Breaking of Simplified Data Encryption Standard Using Binary Particle Swarm Optimization

Lavkush Sharma[1], Bhupendra Kumar Pathak[2] and Nidhi Sharma[3]

[1] Department of Computer Science & Engineering, Faculty of Engineering and Technology, Raja Balwant Singh College, Bichpuri, Agra, Uttar Pradesh--283105, India

[2] Department of Mathematics, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh--173234, India

[3] Department of Computer Application, GICTS Gwalior, Madhya Pradesh, India

ABSTRACT

Cryptanalysis of cipher text by using evolutionary algorithm has gained so much interest in last few years. This paper demonstrates the use of Binary Particle Swarm Optimization with bit change mutation operator for cryptanalysis of S-DES and then compared the results with Genetic Algorithm. An experimental result shows that Binary PSO performs better than the genetic algorithms for such type of problem. Here the cipher text attack is considered and several keys are generated in the iteration of the Binary Particle Swarm Optimization algorithm on the basis of their cost function value which depends upon letter frequency. The results on the S-DES indicate that, this is a promising method and can be adopted to handle other complex block ciphers like DES, AES.

Keywords: *Cryptanalysis, Ciphertext attack, Simplified Data Encryption Standard, genetic algorithm, Binary Particle Swarm Optimization.*

I. Introduction

A cipher is a secret way of writing in which plaintext is encrypted into ciphertext by using a key. Those who know the key can easily decrypt the ciphertext back into the plaintext. Cryptanalysis is the study of breaking ciphers, that is, finding the key or converting the ciphertext into the plaintext without knowing the key. Optimization techniques have got a significant importance in determining efficient solutions of different complex problems. One such problem is to break S-DES. This paper considers cryptanalysis of S-DES. In the brute force attack, the attacker tries each and every possible key on the part of cipher text until desired plaintext is obtained. A brute force approach may take so much time to guess the real key which is used to generate a cipher text. On the other hand optimization technique can be used for the same purpose. Genetic algorithm is an evolutionary algorithm that works well and takes less time to break cipher as compared to Brute force attack. BPSO is also a population based optimization technique which could be applied to solve such optimization problems. Unlike GA, PSO has no evolution operations like crossover and mutation. In the beginning the PSO can handle only continuous optimization problem but Kennedy and Eberhart [12] introduced a discrete binary version of PSO to handle discrete optimization problem. In Binary PSO, each particle represents its position in binary value which are 0 and 1. As an algorithm, the main strength of PSO is its fast convergence.

The remaining paper is organized as follows: Section 2 discusses the earlier works done in this field. Section 3 presents overview of S-DES and Section 4 discusses the Cost function and Section 5 gives the over view of Genetic Algorithm and BPSO. Experimental results are discussed in Section 6. Conclusion is presented in section 7.

2. Related work

In the past years, so many papers have been published in the field of cryptanalysis. R.Spillman etc. showed that Knapsack cipher [4] and substitution ciphers [5] could be attacked using genetic algorithm. In the recent years Garg[1,2] presented the use of memetic algorithm and genetic algorithm to break a simplified data encryption standard algorithm. Nalini [3] used efficient heuristics to attack S-DES. In 2006 Nalini used GA, Tabu search and Simulated Annealing techniques to break S-DES. Matusi [7] showed the first experimental cryptanalysis of DES using an linear cryptanalysis technique. Clark [6] also presented important analysis on how different optimization techniques can be used in the field of cryptanalysis. Vimalathithan [9,14] also used GA and PSO to attack Simplified-DES. Lavkush [15] also used Genetic Algorithm to break SDES. In 2011 Vimalathithan used Computational Intelligence for cryptanalysis of S-DES [13].

In this paper, a Binary PSO with bit change mutation [16] is used to break S-DES and then compare the results with Genetic Algorithm. A population of keys is generated and their fitness is calculated by using efficient fitness function. At the end, we will find the key in less time.

3. S-DES

In this section we will provide the overview of S-DES Algorithm. Simplified DES, developed by Professor Edward Schaefer of Santa Clara University is an educational rather than a secure encryption algorithm. The S-DES [8, 10] encryption algorithm takes an 8-bit block of plaintext and a 10-bit key as input and produces an 8-bit block of ciphertext as output. The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext. The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled f_K, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function f_K again; and finally a permutation function that is the inverse of the initial permutation (IP^{-1}).The function f_K takes as input not only the data passing through the encryption algorithm, but also an 8-bit key. S-DES uses a 10-bit key from which two 8-bit subkeys are generated. In this, the key is first subjected to a permutation (P10). Then a shift operation is performed. The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K_1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K_2).

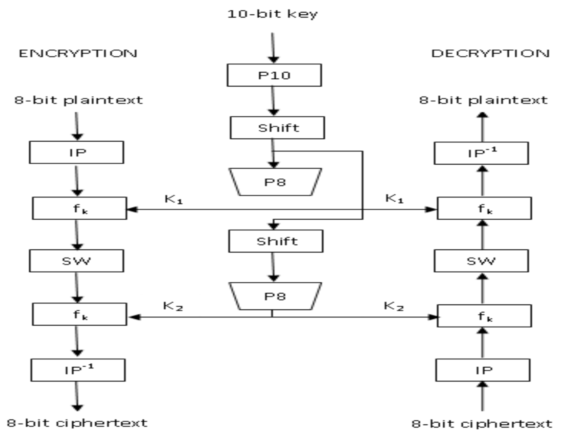

Figure 1: Simplified Data Encryption Algorithm

3.1 Initial and Final Permutations

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function IP= [2 6 3 1 4 8 5 7].This retains all 8-bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is applied; the inverse permutation is done by applying, IP^{-1} = [4 1 3 5 7 2 8 6] where we have $IP^{-1}(IP(X)) = X$.

3.2 The Function f_K

The function f_k, which is the complex component of S-DES, consists of a combination of permutation and substitution functions. The functions are given as follows.
Let L, R be the left 4-bits and right 4-bits of the input, then,

$$\mathbf{f_K (L, R) = (L \text{ XOR } f(R, key), R)}$$

Where XOR is the exclusive-OR operation and key is a sub -key. Computation of f(R, key) is done as follows.
1. Apply expansion/permutation E/P= [4 1 2 3 2 3 4 1] to input 4-bits.
2. Add the 8-bit key (XOR).
3. Pass the left 4-bits through S-Box S0 and the right 4-bits through S-Box S1.
4. Apply permutation P4 = [2 4 3 1].

The S-boxes operate as follows:

The first and fourth input bits are treated as 2-bit numbers that specify a row of the S-box and the second and third input bits specify a column of the S-box.

The entry in that row and column in base 2 is the 2-bit output.

$$S0 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 3 & 2 \\ 1 & 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 & 3 \\ 3 & 3 & 1 & 3 & 2 \end{array}$$

$$S1 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 3 \end{array}$$

Figure 2: Working of S-box

3.3 The Switch Function

The function f_K only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that the second instance of f_K operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K2.

4. Cost function

Equation (1) is a general fitness function used to determine the suitability of a assumed key (k). Here, A denotes the language alphabet (i.e., for English, [A... Z, _], where _ represents the space symbol), K and D denote known language statistics and decrypted message statistics, respectively, and the u, b, and t denote the unigram, digram and trigram statistics respectively; α, β and γ are the weights assigning different weights to each of the three statistics where α+ β + γ = 1. In view of the computational complexity of trigram, only unigram and digram statistics are used.

$$\mathbf{C}^{\mathbf{K}} = \alpha \ \Sigma(i \ \varepsilon \ \tilde{A}) \ |K(i)^u - D(i)^u| + \beta \ \Sigma(i, j \ \varepsilon \ \tilde{A}) \ | K(i, j)^b - D(i, j)^b| + \gamma\Sigma(i,j,k\varepsilon\tilde{A})|K(i,j,k)^t - D(i,j,k)^t| \qquad (1)$$

5. Methodology

5.1Genetic Algorithm

The genetic algorithm [21, 23] is a search algorithm based on the natural selection and "survival of the fittest", the main idea is that in order for a population of individuals to adapt to some environment, it should behave like a natural system. This means that survival and reproduction of an individual is promoted by the elimination of useless traits and by rewarding useful behavior. The genetic algorithm belongs to the family of evolutionary algorithms. An evolutionary algorithm maintains a population of solutions for the problem at hand. The population is then evolved by the iterative application of a set of stochastic operators. The simplest form of genetic algorithm involves three types of operators: selection, crossover and mutation.

A selection operator is applied first.

Selection: This selection operator selects chromosomes in the population for reproduction. The better the chromosome, the more times it is likely to be selected to reproduce.

Crossover: Crossover selects genes from parent chromosomes and creates a new offspring. The Simplest way to do this is to choose randomly some crossover point and everything before this point is copied from the first parent and then, everything after a crossover point copied from the second parent.

Mutation: After a crossover, mutation is performed. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. In binary GA we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1.

In genetic algorithm, we used Ring crossover operator [11].In ring crossover two parents such as parent1 and parent2 are considered for the crossover process, and then combined in the form of ring, as shown in fig. 3(b). Later, a random cutting point is decided in any point of ring. The children are created with a random number generated in any point of ring according to the length of the combined two parental chromosomes. With reference to the cutting point, while one of the children is created in the clockwise direction, the other one is created in direction of the anti-clockwise, as shown in fig. 3(c).Then swapping and reversing process is performed in the Ring Crossover operator, as shown in fig. 3(d).

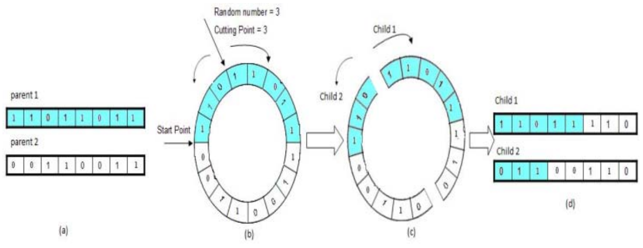

Figure 3: Ring Crossover Procedure

The procedure to carry out the cryptanalysis using GA in order to break the key is as follows

1. Input: ciphertext, and the language statistics.
2. Randomly generate an initial pool of solutions.

3. Calculate the fitness value of each of the solutions in the pool using equation (1).
4. Create a new population by repeating following steps until the new population is complete
   a. Select parent (keys) from a current population according to their fitness value (the better fitness, the bigger chance to be selected). Here Tournament selection is used.
   b. With a crossover probability cross over the parents to form new offspring (children). In our genetic algorithm we are using Ring Crossover Operator
   c. For each of the children, perform a mutation operation with some mutation probability to generate new offspring.
   d. Place new offspring in the new population
5. Use new generated population for a further run of the algorithm
6. If the end condition is satisfied, stop, and return the best solution in current population

5.2 Binary Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based optimization algorithm developed by Kennedy and Eberhart. It was inspired by the social behavior of group of birds when searching for food. In PSO algorithm, each solution is called particle, and each particle flies around in the hyper-dimensional search space with a velocity, which is updated constantly according to the particle's own value (solution) and the value of the particle's neighbors. Each particle keeps track of its best value (solution) it has achieved so far, this value is called P_{best}. Moreover, each particle knows the best value so far achieved in the group is known as G_{best}. The next move of each particle is controlled by a velocity vector and this is influenced by both P_{best} and G_{best}. In the search space, the updating of each particle's velocity $V_n(t)$ and position $X_n(t)$ is based on the following equation.

$$V_n(t+1) = w*V_n(t) + R_1C_1*(P_{best\,,\,n} - X_n(t)) + R2C2*(G_{best\,,n} - X_n(t)) \quad (2)$$

$$X_n(t+1) = X_n(t) + V_n(t+1) \quad (3)$$

C1 and C2 are positive acceleration constants,R1 and R2 are random numbers between 0 and 1,and w is the inertia weight.

A Binary version of PSO was introduced by Kennedy and Eberhart [12]. But Binary version of Particle swarm optimization has trouble jumping out of good local optima. This problem can be handled by Binary PSO with Bit change mutation [16].

In the binary PSO, the particle's personal best and global best is updated as in continuous PSO. The major difference between binary PSO with continuous PSO is that velocity of the particles uses the probability that a bit takes on 0 or 1. Using this definition a velocity must be restricted within the range [0, 1]. In the BPSO the equation (2) of updating a velocity remains unchanged, but the equation (3) for updating a position is re-defined by the following equation (4).

If (rand () < S (v_n(t+1)) then x_n (t+1) = 1
Else x_n (t+1)=0 (4)

Where S (.) is a sigmoid function and this function is used for transforming the velocity to the probability constrained to the interval [0, 1] and rand () is a random number selected from the interval [0, 1].

When Binary Particle Swarm Optimization starts the iteration to find an optimum solution, the velocity tend to go into v_{max} or $-v_{max}$ by the velocity update equation (2) according as the corresponding target position is one or zero, respectively. If a velocity converges near v_{max} or $-v_{min}$, it is very difficult to change the corresponding position with a small variation of velocity, which makes it hard to escape out from a good local optimum in BPSO. In order to handle this undesired position, large movement of velocity is required, which is not related to the p_{best} and g_{best}. To accomplish the above objective, the following operation is inserted between velocity update and position update in BPSO process

If rand () < r_{mute}) then v_n (t+1) = $-v_n$ (t+1) (5)

Where r_{mute} is a probability. If this operation is executed when velocity is near v_{max} and $-v_{max}$, then the position will be changed from one to zero or zero to one, respectively.

The Algorithm to carry out the cryptanalysis using BPSO in order to break the key is as follows.

Algorithm:

Step1. Initialize the particles randomly to form swarm and parameters of PSO.
Step2. Calculate the fitness values of each particles according to the fitness function given in equation (1).The fitness function decide whether the solution is good or not.
Step3. Update the velocity and particles position according to the equations (2), (4) and equation (5).
Step4. Update the local optima and the global optimum.
Step4.If the Max no of iteration has exceeded then stop the iteration or if the key with low fitness value is found, then go to step 5 else go to the step 2.
Step5.Display the best key is found so far and then exit

6. Results and Discussion

Our objective in this paper is to compare the results obtained from Binary Particle swarm optimization algorithm with bit change mutation with the genetic algorithms. The experiments were conducted on Core 2 Duo system. There are a variety of cost functions used by other researchers in the past. The most common cost function uses gram statistics. Some use a large amount of grams while others only use a few. Equation (1) is a general formula used to determine a proposed key. A number of experiments have been carried out by giving different inputs and applying Genetic Algorithm and Binary PSO for breaking Simplified Data Encryption Standard. The results are shown in table 1.The table below shows that the key bits matched using Genetic Algorithm and Binary Particle Swarm optimization algorithm for the given cipher text .the choice of the Genetic parameters and BPSO parameters are described below:

GA Parameters
The following are the GA parameters used.
Population Size: 100
Selection: Tournament Selection operator
Crossover: Ring Crossover
Crossover: .85
Mutation: .02
No. of Generation: 50

BPSO parameters

Self Recognition Parameter C1	2
Social Parameter C2	2
Inertia Weight	$0.99 < w < 0$
Initial Population	100
No of Iteration	50
r_{mute}	.004

Table 1: Comparison of Genetic Algorithm and Binary Particle Swarm optimization Algorithm.

S. No	Amount of Cipher Text	No. of bits matched using GA	No. of bits matched using Binary PSO	Time Taken by GA (M)	Time Taken by Binary PSO (M)
1.	200	5	7	4.7	3.9
2.	400	4	7	2.1	2.1
3.	600	7	8	1.9	1.7
4.	800	8	9	3.1	2.8
5.	1000	9	10	2.6	2.3
6.	1200	9	10	2.1	1.9

Initially the random keys are generated and the known Ciphertext are decrypted using these random keys .the cost value is computed by the fitness function using equation (1) .For the both algorithm GA and BPSO the parameters chosen shown above and in my view this is the best configuration found. From the above table, it is found that Binary Particle swarm optimization algorithm works better than Genetic Algorithm in terms of time taken as well as obtaining number of key bits. This is because we used a bit change mutation operator with BPSO and then search spaces are searched optimally. Also we can say that including mutation operator with Binary PSO can help the BPSO to improve its performance.

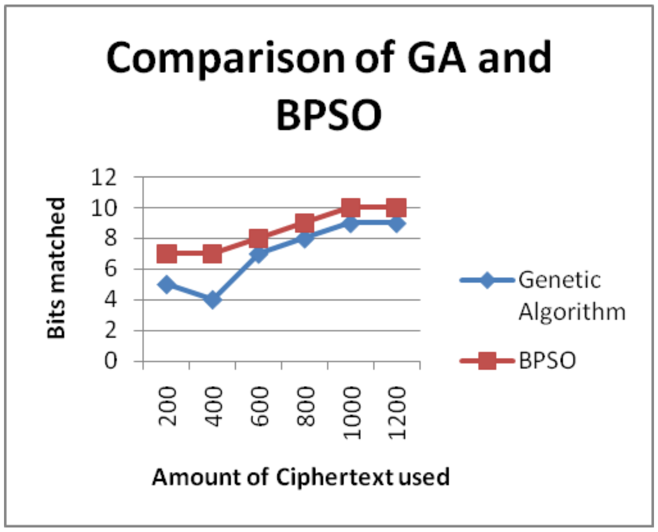

Figure 4: comparison of Genetic algorithm and Binary particle swarm optimization algorithm

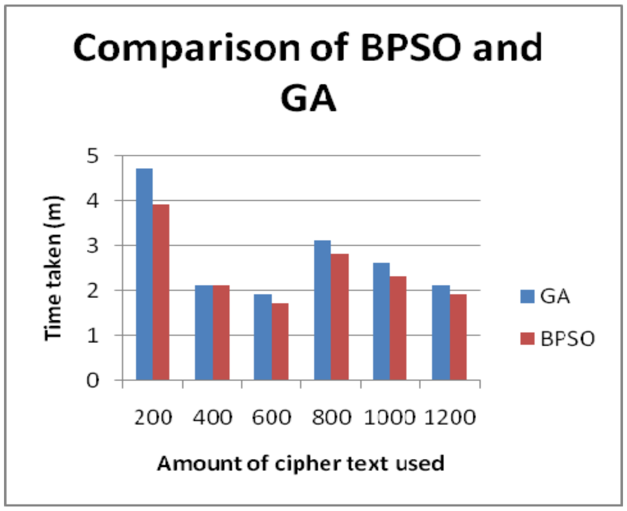

Figure 5: The running time comparison of Genetic Algorithm and BPSO Algorithm

7. Conclusion

In this paper, we have used Binary PSO with bit change mutation and Genetic algorithm with Ring crossover for the cryptanalysis of Simplified Data Encryption Standard. We found that BPSO is better than genetic algorithm for cryptanalysis of S-DES. Although S-DES is a simple encryption algorithm, BPSO with bit change mutation operator method can be adopted to handle other complex block ciphers like DES and AES.

References

[1] G Poonam, Memetic Algorithm Attack on Simplified Data Encryption Standard algorithm, proceeding of International Conference on Data Management, February 2008, pg 1097-1108

[2] Garg Poonam, Genetic algorithm Attack on Simplified Data Encryption Standard Algorithm, International journal Research in Computing Science,ISSN1870-4069, 2006.

[3] Nalini, Cryptanalysis of S-DES via Optimization heuristics, International Journal of Computer Sciences and network security, vol 6, No 1B, Jan 2006.

[4] Spillman, R.: Cryptanalysis of Knapsack Ciphers Using Genetic Algorithms.Cryptologia XVII(4), 367–377 (1993)

[5] Spillman, R., Janssen, M., Nelson, B., Kepner, M.: Use of A Genetic Algorithm in the Cryptanalysis of simple substitution Ciphers. Cryptologia XVII(1), 187–201 (1993)

[6] Clark A and Dawson Ed, "Optimisation Heuristics for the Automated Cryptanalysis of Classical Ciphers", Journal of Combinatorial Mathematics and Combinatorial Computing, Vol.28,pp. 63-86, 1998.

[7] M. Matsui, Linear cryptanalysis method for DES cipher, Lect. Notes Comput. Sci. 765 (1994) 386–397.

[8]William Stallings, Cryptography and Network Security Principles and Practices, Third Edition,Pearson Education Inc.,2003.

[9]Vimalathithan.R,M.L.Valarmathi,"Cryptanalysis of SDES Using Genetic Algorithm", International Journal of Recent Trends in Engineering, Vol2, No.4, November 2009, pp.76-79.

[10]Schaefer E, "A Simplified Data Encryption Standard Algorithm", Cryptologia, Vol .20, No.1, pp. 77-84, 1996.;

[11]Yılmaz Kaya, Murat Uyar, Ramazan Tekdn," A Novel Crossover Operator for Genetic Algorithms: Ring Crossover".

[12]J,kennedy, R.Eberhart, "A discrete Binary version of the Particle Swarm Algorithm," International Conference on Neural network, Vol. IV,pp:4104-4108, Australia,1997.

[13]Vimalathithan.R,M.L.Valarmathi,"Cryptanalysis of Simplified-DES using Computational Intelligence ,WSEAS transactions on computers Issue 7, Volume 10, July 2011.

[14]Vimalathithan.R,M.L.Valarmathi,"Cryptanalysis of SDES Using Particle Swarm Optimization", 10th National Workshop on Cryptology, Coimbatore, India, Sep 2010.

[15]Lavkush Sharma, "Breaking of SDES using GA" GJCST,Volume 12 Issue 5 Version 1.0 March 2012

[16] Sangwook LEE " Binary Particle Swarm Optimization with Bit Change Mutation", IEICE Transactions on Fundamentals of Electronics, Communications and C omputer Sciences,Volume E90-A Issue 10, October 2007

[17]C.Papagianni,"Communication n/w design using PSO"computer science and Information technology, 2008,ISSN 1896-7094

[18] Xiao-Feng Xie ,"A Dissipative Particle swarm optimization" CEC,USA,2002:1456-1261

[19] Davis,L. "Handbook of Genetic Algorithm",Van Nostrand Reinhold, New York,1991

[20] D. E. Goldberg,"Genetic algorithms in search. Optimization and Machine Learning.Reading. M.A. addison -Wesley.1989.

[21] A,Michalewiez and N. Attia." Evolutionary optimization of constrained problems." InProc.3rd annu. Conf. on Evolutionary Programming. 1994.pp 98-108

[22] R. Toemeh, S. Arumugam, Breaking Transposition Cipher with Genetic Algorithhm Electronics and Electrical Engineering,ISSN 1392 – 1215 2007. No. 7(79)

[23]Kalyanmoy Deb, Multi-objective Optimization using Evolutionary Algorithms, John Wiley and Sons, 2001.

Lavkush Sharma: I completed my B.Tech in 2004 and M.Tech (Computer Science & Engineering) in 2009 from Uttar Pradesh Technical University, Lucknow, India. I am working as a Assistant Professor in Computer Science & Engineering Department in the Faculty of Engineering, RBS College, Bichpuri Agra. since 2005. My research area is Cryptography and network security. I have published more than 6 papers at various international and national conferences and journals,

Dr. Bhupendra Kr Pathak: He has completed his PGDCSA and Ph.D. from Dayalbagh Deemed University, Agra. He worked as a Lecturer at R.B.S. Degree College, Agra from 2003 to Aug. 2007. From 2008 to 2009 worked as a Lecturer at AMITY University, Noida. From 2009 to 2011 worked as a Assistant Professor at ITM University, Gurgaon. From 2011 to present working as Senior Lecturer at JAYPEE University of Information Technology, Waknaghat, Solan. He has published more than 10 papers at various international and national conferences and journals, His research interest are: Soft Computing Techniques (Genetic Algorithms, Neural Network, Fuzzy Set Theory), Optimization Techniques.

Nidhi Sharma: is currently working as a Assistant Professor in the Department of Computer Application at GICTS, Gwalior, India. She Obtained MCA from Uttar Pradesh Technical University in 2006. Her areas of research are Cryptography, Information security, optimization.