Cryptanalysis of Vigenere Cipher using Particle Swarm Optimization with Markov chain random walk

Aditi Bhateja
Student, Department of Information Security
Ambedkar Institute of Advance Communication Technologies & Research
Delhi-110031, India
aditibhateja89@gmail.com

Shailender Kumar
Ambedkar Institute of Advance Communication Technologies & Research
Delhi-110031, India
shailenderkumar@aiactr.ac.in

Ashok K. Bhateja
Senior Scientist
Scientific Analysis Group
Defence Research & Development Organization
Delhi- 110054, India
akbhateja@gmail.com

***Abstract* - Vigenere cipher is a polyalphabetic substitution cipher with a very large key space. In this paper we have investigated the use of PSO for the cryptanalysis of vigenere cipher and proposed PSO with Markov chain random walk in which some of the worst particles are replaced with new better random particles to enhance the efficiency of PSO algorithm. Based on our experimental results, it is shown that the proposed algorithm is more effective than PSO for the analysis of Vigenere cipher**.

Key words: Cryptanalysis; Vigenere cipher; Particle Swarm Optimization; Markov chain; random walk.

I. INTRODUCTION

Cryptography is a complex and mathematically challenging field of study. It involves taking some data or message with the goal of hiding the meaning of the message [1] i.e. unreadable by unauthorized parties (interceptor). Before the message becomes encrypted it is referred to as the plain text. Once a message becomes encrypted it is then referred to as the cipher text [2]. The study of cipher text in an attempt to restore the message to plaintext is known as cryptanalysis. Cryptanalysis is equally mathematically challenging and complex as cryptography.

Cryptosystems are broadly divided into symmetric and asymmetric cryptosystems. A symmetric cryptographic system is a system involving two transformations: one for the originator and one for the recipient, both of which make use of either the same secret key (symmetric key) or two keys easily computed from each other. An asymmetric cryptographic system is a system involving two related transformations: one defined by a public key (the public transformation), and another defined by a private key (the private transformation), with the property that it is computationally infeasible to determine the private transformation from the public transformation. The most widely known classical symmetrical cipher is the Vigenere cipher.

Vigenere cipher is a polyalphabetic substitution cipher, proposed by Blaise de Vigenere in the sixteenth century. A polyalphabetic substitution cipher involves the use of two or more cipher alphabets. Instead of there being a one-to-one relationship between each letter and its substitute, there is a one-to-many relationship between each letter and its substitutes. A simple Vigenere cipher [3] of period t, over an s-character alphabet, involves a t-character key k_1, k_2 . . . k_t. The mapping of plaintext $m = m_1, m_2$ ... to ciphertext $c = c_1, c_2,\ldots$ is defined on individual characters by $c_i = m_i + k_i \bmod s$, where subscript i in k_i is taken modulo t (the key is re-used). The computational complexity to analyze vigenere cipher is s^t. Generally the number of characters, s, is 26. This number is far too large to allow a

brute force attack even on the fastest of today's computers. However, because of the properties of the vigenere cipher they are relatively easy to analyze [4, 5].

Mehmet E. Dalkilic, Cengiz Gungorde signed and implemented interactive cryptanalysis software based on the Kasiski test, and a novel use of the Index of Coincidence (IC) concept mentioned in [6]. They observed that using this method, cryptanalysis of vigenere cipher is possible for very short text lengths where classical cryptanalysis methods fail.

Conventional computing paradigms often have difficulty dealing with real world problems, such as key search where domain size is very large. Natural systems inspired by several natural computing paradigms have evolved to solve such problems where conventional computing techniques perform unsatisfactorily. Particle swarm optimization is a heuristic global optimization search method proposed by Kennedy and Eberhart [7] in 1995. It is developed from swarm intelligence and is based on the research of bird and fish flock movement behavior. While searching for food, the birds are either scattered or go together before they locate the place where they can find the food. While the birds or particles are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird is perceptible of the place where the food can be found, having the better food resource information. Particles move through the search space using a combination of an attraction to the best solution that they individually have found, and an attraction to the best solution that any particle in their neighbourhood has found.

Uddin, Mohammad Faisal [8] proposed Cryptanalysis of Simple Substitution Ciphers Using Particle Swarm Optimization. They showed that PSO provides a very powerful tool for the cryptanalysis of simple substitution ciphers using a ciphertext only attack. Heydari et.al [9] proposed Genetic Algorithm based scheme for cryptanalysis of transposition cipher with key lengths up to 25. Vimalathithan et.al [10, 11] applied PSO based Computational Intelligence technique for the cryptanalysis of simplified-DES and simplified AES.

In this paper we have investigated the use of Particle Swarm Optimization for the cryptanalysis of vigenere cipher assuming that the length of the key is known and we have also proposed PSO with Markov chain random walk through which it is possible to obtain the better performance for this kind of problem.

II. VIGENERE CIPHER

The Vigenere Cipher, proposed by Blaise de Vigenere, is a polyalphabetic substitution based on the Vigenere tableau of size 26×26. The method of obtaining the sequence of cipher text letters involves choosing a keyword. If the plaintext message is longer than the keyword, then the sequence is obtained by repeating it as many times as is necessary. Thus the period is the length of the keyword. One important difference between Vigenere cipher and the other polyalphabetic ciphers is that Vigenere key stream does not depend on the plaintext characters, it depends only on the position of character in the plaintext. Consider the following example:

Plaintext	O	N	A	P	L	A	N	E	T	H	E	P	L	A	N	E	I	S	D	U	E
Keyword	M	I	L	K	M	I	L	K	M	I	L	K	M	I	L	K	M	I	L	K	M
Ciphertext	A	V	L	Z	X	I	Y	O	F	P	P	Z	X	I	Y	O	U	A	O	E	Q

The set of related mono alphabetic substitution rules makes use of 26 Caesar Ciphers with shifts 0 to 25. The cipher text is modelled as C = (P + K) mod 26 where C is the cipher text, P is plaintext and K is key word letters. Similarly we can obtain the plain text P if we know the key by the formula P = (C – K) mod 26. The strength of Vigenere cipher is the unknown key. To analyze the Vigenere cipher, the first step is to determine the length of the key. The next step is to find the actual key. To get the actual key, we have used PSO algorithm and modified it to obtain better performance.

Methods to determine the length of the key

The Kasiski and Friedman tests can help to determine the key length. Riedrich Kasiski in 1863 published Kasiski Test [12]. In Kasiski test, the cryptanalyst searches for repeated text segments, of at least three characters, in the ciphertext. Then, the distances between consecutive occurrences of the strings are likely to be multiples of the length of the keyword. Finding more repeated strings narrows down the possible lengths of the keyword, since we can take the greatest common divisor of all the distances. The key length is the multiple of the greatest common divisor. Friedman Test developed in 1925 by William Friedmanis, a probabilistic test that can be used to determine

the likelihood that the ciphertext message produced comes from a monoalphabetic or polyalphabetic cipher. This technique uses the index of coincidence, to measure the unevenness of the cipher letter frequencies. By knowing K_p (probability that any two randomly chosen source-language letters are the same, in case of English $K_p = \sum_{i=0}^{25} {p_i}^2 \approx 0.067$, p_i is the probability that both the alphabets are *i*.) and K_r (probability of a coincidence for a uniform random selection from the alphabet, in case of English K_r =1/26), the estimated key length (*l*) can be:

$$l = \frac{K_p - K_r}{K_o - K_r}$$

Where K_o (observed coincidence rate) is:

$$K_o = \frac{\sum_{i=1}^{c} n_i(n_i - 1)}{N(N-1)}$$

Where *c* is the size of the alphabet (26 for English), *N* is the length of the text, and n_1 to n_c are the observed cipher text letter frequencies.

III. PARTICLE SWARM OPTIMIZATION

PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms, developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer) [7]. It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in an N - dimensional space which adjusts its position according to its own flying experience as well as the flying experience of other particles. Each particle keeps track of its personal best position (*pbest*) and best position obtained so far by any particle (*gbest*). The basic concept of PSO lies in accelerating each particle towards its *pbest* and the *gbest* locations, with a random weighted acceleration at each time step. The two basic equations which govern the working of PSO are that of velocity vector and position vector given by:

$$\mathrm{v}_i^{t+1} = w.\mathrm{v}_i^t + C_1.r_1 \times (pbest_i - \mathrm{x}_i^t) + C_2.r_2 \times (gbest_i - \mathrm{x}_i^t)$$

$$\mathrm{x}_i^{t+1} = \mathrm{x}_i^t + \mathrm{v}_i^{t+1}$$

where,
v_i^t : velocity of agent *i* at iteration *t*,
w : weighting function,
C_1: Self confidence weighting factor,
C_2: swarm confidence weighting factor,
r_1 and r_2 : uniformly distributed random numbers between 0 and 1,
x_i^t : current position of particle *i* at iteration *t*,
$pbest_i$: personal best of particle *i*,
gbest : global best position of any particle.

IV. FITNESS FUNCTION

To implement fitness function, the frequency of each character in the decrypted text is calculated. This frequency is normalized by dividing it by the total number of characters in the file. This normalized frequency is then subtracted from the expected frequency of the character in normal English text. The absolute value of this difference is taken. The differences for all characters are added together. The bigram is an extension of unigram to two characters. Now rather than calculating frequency of individual character, we calculate frequency of 'pairs' of letters. For example, a pair 'an' will always appear more frequently than pair 'bt'. Again statistics for the frequencies of these pairs are also available. These statistics are compared with the statistics obtained from the decrypted text. The frequency of each pair of letters in the decrypted text is calculated and is normalized. The absolute value of the difference of the normalized frequency and the expected frequency of the pair in normal English text is calculated. The differences for all pairs are added together. The normalization takes care that this value always lies between 0 and 1. To implement fitness function, weighted sum of both the sum of the differences is calculated. The fitness function used based on monogram and bigram is given by:

$$\text{fitness} = 0.23 \times \sum_{i=1}^{26} |SF(i) - OF(i)| + 0.77 \times \sum_{i=1}^{25} |SDF(i) - ODF(i)|$$

Where:
$SF(i)$ is standard frequency of i^{th} monogram in normal English.
$OF(i)$ is observed frequency of i^{th} monogram in decrypted text.
$SDF(i)$ is standard frequency of i^{th} bigram in normal English.
$ODF(i)$ is observed frequency of i^{th} bigram in decrypted text.

Here letters A…Z are referenced by the indices 1…26. If the experimental key is closer to the actual key used then the fitness value will be small otherwise its value will be large. Now this problem has been reduced to an optimisation problem where it is required to reduce the error or minimize the fitness value.

V. PROPOSED ALGORITHM FOR CRYPTANALYSIS OF VIGENERE CIPHER USING PSO WITH MARKOV CHAIN RANDOM WALK

Particle Swarm Optimization [7] might struck at the local optimum point because the search space in the cryptanalysis of the Vigenere cipher is very large. To overcome this problem, we propose a modified PSO algorithm by discarding the worst particles of the swarm using Markov chain random walk that depends on the transition probability. This approach provides a good way to move away from local minimum and to search on the global scale. To be computationally efficient and effective in searching for new solutions, the best solutions found so far should be kept, and increase the mobility of the random walk so as to explore the search space more effectively. More importantly, the walk should be controlled in such a way that particles can move towards the optimal solutions more quickly, rather than wander away from the potential best solutions.

A random walk is a random process which consists of taking a series of consecutive random steps. Mathematically, let S_N denotes the sum of each consecutive random step X_i, then S_N forms a random walk.

$$S_N = \sum_{i=1}^{N} X_i$$

$$\text{i.e.} \quad S_N = S_{N-1} + X_N$$

Where, X_i is a random step drawn from a random distribution. The next state S_N depends on the current existing state S_{N-1} and the motion or transition X_N from the existing state to the next state. If the step length obeys the Gaussian distribution, the random walk becomes the Brownian motion [13]. If step length depends on the transition probability, which indeed has the properties of a Markov chain, i.e. random walk is a Markov chain. Markov chain algorithm starts with an initial solution, and proposes a new solution if the random number generated is less than the transition probability.

Proposed Algorithm for finding the actual key

1. *Initialization of PSO search algorithm parameters*

   The PSO parameters are set in the first step. These parameters consist of number of particles (N_p), the size of the key (N_d), maximum number of Iterations (N_t), Self-confidence factor (C_1), Swarm confidence factor (C_2), and inertia weight (w).

2. *Initialization of discrete birds or population*

   a) For cryptanalysis of vigenere cipher: the initial positions of the particles are determined by randomly choosing the permutations of size N_d, sampled uniformly at random from integers 0 to 25.
   b) Initialize velocity of each particle using:

$$v_i = v_{min} + (v_{max} - v_{min}) \times rand$$

where:
v_i is velocity of i^{th} particle,
v_{max} is maximum velocity,
v_{min} is minimum velocity,
rand is a random number between 0 and 1.

3. *Calculate fitness function value*

For each particle
a) Decrypt the cipher text using the position of the particle as the key.
b) Find the fitness function value of the text obtained in step 3 (a).

4. *Update velocity and position of the particles*

$$v_i^{t+1} = w.v_i^t + C_1.rand_1 \times (pbest_i - x_i^t) + C_2.rand_2 \times (gbest_i - x_i^t)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Calculate the fitness function value of each particle as discussed in step 3.

5. *Discarding worst particles using Markov Chain random walk*

For each component of each solution define the transition probability:

$$P_{ij} = \begin{cases} 1 & if\ rand < pd \\ 0 & if\ rand \geq pd \end{cases}$$

where *rand* is a random number in [0, 1] interval and *pd* is the discarding probability. Existing particles are replaced by the newly generated ones from their current positions (if the newly generated is better) through random walks with step size such as:

$$S = rand.(x(rp1(n),:) - x(rp2(n),:))$$

$$x_i^{t+1} = x_i^t + S.* P$$

Where, *rp*1 and *rp*2 are random permutations functions and *P* is the transition probability matrix.

6. *Termination criterion*

Repeat steps 3 to 5 until termination criterion is satisfied. The maximum number of iterations or the saturation of fitness value of the *gbest* particle is considered as termination criterion of the algorithm.

VI. EXPERIMENTAL RESULTS

An English plain text file from various text books and articles is formed. After removing all the punctuations, numerals and structure (sentences/paragraphs marks, space characters, and newline characters) a sample text file consisting of 480526 characters is formed. All plaintexts of varying size, used in the experiments are taken from this file. Particle swarm optimization and the proposed modified scheme (Particle Swarm Optimization with Markov chain random walk) are implemented in MATLAB®. The values of the parameters used in the implementations are as follows:
Self-confidence C_1= 2.05, Swarm confidence C_2 = 2.05 and inertia weight w = 1.

For each key of length 3 to 25, ten vigenere cipher texts were created by randomly selecting the keys and randomly selecting the starting point of plain texts of size 400 characters from the sample text file. Thus total cipher texts created were 22 × 10 = 220. All these cipher texts were analyzed by both PSO and PSO with Markov chain random walk (Modified PSO or MPSO) with 200 particles (for finding the optimal number of particles several experiments were conducted with different number of particles and key sizes), assuming the key length is known.. The results of this experiment are shown Table 1.

TABLE 1: Analysis of vigenere cipher of length 400 characters by PSO and modified PSO with varying key lengths

Key Size	**Average Number of key characters Recovered Correctly**		**Minimum Number of key characters Recovered Correctly**		**Maximum Number of key characters Recovered Correctly**		**Standard Deviation**	
	PSO	***MPSO***	***PSO***	***MPSO***	***PSO***	***MPSO***	***PSO***	***MPSO***
3	2.65	3.00	3	3	3	3	0.00	0.00
4	3.60	4.00	3	4	4	4	0.55	0.00
5	4.50	4.82	4	4	5	5	0.65	0.30
6	4.30	5.10	3	4	6	6	0.37	0.54
7	4.70	6.16	4	5	6	7	0.68	0.90
8	5.20	7.00	4	6	7	8	1.30	0.71
9	6.10	7.24	4	5	7	9	1.56	0.89
10	6.50	8.16	5	6	8	9	2.15	1.25
11	6.50	8.37	5	7	8	10	1.43	1.67
12	6.75	8.50	5	7	8	10	1.65	2.43
13	6.85	8.65	5	7	8	11	1.56	1.63
14	7.20	8.76	6	8	9	11	0.78	1.90
15	7.53	9.50	6	9	9	12	1.78	1.76
16	7.51	9.90	7	7	9	12	1.28	1.22
17	8.24	10.05	7	8	9	13	2.12	1.76
18	8.75	10.30	6	9	10	13	2.43	2.45
19	10.0	11.25	6	9	11	13	1.80	2.50
20	10.50	12.35	7	10	11	14	1.32	1.98
21	10.50	13.52	7	10	12	15	2.21	1.54
22	11.20	14.20	7	10	13	16	2.24	1.76
23	11.20	15.73	8	11	13	18	2.78	2.10
24	12.25	16.20	8	12	14	18	1.80	1.08
25	12.50	17.50	8	12	14	19	2.67	1.90

The number of iterations required till the fitness function value gets saturated, increases with the increase of key size. The number of iterations required with the varying key size for cipher text of length 400 is shown in Fig. 1.

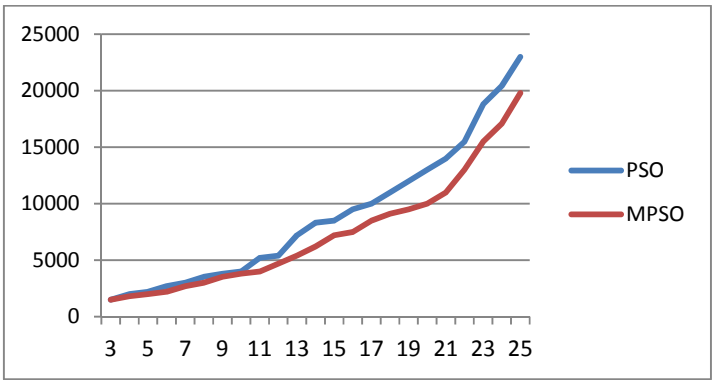

Fig 2. Number of iterations vs key size for cipher text size of 400 characters

Experiments were also performed for ciphertext of size 200 and 600 characters by randomly selecting the keys (of size 5, 10, 15, 20 and 25) and randomly selecting the starting point of plain texts. The results are shown in Table 2.

TABLE 2: Analysis of vigenere cipher of length 200 and 600 characters by PSO and modified PSO with varying key lengths

Vignere Cipher Text length	**Key size**	**Average number of key characters Recovered Correctly**		**Minimum Number of key characters Recovered Correctly**		**Maximum Number of key characters Recovered Correctly**		**Standard Deviation**	
		PSO	***MPSO***	***PSO***	***MPSO***	***PSO***	***MPSO***	***PSO***	***MPSO***
200 characters	5	3.7	4.5	3	4	4	5	0.47	0.2
	10	6.3	7.3	4	5	8	9	1.5	0.74
	15	8.4	8.7	5	7	9	11	0.7	1.52
	20	9.5	10.2	5	8	11	13	1.3	1.78
	25	10.6	13.4	6	10	13	17	2.6	2.17
600 characters	5	4.91	5.0	4	5	5	5	0.21	0.0
	10	7.8	9.12	7	8	10	10	1.9	0.16
	15	9.2	11.61	8	10	13	14	1.27	0.32
	20	12.5	14.29	11	13	15	17	1.81	1.27
	25	15.1	18.81	13	16	17	20	2.31	1.61

From the results it can be concluded that for cipher text of size 400 characters, PSO can find approximately 70% of the key characters for shorter key length (≤ 10) and finds only 50% of the key characters for larger key size (≥ 20). PSO with Markov chain random walk i.e MPSO gives better results than PSO. For shorter key lengths it can find approximately 80% of the key characters and for larger key lengths it finds approximately 70% of the key characters.

VII. CONCLUSION

Particle swam optimization is a meta-heuristic optimization method based on swarm intelligence. PSO provides a powerful tool for the ciphertext only attack of Vigenere ciphers. It is very simple, efficient search technique and needs few parameters. Statistical attacks for Vigenere cipher are not able to find the actual key (or most of the key characters) in real time. The experimental results shows that PSO and PSO with Markov chain random walk are successful in finding the key of Vigenere cipher. For smaller key length (≤ 10), PSO is able to give 70% correct key characters and for larger key lengths (≥ 20), it gives 50% correct key characters. The proposed PSO with Markov chain random walk correctly finds approximately 20% more key characters than PSO and requires less number of iterations i.e. the proposed scheme provides better and efficient results compared to PSO for the cryptanalysis of Vigenere cipher.

REFERENCES

[1] Christof Paar and Jan Pelzl, "Understanding Cryptography", Springer-Verlag Press, Berlin, Germany, 2009.
[2] Mao, W., "Modern Cryptography: Theory & Practice", Upper Saddle River, NJ: Prentice Hall PTR, 2004.
[3] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press 1997.
[4] Douglas R. Stinson., "Cryptography: Theory and Practice", CRC Press, Boca Raton, Florida, USA, 1995.
[5] Henry Beker and Fred Piper, "Cipher Systems: The Protection of Communications", Wiley-Inter Science, London, 1982.
[6] Mehmet E. Dalkilic, Cengiz Gungor "An Interactive Cryptanalysis Algorithm for the Vigenere Cipher", Advances in Information SystemsLecture Notes in Computer Science Volume 1909, 2000, pp 341-351.
[7] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.
[8] Uddin, M.F. & Youssef, Amr M. "Cryptanalysis of simple substitution ciphers using particle swarm optimization", IEEE Congress on Evolutionary Computation, Canada, 2006. pp. 677-80.
[9] Morteza Heydari, Gholamreza Latif Shabgahi Mohammad & Mehdi Heydari, "Cryptanalysis of Transposition Ciphers with Long Key Lengths Using an Improved Genetic Algorithm", World Applied Sciences Journal 21 (8): 1194-1199, 2013.
[10] Vimalathithan. R & M.L.Valarmathi, "Cryptanalysis of Simplified-DES using Computational Intelligence", WSEAS transactions on computers Issue 7, Volume 10, July 2011, pp 210-219.
[11] Vimalathithan R. & M.L.Valarmathi, "Cryptanalysis of Simplified-AES using Particle Swarm Optimisation", Defence Science Journal, Vol. 62, No. 2, March 2012, pp. 117-121.
[12] Forouzan, Behrouz A., "Cryptography and network security", Tata McGraw Hill Education, 2008.
[13] Xin-She Yang, "Nature-Inspired Metaheuristic Algorithms", Second Edition, University of Cambridge, United Kingdom, Luniver Press.

AUTHORS PROFILE

Aditi Bhateja has completed her B.Tech. in Information Technology from Guru Gobind Singh Indraprastha University, Delhi. She is currently pursuing M.Tech. in Information Security at Ambedkar Institute of Advance Communication Technologies & Research, Delhi, India

Shailender Kumar received his B.E (CSE) from MDU, Rohtak and M.Tech. from Rajasthan University. He has more than 11 years of experience in teaching at various esteemed engineering colleges like Delhi College of Engineering, Netaji Subhash Institute of Technology etc. Currently he is working as Associate professor at AIACTR Delhi. His area of interest is databases, Network security, and compiler design.

Ashok K. Bhateja is currently working as Senior Scientist (Scientist 'G') in Defence Research & Development Organization (DRDO), Delhi. He completed his M.Sc. in Mathematics from University of Rajasthan, Jaipur and M.Tech. in Computer Science from Indian Institute of Technology, Delhi. His area of research is Artificial Intelligence, Biometrics, Cryptanalysis, Algorithm Design and Analysis and Parallel Computing System.