ISSN: 0067-2904

Proposed Hybrid Technique in Cryptanalysis of Cryptosystem Based on PSO and SA

Riyam Noori Jawad*
Total Quality Management Techniques, Technical College for Management, Middle Technical University, Baghdad, Iraq

Received: 25/10/2021 Accepted: 5/3/2022 Published: 30/10/2022

Abstract

Modern ciphers are one of the more difficult to break cipher systems because these ciphers high security, high speed, non - propagation error and difficulty in breaking it. One of the most important weaknesses of stream cipher is a matching or correlation between the output key-stream and the output of shift registers.

This work considers new investigation methods for cryptanalysis stream cipher using ciphertext only attack depending on Particle Swarm Optimization (PSO) for the automatic extraction for the key. It also introduces a cryptanalysis system based on PSO with suggestion for enhancement of the performance of PSO, by using Simulated Annealing (SA). Additionally, it presents a comparison for the cryptanalysis system results that were obtained by the proposed technique, which is called Modified PSO (MPSO) with classical PSO and GA. These algorithms can be used for reducing the number of attempts or trials of key space, which are needed to reach to the optimal solution (the exact initial setting of Linear Feedback Shift Register (LFSR)),and increase the speed of the search process to find the best solution. Based on the obtained results, these algorithms were shown to be effective at finding the optimal solution and the MPSO method operates better compared with PSO in the term of time and accuracy. Well known stream cipher systems were attacked by the two soft computing methods as the cases of study, which are Geffe , Brüer and Shrinking stream cipher systems.

Keywords: Particle Swarm Optimization, Modified Particle Swarm Optimization, Stream Cipher, Bruer Generator, Shrinking Generator, Cryptanalysis.

اقتراح تقنية هجينة في تحليل أنظمة التشفير بالاعتماد على خوارزمية سرب الطيور وخوارزمية SA

ريام نوري جواد

قسم تقنيات إدارة الجودة الشاملة ,الكلية التقنية الإدارية , الجامعة التقنية الوسطى , بغداد, العراق.

الخلاصة:

تعد الشفرات الحديثة (الشفرات الانسيابية) أحد أنظمة التشفير الإلكترونية الصعبة بسبب الأمان العالي والسرعة العالية وعدم انتشار الخطأ وصعوبة كسره. واحدة من أهم نقاط الضعف في التشفير الانسيابي هي المطابقة أو الارتباط بين المخرجات من المفتاح والمخرجات من المسجلات الزاحفة .

*Email: riyamnoori@mtu.edu.iq

يأخذ هذا العمل في الاعتبار اكتشاف طريقة جديدة لتحليل أنظمة التشفير الانسيابية غير الخطية باستخدام نظرية (هجوم النص المشفر فقط) اعتمادا على خوارزمية سرب الطيور (PSO) للاسترداد التلقائي للمفتاح. يقدم أيضًا نظام تحليل التشفير القائم على PSO مع اقتراح عمل لتحسين PSO ، باستخدام التحمية المحاكاة (SA) ، ويقدم مقارنة لنتائج تحليل التشفير التي تم الحصول عليها بواسطة التقنية المقترحة ، والتي تسمى (MPSO) مع PSO الكلاسيكية والخوارزمية الجينية. كما هو موضح حول هذه الخوارزميات يمكن استخدامها في تقليل عدد المحاولات أو التجارب الخاصة بفضاء المفتاح ، والتي تكون ضرورية للوصول إلى الحل الأمثل (القيم الأولية الدقيقة للمسجلات الزاحفة الخطية (LFSR)) وزيادة سرعة عملية البحث للعثور على أفضل حل. بناءً على النتائج التي تم الحصول عليها ، فقد ثبت أن هذه الخوارزميات فعالة في إيجاد الحل الأمثل وأن طريقة MPSO تعمل بشكل أفضل مقارنة مع PSO من حيث الوقت والدقة. تتعرض أنظمة التشفير الانسيابي المعروفة للهجوم من خلال طريقتين للحوسبة الناعمة واستخدمت كحالات دراسة وهما من أنظمة التشفير الانسيابي Geffe و Brüer و Shrinking .

1. Introduction

Cryptanalysis is the science of extracting the cleartext from a cipher without knowing the key. It is a method of converting a ciphertext to the plaintext without access to the key [1].
This paper considers a new method to cryptanalysis based on soft computing techniques to attack all types of stream cipher (linear and nonlinear systems) with different lengths of ciphertext and different lengths of LFSRs.
Firstly, in the case of stream cipher, the key extracted from a bit sequence generator and the sequence are be mixed with the clear text to generate ciphertext and the output of running key called Key Generator (KG)[2][3].
The cryptanalysis has been the a major area of research. Many researchers worked extensively in this subject, such as: Nalini N. et al, 2008, who establishes the ability of a couple of optimization heuristics to attack or cryptosystems analysis studies, in the paper Data Encryption Standard (DES) were selected [4]. Sarab M. Hameed et al., 2010, this work explained a new work for cryptanalysis transposition cipher depending upon Particle Swarm Optimization (PSO) [5]. Hussein. A. Mohammed, 2010**,** this paper focus to apply cryptanalysis attack algorithms upon stream cipher systems using plaintext attack or part from it [6] .
Benjamin Nicholas Ferriman, 2012, in his paper explained the RC4 algorithm and made a new method for attacking state register of RC4s. This is needed in any methods communication to enhance the comprehensive search process for a swapping using GA and PSO algorithms [7]. As for Ali A. Abd et al., 2013, this work proposed a new technique to attack depending on the implementation of direct search algorithm called genetic algorithm [8].
In this paper, a new technique (MPSO) in cryptanalysis stream ciphers system using ciphertext only attack is proposed to detect the validity of these techniques in the cryptanalysis scope. Also, to compare between the results obtained by these techniques in cryptanalysis based on the accuracy and time that is needed to determine the initial setting of the attacked generator with different lengths of ciphertext and different lengths of LFSRs using soft computing techniques.
This paper is structured as follows: In section 2, a brief explanation is included of the swarm intelligence techniques that are being used in proposed system, which in **"Soft Computing Techniques".** Section 3 present an explanation of the proposed system and its structure. While section 4 **includes** the execution , analysis and testing **"Experimental Results "**. The "**Conclusion**" and future work are in section 5.

2. Soft Computing Techniques

In the collective knowledge leading to swarm intelligence, the most critical aspect is social learning [9].The term Soft Computing (SC) consisting of many techniques that include Probabilistic Reasoning (PR), Genetic Algorithms (GA), Belief Network (BN) and aspect of Learning Theory (LT), Chaotic Systems (CS). SC techniques are different from analytical method in that they exploit computing techniques that can represent imprecise, uncertain and vague concepts. Swarm Intelligence (SI) heuristics are based on a population of individuals which represents candidate solutions [10]. Analytical, also called hard computing, techniques on the other hand use binary logic, deterministic reasoning crisp classification .

In their editorial review, (Hoffmann, 2005) observed that:" compared with hard computing methods that only treat with rigor, certainty and precision. Soft computing is effective in sub-optimal or acquiring imprecise but competitive and economic solutions. It takes advantage of intuition, which that's mean the subjective thinking and human mind-based intuitive is implemented here". Techniques or methods in SC are capable of handling non-linearity and they offer computational naivety in constraint with the analytical techniques. These methods have been shown to be capable of managing enormous amounts of data and information and mimic biological systems in learning, linguistic conceptualization, generalization capabilities and optimization [11].

2.1 Particle Swarm Optimization

Swarm Intelligent is a type of artificial intelligence that depends on the behavior of animals living in groups and having some capability to interact with another and with the environment in which they are inserted. The particles alternate the information to correcting their positions and velocities by using the received information. The evolutionary principles can be implemented in search and optimization process [12].

Velocity calculated as follows:

$$v^i_{j+1} = v^i_j + c1\ r1\ (p^i_j - x^i_j) + c2\ r2\ (p^g_j - x^i_j) \qquad j=1,..n \ldots (eq.1)$$

With the *Position* as follows:

$$x^i_{j+1} = x^i_j + v^i_{j+1} \qquad j=1,..,n \ldots (eq.2)$$

Where:

x^i_j : Particle position

v^i_j : Particle velocity

p^i_j: Best position found by jth particle (personal best)

p^g_j: Best position found by swarm (global best)

c1 and c2 : are the cognitive (individual) and social, (group) learning, rates, respectively.

r1 and r2 : are uniformly distributed random numbers in the range 0 and 1.

2.1.1 Steps of PSO Algorithm

1- Setting of PSO Parameters
2- Create first swarm
3- Extract the fitness of all particles
4- Save the best fitness value in all particles
5- Save global best particle
6- Particles met the stopping points?
7- If Step 6 is yes, then end
8- Else, update the position, update the velocity of particle, then go to step3.

2.2 Simulated Annealing (SA)

Simulated annealing, investigated by Kirkpatrick et al. [13] Simulated annealing is dependent upon the scheme of annealing [14]. The algorithm of simulated annealing is as following: [14]

T is a controls parameter named computational temperature, which controls the magnitude of the unrest of the energy function E(x). The probability of a state modify is specified by the Boltzmann distribution (P) of the energy difference of the two states while the δ means the delta value

Algorithm (1): Simulated Annealing (SA) algorithm

1. Initialize the system configuration. Randomize X(0).
2. Initialize T with a large value.
3. **Repeat:**

a. **Repeat:**

i. Apply random perturbations to the state $x = x + \delta x$.

ii. Evaluate δ E(x)= $E(x + \delta x) - E(x)$

if $\delta E(x) < 0$, keep the new state;

Otherwise, accept the new state with probability $P = e^{-\delta E / T}$.

Until number of accepted transitions is below a threshold level.

b. Set $T = T -$ T.

Until T is small enough.

3. Employing PSO and MPSO in Attacking Stream Ciphers

In this work, the advancement of modern technologies and methods of data analysis, the cryptanalysis operation can be programmed. Computational swarm algorithms and evolutionary algorithms may be some techniques of programming process [15]. The designed system contains many components as shown in Figure 1, the figure illustrates the use of the techniques, PSO and MPSO to attack stream cipher using ciphertext only for different lengths of *LFSRi*. The required parameters of each method are listed for each other separately (See Figure 1).

3.1 Cryptanalysis Using Particle Swarm Optimization

This paper compares the results of Particle Swarm Optimization algorithm with the results of the MPSO as it is considered another form of evolutionary algorithms. PSOs are commonly used to model swarm formations that are founds in nature like the patterns of bees or schools of fish. There are two basic properties in Particles are given velocities and positions that are into the search space of the problem. They implement these velocities for any particle in the swarm to move them by the search space at different rates. One major difference between PSO and Genetic Algorithm (GA) is the fact that an individual particle, also storage it's better solution as well as a global best solution of the swarms (this is mildly comparable to elitism in GA). This method is not predicted to be the more successful approach of cryptanalysis of stream ciphers.

3.1.1 Solution Representation

In the initial population, the key size is represented as randomly generated numbers between {0, 1} (binary representation) that is according to the swarm size

To evolve solutions (keys), each particle is updated according to two values:

1. Pbest(p^i_j): Best position found by jth particle
2. $gbest_i$(p^g_j): Best position found by swarm

3.1.2 Fitness Evaluation

In the evaluation operation, the fitness operation contains the fitness function calculation process for any particle (selected key) should be calculated for all generation.

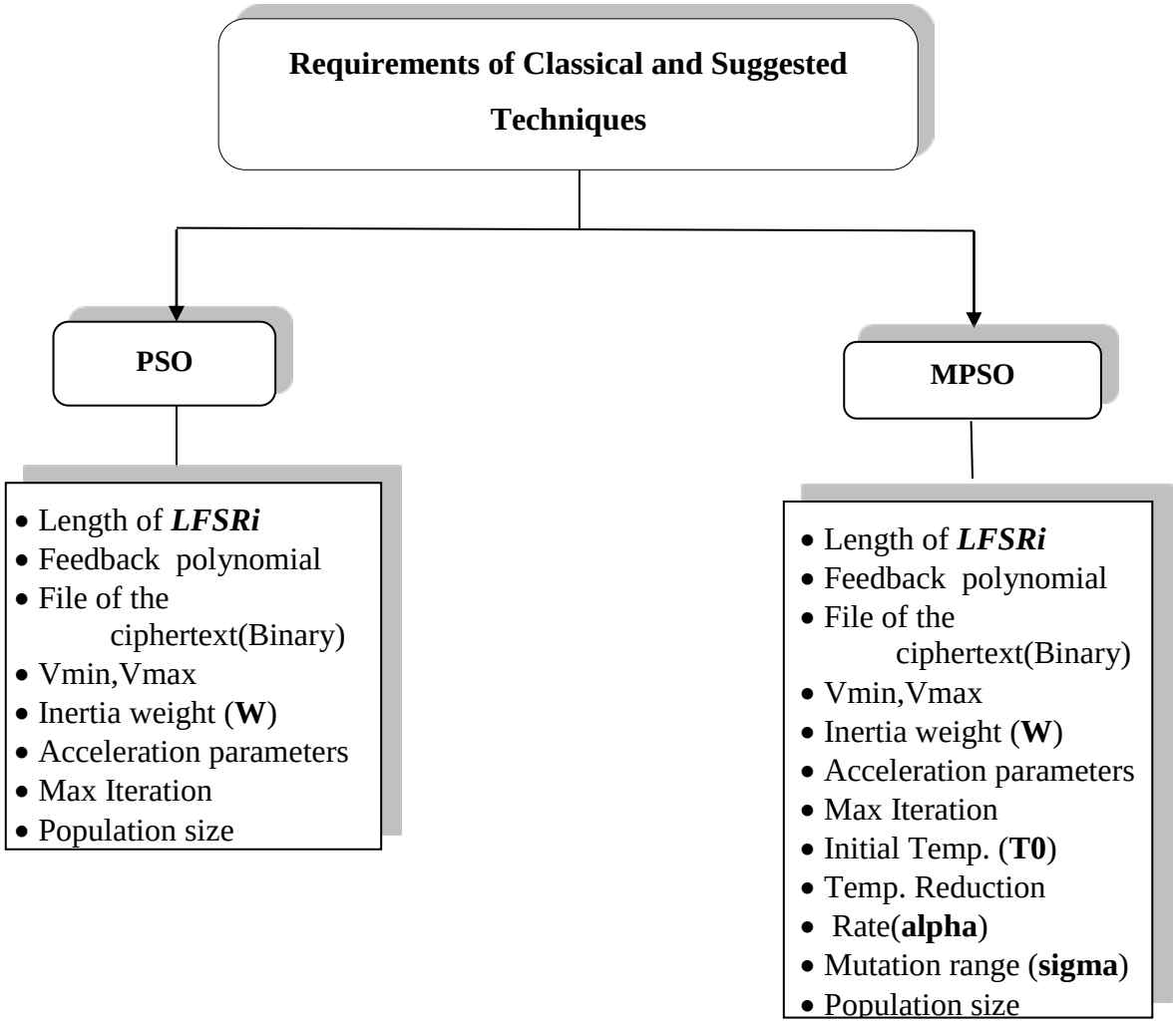

Figure 1: The Proposed System Components.

3.1.3 PSO Parameter Configuration

There are major parameters used in the designed system, these parameters are detailed in Table1:

Table 1: Configuration of PSO Parameters

Term	Meaning	Range
Swarmsize	Number of individuals in the swarm	[20-200]
LengSt	Key Length	[15,19]
LEN	Lengths of LFSRs	{3,5,7},{3,5,11}
T	Feedback Polynomials	Stated in Table 2
Text length	Ciphertext Length	[10-150]
Max Iteration	Maximum Iteration	[100-300]
V_{max}	Maximum Velocity	4
V_{min}	Minimum Velocity	$-V_{max}$
W_{max}	Maximum Inertia Weight	0.9
W_{min}	Minimum Inertia Weight	0.4
C_1,C_2	Acceleration parameter	[0.5-2]
R_1,R_2	Random number between [0,1]	[0-1]

3.2 Cryptanalysis Using Modified Particle Swarm Optimization

Automated attacks can be formed using swarm intelligence techniques that can search the ciphers key in an acceptable amount of time [16]. Modified PSO (MPSO) is a relatively new approach to attack stream cipher systems. In MPSO proposed cryptanalysis system is applied ciphertext only attack to cryptanalysis the Geffe ,Bruer and shrinking systems as cases . For this study, three keystream generator systems of nonlinear stream cipher were be attacked; Geffe , Bruer and shrinking systems were used as the case study with different LFSRs; 3, 5, 7 and 3,5,11 and different feedback polynomial as shown in Table 2.

3.2.1 Steps of MPSO Cryptanalysis

1- Input

- File of ciphertext (binary).
- PSO parameters and attacked system (Geffe or Bruer generator parameters).

As follow:

- PSO parameters: Setting of PSO parameters: Swarmsize, lengst (keylength), Max Iteration, Position(x), Velocity (v).
- Attacked system (Geffe or Bruer) Parameters: Lengths of LFSRs, Feedback Polynomials (T).
- Length of ciphertext (N).

2- Setting of SA parameters

- Initial Temperature (T_0),
- Temperature Reeducation Rate (Alpha),
- Mutation Range (Sigma).

3- Setting PSO Parameters

- Acceleration parameter (C1, C2),
- Inertia Weight (W),
- Random Number (R1, R2).

4- Initialize PSO Parameters

- Produce group of particles (indicate solution or key) in randomly form, each particle means a solution or key.

5- Initialize Swarm

6- Evaluate initial population as follow:

- Key Generation: Generate keystream from generators will be attacked (Geffe or Bruer or shrinking systems).
- Fitness Calculation**:** process the fitness function to any particle in population (indicate key or optimal fitness)
- Sorting: now, re-arrange the particles (indicate keys) in descending order depending upon their Fitness values.

7- Swarm Evaluation

- For each particle process (Update) particles velocity by using equation (eq.1) and particle position according to equation (eq.2).
- Fitness Function Calculation: Calculate the fitness value for each particle after updating (X',V').
- If the fitness value of the current position X' is better than the best fitness value ($pbest_i$) in history X, then set the current position as the new $pbest_i$.
- The parameters will be computed by Simulated Annealing (Delta and then P), if Rand (Random number between 0 and 1) <= P, then,
- Set the current position as the new local best $pbest_i$ and choose the particle (candidate (key)) that has the best fitness value from all the particles in swarm as the Global best $gbest_i$.

8- Evaluate Iteration

- Evaluate by repeating step 6 and 7 for each iteration

9- Stopping MPSO

- Terminate MPSO after condition has been met either to get optimal fitness or to reach the maximum iteration and save the optimal particle as a candidate key.
-

Table 2: Feedback Polynomials Used in Attacked Generators.

Bits	Feedback polynomial	period
n		2^n -1
3	X^3+X^2+1	7
5	X^5+X^3+1	31
7	X^7+X^6+1	127
11	$X^{11}+X^9+1$	2047

Table 3: Configuration of MPSO Parameters.

Symbol	Meaning	Value
Swarmsize	Particles in the Swarm	**[20-200]**
LengSt	Length of the Key	**[15,19]**
LEN	Lengths of LFSRs	**{3,5,7},{3,5,11}**
T	Feedback Polynomials	**Stated in Table 2**
Text length	Text Length	**[10-100]**
Max Iteration	Maximum Iteration	**[100-300]**
$\mathbf{V_{max}}$	Maximum of velocity	**4**
$\mathbf{V_{min}}$	Minimum of velocity	$\mathbf{-V_{max}}$
$\mathbf{W_{max}}$	Maximum Inertia Weight	**0.9**
$\mathbf{W_{min}}$	Minimum Inertia Weight	**0.4**
$\mathbf{C_1,C_2}$	Acceleration Parameter	**[0.5-2]**
$\mathbf{R_1,R_2}$	Random number between [0,1]	**[0-1]**
$\mathbf{T_0}$	Initial Temperature	**0.1**
$\mathbf{Var_{max}}$	Upper bound of decision variables	**10**
$\mathbf{Var_{min}}$	**Lower bound of decision variables**	**-10**

4. Implementation and Experimental Results:

The following Tables from 6 to 9 states the results of the cryptanalysis system for different lengths of texts and different length of attacked registers for key lengths {15,19}. Table 4 explains some of notations used in cryptanalysis process, while Table 5 states the best fitness for different plaintext length. Also, in this paper the Brute Force Attack was applied for stream cipher using Geffe generator system for different lengths TxtLen={100,40,35,30,20 and 10} with KeyLen=15 as shown in Table 9.

Table 4: Notations of Cryptanalysis Implementation Process.

Symbol	Meaning
Popsize	**Population size**
MaxIter	**Maximum Iteration**
BF	**Best Fitness**
T/sec	**Time/second**
T.T/sec	**Total Time/second**
Iter_Num	**Iteration Number**

Table 5: The Best Fitness for Different Plaintext.

TxtSize	**150**	**40**	**35**	**30**	**20**	**10**
BF	0.6301	0.6188	0.6179	0.6000	0.6062	**0.6250**

Table 6: Results of Applying PSO and MPSO for Text length=150.

Population size	Max Iteration	GA				PSO				MPSO			
		BF	**T/ sec**	**T.T/ sec**	**Gen _ Num**	**BF**	**T/ sec**	**T.T/ sec**	**Iter _ Num**	**BF**	**T/ sec**	**T.T/ sec**	Iter_ Num
20	100	0.5869	2.09	15.91	13	0.6188	0.23	16.31	1	0.5757	0.35	10.70	**1**
	300	0.5496	0.67	48.30	4	0.5520	0.58	47.20	1	0.5682	0.32	33.39	**1**
100	100	0.6301	3.21	79.33	3	0.5656	1.21	82.32	2	0.6301	1.43	62.55	**1**
	300	0.6023	4.84	238.36	6	0.5556	0.82	240.98	1	0.6301	1.45	205.47	**1**
200	100	0.6301	4.86	163.88	3	0.5850	3.29	160.47	2	0.5769	2.89	140.13	**1**
	300	0.5694	5.62	476.26	4	0.5760	1.58	460.32	1	0.6301	2.80	420.01	**1**

Table 7: Results of Applying PSO and MPSO for Text length=50

Population size	Max Iteration	GA				PSO				MPSO			
		BF	**T/ Sec**	**T.T/ sec**	**Gen_ Num**	**BF**	**T/ sec**	**T.T/ sec**	**Iter_ Num**	**BF**	**T/ sec**	**T.T/ sec**	Iter_ Num
20	100	0.6062	0.20	3.23	6	0.6062	0.05	2.99	1	0.6250	0.07	3.20	**1**
	300	0.5787	0.14	9.64	4	0.5875	0.10	9.73	2	0.6062	0.09	9.08	**1**
100	100	0.6188	0.95	15.69	6	0.6062	0.19	15.22	1	0.5938	0.30	14.39	**1**
	300	0.6062	0.62	46.92	4	0.6000	0.18	44.22	1	0.6062	0.28	48.55	**2**
200	100	0.6422	1.46	33.50	4	0.6125	0.31	28.44	1	0.6062	0.60	29.30	**1**
	300	0.6188	0.97	100.57	3	0.6062	0.30	87.70	1	0.6062	0.57	100.76	**1**

Table 8: Results of Applying PSO and MPSO for Text length=30.

Population size	Max Iteration	GA				PSO				MPSO			
		BF	**T/ sec**	**T.T/ sec**	**Gen-Num**	**BF**	**T/ sec**	**T.T/ sec**	**Iter_ Num**	**BF**	**T/ sec**	**T.T/ sec**	Iter-Num
20	100	0.6250	0.11	2.02	5	0.6250	0.05	2.10	1	0.6250	0.03	1.75	**1**
	300	0.6750	0.83	5.87	42	0.6625	0.04	5.60	2	0.6250	0.05	6.15	**1**
100	100	0.6250	1.96	10.99	18	0.6625	0.09	8.50	1	0.6375	0.22	9.19	**1**
	300	0.6625	27.25	33.36	250	0.6625	0.81	24.40	1	0.6250	0.22	26.10	**1**
200	100	0.6750	12.25	27.34	45	0.6625	0.18	16.80	2	0.6250	0.36	17.51	**1**
	300	0.6250	13.92	82.10	51	0.6250	0.35	50.47	2	0.6250	0.38	51.93	**1**

Table 9: Results of Applying PSO and MPSO for Text length=10.

Population size	Max Iteration	GA				PSO				MPSO			
		BF	**T/ sec**	**T.T/ sec**	**Gen _ Num**	**BF**	**T/ sec**	**T.T/ sec**	**Iter _ Num**	**BF**	**T/ sec**	**T.T/ sec**	Iter_ Num
20	100	0.5906	2.11	6.98	31	0.5656	0.17	6.97	2	0.5563	0.14	6.69	**1**
	300	0.5938	0.21	19.81	3	0.5875	0.18	22.41	2	0.5656	0.22	19.96	**1**
100	100	0.5625	25.48	33.99	75	0.6262	0.70	32.75	2	0.5750	0.65	32.56	**1**
	300	0.6262	20.30	101.30	35	0.6188	0.41	97.51	1	0.6250	0.55	98.70	**1**
200	100	0.6188	30.10	65.50	35	0.6125	0.66	64.53	2	0.6188	0.22	64.46	**1**
	300	0.6188	5.60	465.30	4	0.6188	1.60	463.10	1	0.6188	0.56	192.10	**1**

Table 10: Results of applying Brute Force Attack for Geffe Generator

TxtLen	Brute Force Attack		
	BF	**T/sec**	**T.T/ sec**
100	0.6301	19.85	180.71
40	0.6188	15.09	166.99
35	0.6179	19.42	146.79
30	0.6000	42.43	140.15
20	0.6062	13.01	530.15
10	0.6250	23.17	107.97

5. Conclusions

1. The Three soft computing techniques (GA, PSO and MPSO) confirm remarkable success in cryptanalysis of a stream cipher systems and determining the initial setting of the attacked generators (Geffe , Bruer and Shrinking systems).
2. In Siegenthaler method only Nonlinear stream cipher systems were attacked with different lengths of registers separately using correlation method, while in this work the MPSO attacks the entire system (all registers at the same time).
3. This work suggests an innovative approach called Modified PSO (MPSO) to recover the right solution for the attacked generator.
4. The techniques (GA, PSO and MPSO) shows remarkable success in cryptanalysis when keysize {15, 19} are used.
5. As shown in Tables 6 to 9 the implementation of GA, PSOi and MPSO when attacking stream ciphers, it can be concluded that 3 iterations were enough to detect the good or optimal for both Geffe , Bruer and Shrinking systems

6. The accomplishment or implementation of GA is less than the production of the other two techniques in attacking stream cipher systems in term of time as shown in Tables 6 to 9.

7. The analytical study of applying GA, PSO and MPSO gives the following results:
- As shown in Tables 6 to 9, GA, PSO and MPSO in the text lengths;150,50,30, and 10 are obtained in population size=100 and Max Iteration=300, and the results of MPSO is considered the best in the term of time than other techniques.
- MPSO considered the best in term of time, also MPSO finds the optimal solution.
- As shown in Tables 6,7,8 and 9, MPSO considered in term of time the best than other techniques.

8. From above (point 7), population size=200 and Max Iteration=300 in all text lengths is enough to get the right key for the presented techniques, and the results of MPSO is considered the best in term of time than other techniques.
9. Brute Force Attack have been applied for stream cipher (for two case study: Geffe and Bruer), that when text length >= 30 that lead to unique solution (initial setting of LFSRs), and hence we can get real plaintext, while for text length < 30,the brute force attacks gives more than one solution , so we can conclude the real plaintext by decrypting ciphertext as shown in Table 10.
10. Soft computing techniques can be used as a powerful tool in generating pseudorandom sequences with good statistical properties and a high linear complexity and overcome all problems and difficulties that face designers of cipher systems.

6. Suggestions for Future Scope:

Several areas for future work are presented as result of this work:
1. Additional, or another, fitness functions may be inspected that could give superior results than those reported here.
2. In addition, more methods or operators of soft computing techniques may be modified. For example, in GA use of another type of selection might be investigated like Rank-Based Selection and Tournament-based Selection or another type of crossover, such as two-point crossover.
3. Use of a parallel soft computing technique in the cryptanalysis of stream cipher systems might be investigated.
4. Use of soft computing techniques in cryptanalysis of stream cipher systems can be used to find the primitive feedback polynomials of the *LFSRs*.
5. Using another artificial intelligent approach or hybridization algorithms of it such as ant Colony,Bees etc. in cryptanalysis and encryption stream cipher or another types of algorithms.

7. References:

[1] R.N. Jawad and F.H. Ali . **2020**. "Using Evolving Algorithms to Cryptanalysis Nonlinear Cryptosystems", *Baghdad Science Journal*, 17(2),0682-0688. https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/3985.

[2] Ghazi, A. A., & Ali, F. H. 2018. Design of New Dynamic Cryptosystem with High Software Protection. Iraqi Journal of Science, 59(4C), 2301-2309. Retrieved from https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/578.

[3] Ghazi, A. A., & Ali, F. H. (2018). Robust and Efficient Dynamic Stream Cipher Cryptosystem. Iraqi Journal of Science, 59(2C), 1105-1114. Retrieved from https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/401.

[4] N. Nalini and G. Rao .**2008**. "Cryptanalysis of Block Ciphers via Improvised Particle Swarm Optimization and Extended Simulated Annealing Techniques". *International Journal of Network Security*, 6(3): 342-353.

[5] S.M. Hameed and D.N. Hmood .2010 . "Particles Swarm Optimization for the Cryptanalysis o Transposition Cipher" . *Journal of Al-Nahrain University* , 13 (4): 211-215.

[6] H.A. Mohammed .**2010** . "Cryptanalysis of Stream Cipher System Using Particle Swarm Optimization Algorithm" . *Journal of Kerbala University*, 6 (2):384-394.

[7] B.N. Ferriman. **2013** . "Cryptanalysis of the RC4 Stream Cipher using Evolutionary Computation Methods". Master Thesis. University of Guelph .

[8] A.A. Abd, H.A Younis and W.S. Awad . **2013**. "Attacking of stream Cipher Systems Using a Genetic Algorithm". *Journal of University of ThiQar*, 8(3):188-194.

[9] A.K. Sabonchi and B. Akay. **2020** . "A Binomial Crossover Based Artificial Bee Colony Algorithm for Cryptanalysis of Polyalphabetic Cipher". *Research-gate*, 27(6): 1825-1835, Available from: A Binomial Crossover Based Artificial Bee Colony Algorithm for Cryptanalysis of Polyalphabetic Cipher (srce.hr), https://doi.org/10.17559/TV-20190422225110.

[10] M. Tahar and A. Zidani .**2018.** "Swarm intelligence algorithms in cryptanalysis of simple Feistel ciphers". *Int. J. Information and Communication Technology*, 13(1) :114-138.

[11] X.S Yang and Z.H Cui .**2014** ."Bio-inspired Computation: Success and Challenges of IJBIC" . *Int. J. Bio-Inspired Computation* , 6(1):1-6 .

[12] K.W. Wong , W.S. Yup, D.C, Wong, R.C. Phan and B.M. Goi. **2020**. "Cryptanalysis of genetic algorithm-based encryption scheme". *Springer*, 79(1): 25259–25276. https://doi.org/10.1007/s11042-020-09191-z.

[13] M.M. Mafarja and S. Mirjalili .**2017**. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. Neurocomputing ELSEVIER, 260: 302–312. https://doi.org/10.1016/j.neucom.2017.04.053.

[14] I.K. Ali and A.G. Mahmod .**2015.** Hybrid Bees Algorithm With Simulated Annealing for Cryptanalysis of Simple Substitution Cipher. *Journal of Babylon University*, 23(22):565-574.

[15] M. Din, S.K. Pal and S.K. Mottoo .**2020**. " Madan S. A Hybrid Computational Intelligence-based Technique for Automatic Cryptanalysis of Playfair Ciphers*". Defense Science Journal*,70 (6):612-618, DOI: 10.14429/dsj.70.15749.

[16] A.Jain, S.K. Vishwakarma ,P.C. Sharma and N.K. Gupta. **2021**. A Review on Swarm Intelligence Techniques in Automated Cryptanalysis of Classical Substitution Cipher IOP Conf. Series Mater. Sci. Eng. 1099, doi:10.1088/1757-899X/1099/1/012047.