. *WSEAS Transactions on Communications*, *7*(12), 1235-1245.

Rachman, Y.B., Mutiarani, H., & Putri, D.A. (2018). Content analysis of Indonesian academic libraries' use of Instagram. *Webology*, *15*(2), 27-37.