COMPUTATIONAL INTELLIGENCE BASED SIMULATED ANNEALING GUIDED KEY GENERATION IN WIRELESS COMMUNICATION (CISAKG)

ARINDAM SARKAR[1] AND J. K. MANDAL[2]

[1]Department of Computer Science & Engineering, University of Kalyani, W.B, India
[2]Department of Computer Science & Engineering, University of Kalyani, W.B, India

ABSTRACT

In this paper, a Computational Intelligence based Simulated Annealing (SA) guided approach is use to construct the key stream. SA is a randomization technique for solving optimization problems. It is a procedure for finding good quality solutions to a large diversity of combinatorial optimization problems. This technique can assist to stay away from the problem of getting stuck in local optima and to escort towards the globally optimum solution. It is inspired by the annealing procedure in metallurgy. At high temperatures, the molecules of liquid move freely with respect to one another. If the liquid is cooled slowly, thermal mobility is lost. Parametric tests are done and results are compared with some existing classical techniques, which shows comparable results for the proposed system.

KEYWORDS

Simulated Annealing (SA), Key Genaeration, Computational Intelligence..

1. INTRODUCTION

In recent times wide ranges of techniques are developed to protect data and information from eavesdroppers [1]-[4]. These algorithms have their merits and shortcomings. For Example in DES, AES [4] algorithms the cipher block length is nonflexible. ANNRPMS [1] and ANNRBLC [2] allow only one cipher block encoding. In NNSKECC algorithm [3] any intermediate blocks throughout its cycle taken as the encrypted block and this number of iterations acts as secret key. In this paper we have proposed a SA based encryption technique for wireless communication.

The organization of this paper is as follows. Section 2 of the paper deals with the proposed SA based key generation technique. Example of the key generation and encryption technique has been discussed in section 3. Results are described in section 4. Conclusions are drawn in section 5 and that of references at end.

2. THE SA BASED KEY GENERATION TECHNIQUE

In SA, the solution starts with a high temperature, and a sequence of trail vectors are generated until inner thermal equilibrium is reached. Once the thermal equilibrium is reached at a particular temperature, the temperature is reduced and a new sequence of moves will start. This process is continued until a sufficiently low temperature is reached, at which no further improvement in the objective function can be achieved. Thus, SA algorithm consists of: configurations, reconfiguration technique, cost function, and cooling schedule.

The key stream generators considered here are LFSR-based generators. In Key generation process some of the operators are being used in a particular format. The following table illustrated the format of the operator and their corresponding meaning.

Table 1. Operator's format and their meaning

Operator	**Format**	**Meaning**
|	|ab	Bitwise OR
&	&ab	Bitwise AND
^	^ab	Bitwise XOR
X		Character sequence from 'a'… p' represents the number 0..15
SR	SRx	Shift Register is represents as SR and x denotes the feedback polynomial

2.1 Chromosome Representation Scheme

The population chromosome that represents candidate key stream generators is strings of characters which are expressions represented using prefix notation. These syntactic rules should be preserved during the generation of the initial population, and by the genetic operations. The initial states and feedback functions of the shift registers are represented as strings of the letters 'a'..'p'. These letters represent the numbers 0..15. Thus, each letter is a sequence of four bits. The length of a LFSR is determined by the number of letters which are initially generated randomly. The number of these letters must be even, half of them for the initial state, and the second half for the feedback function. For example, if the number of these letters is eight letters, then four letters are used for the feedback function, thus, the length of LFSR is 16 bits (4 × 4). Furthermore, the first zeros of the feedback function are ignored. For example, consider the LFSR:"SR abid", 'i' is the number $8 = (1000)_2$, then the first three zeros are ignored, and the length of this LFSR will be five bits (1 + 4). Thus the feedback function will be (11100), or $g(x) = 1+x + x^2 + x^5$.

The following are examples of the chromosomes:
Chromosome: SRggbkbecdeh
Chromosome: □□&|SRbpeiSRhoionm□SRlhhk&SRfmcddiphhcSRcgpjkgSRiechSRkhji
Chromosome: □SRdcaeSRagojdfojfm
Chromosome: |&SRccga□SReehk&|SRpfdmingc□SRjeSRjmlidmbeSRhoSRmhofoh
Chromosome: SRlepjgc

2.2 Construction of Fitness Function

The fitness value is a measurement of the goodness of the key stream generator, and it is used to control the application of the operations that modify a population. There are a number of metrics used to analyze key stream generators, which are key stream randomness, linear complexity and correlation immunity. Therefore, these metrics should be taken in account in designing key stream generators, and they are in general hard to be achieved. The fitness value is calculated by generating the key stream after executing the program, and then the generated key stream is examined. The fitness function used to evaluate the chromosomes is to calculate at what percentage the chromosome satisfies the desired properties of the stream ciphers. Three factors are considered in the fitness evaluation of the chromosomes which are:

1. Randomness of the generated key stream.
2. Key stream period length.
3. Chromosome length.

Following equation is used for the evaluation of key stream randomness using the frequency and serial tests, in which, *nw* is the frequency of *w* in the generated binary sequence.
This function is derived from the fact that in the random sequence:

1. Probability (n_o) = Probability (n_1), and
2. Probability (n_{01}) = Probability (n_{11}) = Probability (n_{10}) = Probability (n_{00})

$$f_1 = |n_0 - n_1| + \left|n_{00} - \frac{SZ}{4}\right| + \left|n_{01} - \frac{SZ}{4}\right| + \left|n_{10} - \frac{SZ}{4}\right| + \left|n_{11} - \frac{SZ}{4}\right|$$

There is another randomness requirement which is: $\frac{1}{2^i} \times n_r$ of the runs in the sequence are of length *i*, where n_r is the number of runs in the sequence. Thus, we have the following function:

$$f_2 = \sum_{i=1}^{M} \left| \left(\frac{1}{2^i} \times n_r \right) - n_i \right|$$

where *M* is maximum run length, and η_i is the desired number of runs of length *i*. Another factor is considered in the evaluation of the fitness value which is the size of the candidate key stream generator (length of the chromosome). Thus, the fitness function used to evaluate the chromosome x will be as follows, where *wt* is a constant and *size* is the key stream period length:

$$fitness\ (x) = \frac{SZ}{1 + f_1 + f_2} + \frac{weight}{length\ (x)}$$

2.3 Parameters value of the Algorithm

The parameters used in this work were set based on the experimental results, the parameter value that show the highest performance was chosen to be used in the implementation of the algorithm. Thus, the genetic operations used to update the population are single point crossover with probability pc=1.0 and mutation with probability pm=0.1. The selection strategy, used to select chromosomes for the genetic operations, is the 2- tournament selection. The old population is completely replaced by the new population which is generated from the old population by applying the genetic operations. Regarding the structure of each chromosome, the maximum chromosome length is 300 characters, and the maximum number of functions (except SR) is ten functions. The probability of the function SR is 0.5, and all other function are of probability 0.5. Finally, the maximum LFSR length is 20 bits. The run of GP is stopped after a fixed number of generations. The solution is the best chromosome of the last generation.

Algorithm Simulated Annealing based Key Stream Generation

1: Input : Length of the key stream
2: Output : Simulated Annealing based key stream
Method:
3: Generate the initial population (pop) randomly
4: Evaluate pop
5: temp ← 250.
*6: **while** not Max Number of generations **do***
7: Generate a new population (pop1) by applying crossover and mutation

Evaluate the fitness of the new generated chromosomes of pop1
Calculate the averages of fitness values for pop and pop1, av and av1 respectively
If (av1 > av) then replace the old population by the new one, i.e. pop ← pop1
Else
Begin
e ← av − av1
Pr ← e/Temp
Generate a random number (rnd)
If (exp(−pr) > rnd) then pop ← pop1
End Else
EndIf
Temp ← Temp □ 0.95
end while
Return the best chromosome of the last generation

3. EXAMPLE OF KEY STREAM GENERATION AND SA BASED ENCRYPTION

Consider Initial population size as 200 and randomly generated each key stream having 128 bits. Then population gets evaluated with the help of fitness function by passes through a number of statistical tests to examine whether the pseudorandom number sequences are sufficiently random or not, which are frequency test, serial test, poker test, auto correlation test and runs test.

1. *Frequency Test:* It calculates the number of ones and zeroes of the binary sequence and checks if there is no large difference.
2. *Serial Test:* The transition characteristics of a sequence such as the number 00, 01, 10 and 11 are evaluated. Ideally, it should be uniformly distributed within the sequence.
3. *Poker Test:* A N length sequence is segmented into blocks of M bits and the total number of segments is N/M. Within each segment, the integer value can vary from 0 to m = 2M-1. The objective of this test is to count the frequency of occurrence of each M length segment. Ideally, all the frequency of occurrences should be equal
4. *Runs Test:* A sequence is divided into contiguous stream of 1's that is referred as blocks and contiguous stream of 0's that is referred as gaps. If r_{i0} is the number of gaps of length i, then half of the gaps will have length 1 bit, a quarter with length 2 bits, and an eighth with length 3 bits. If ri1 is the number of blocks of length i, then the distribution of blocks is similar to the number of gaps.

After the maximum generation this proposed SA based key generation algorithm will generate best fittest key stream having length of 128 bits.
10110011101000000110101001011101100010000100100001111010100000101000011001000101000011100111100010000001100111100011111000000110101
Now, consider the plain text to be encrypted is **"SA Encryption"**
01000001/01010011/00100000/01000101/01101110/01100011/01110010/01111001/01110000/01110100/01101001/ 01101111/01101110
Here "/" is used as the separator between successive bytes.
In this example plain text size is 104 bits. Here plain text size is less than the size of the 128 bit SA based key stream. So, no need to perform key expansion operation.
Perform XOR operation between plain text and SA based key stream.
So, after the XOR operation cipher text is
11110010111100111110100111111100111111011110011100001110111110001111100111111100111010110011110011010000 i.e. **"òóôþ~ó‡||þužh"**

4. RESULTS

Table 2 depicts the average fitness values of different number of generations. Table shows 4 set of entries where 40, 60 80, 100 number of generations are considered. It is observed from the table that increasing the number of generation also increased the fitness values in average.

Table 2 Average of fitness values

Number of Generations	Average of fitness values
40	35.1486
60	35.8713
80	36.2581
100	36.7316

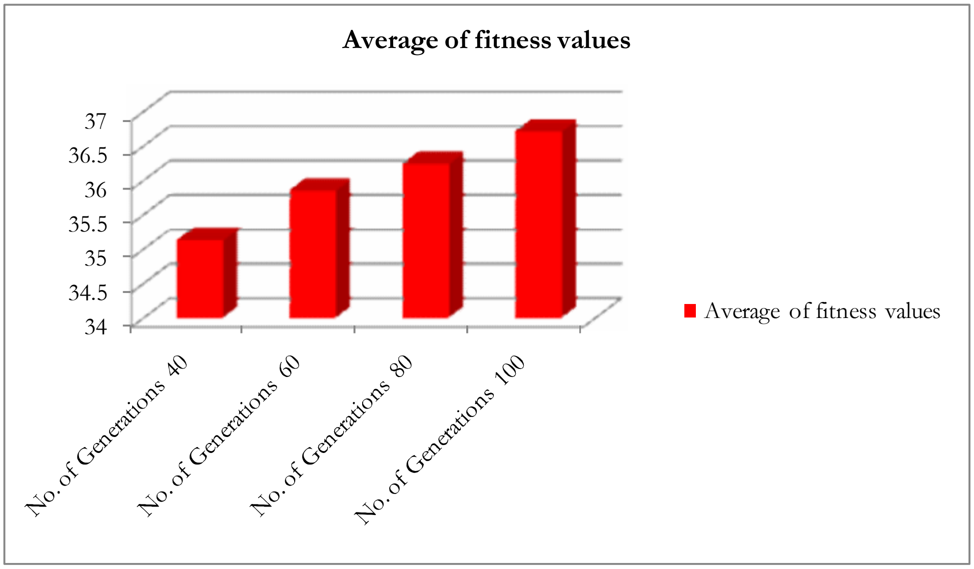

Figure 1 Number of Generation Vs Average of Fitness Values

Table 3 tabulated the fitness values of 50 numbers of iterations and the average fitness value of 50 iterations is 34.89712

Table 3 List of fitness values in 50 iterations

Iteration No.	**Fitness Value**
1	31.4377
2	37.5723
3	34.4687
4	36.3263
5	31.8379
6	39.5962
7	41.6294
8	32.6138
9	28.5972
10	32.2379
11	36.7457
12	35.8027
13	28.0429
14	34.7493
15	25.1223
16	43.6024
17	36.9032
18	38.7839
19	27.9125
20	41.5432
21	30.9120
22	35.2396
23	34.9486
24	29.6919
25	27.1037
26	39.6495
27	36.9377
28	38.3426
29	36.6485

Iteration No.	**Fitness Value**
30	32.4891
31	24.8239
32	31.3793
33	36.1682
34	37.8425
35	33.7348
36	42.3876
37	30.9190
38	28.6409
39	40.1002
40	37.6817
41	36.8629
42	38.6328
43	36.1684
44	38.8292
45	32.9716
46	35.4094
47	29.6962
48	42.7356
49	34.8038
50	37.5792

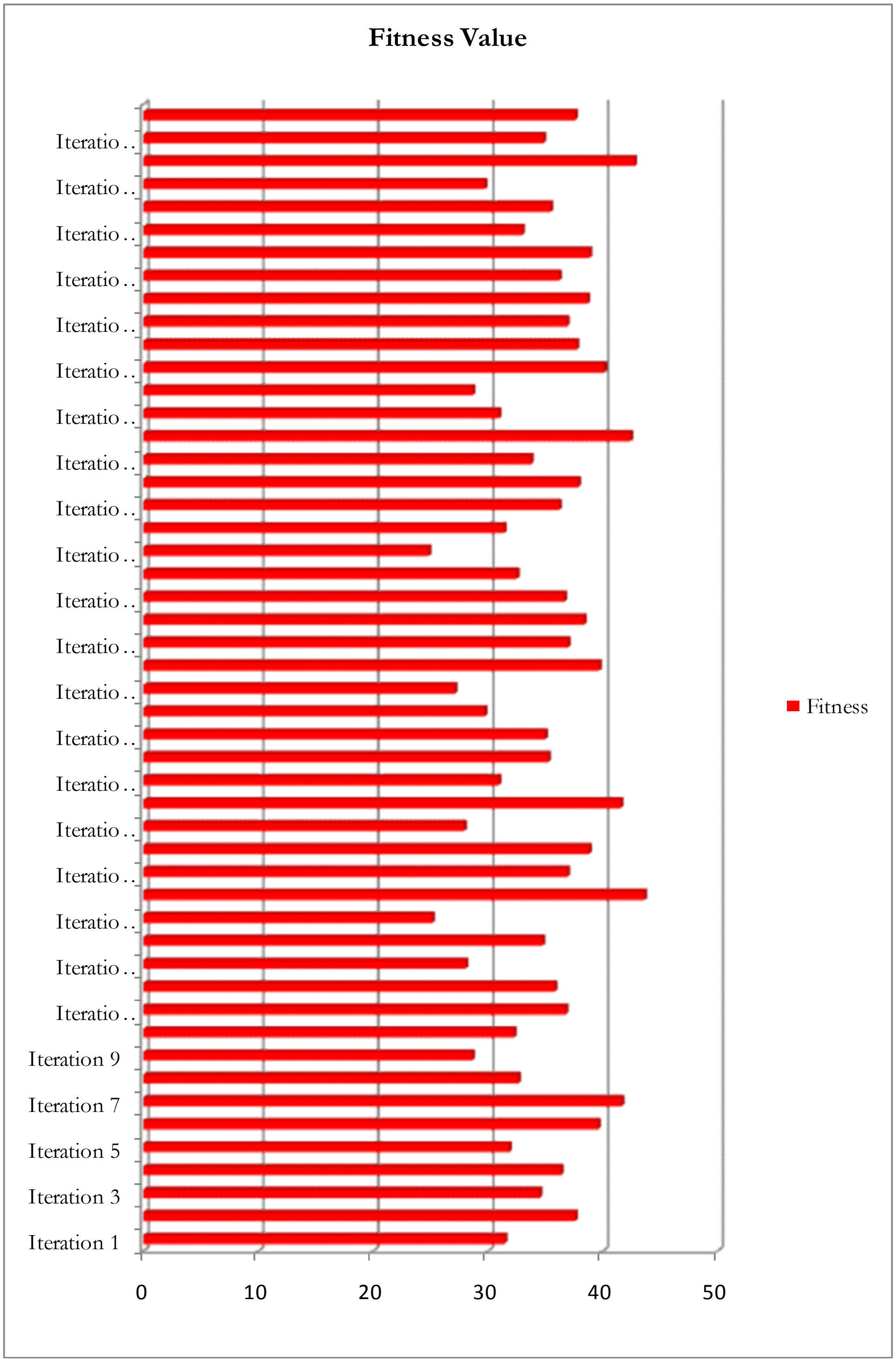

Figure 2 Graph Representation of No. of Iteration Vs Fitness Values

After the maximum generation this proposed SA based key generation algorithm will generate best fittest key stream. First 128 bits from the fittest key stream considered as a SA based key.

Key Storage Comparison and Analysis with Existing Methods

Table 4 shows the comparison of results among Proposed CISAKG, AES, RC4 and Vernam Cipher.

Table 4 Comparison of key storage in Proposed CISAKG, AES, RC4 and Vernam Cipher

Length of Plain text	Key Storage Proposed (CISAKG)	Key Storage (AES)	Key Storage (RC4)	Key Storage (Vernam Cipher)
64	128	128	52	60
120	128	128	106	120
500	128	128	437	500
1000	128	128	913	1000

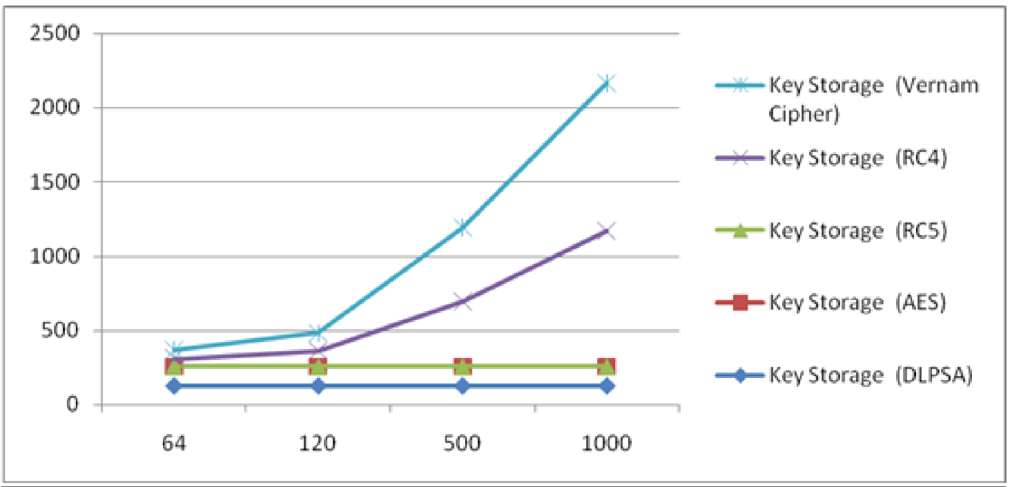

Figure 3 Comparison of key storage in Proposed CISAKG, AES, RC5, RC4 and Vernam Cipher

5. CONCLUSION

In CISAKG the number of keys to be stored is at par AES and less when compared to RC4, Vernam Cipher and the keys are generated by passes through a number of statistical tests to examine randomness of the generated key stream, key stream period length, chromosome length using some statistical test like frequency test, serial test, poker test, auto correlation test and runs test. This procedure ensures the robustness of the key. In CISAKG key stream size is 128. If number of bits in a plain text is grater than the key stream then key stream get expanded and if the plain text size is less than 128 bits than the size of the key stream used for encryption is 128. In AES encryption strategy the minimum key stream requirement is 128 bits. Whereas RC4 stream cipher method is vulnerable to analytic attacks of the state table. 1 out of every 256 keys is a weak key. These keys can be identified by cryptanalysis which can find whether the generated bytes are strongly correlated with the bytes of the key. SA based key generation is a procedure for finding good quality solutions to a large diversity of combinatorial optimization problems. and also helps to avoid from the problem of getting stuck in a local optima In Vernam cipher the keys are randomly generated using random stream generator. The drawback is that the number of keys to be stored and distributed should be equal to the length of the plain text. Also the keys used to encrypt the plain text can be found if the random number generator is cracked.

ACKNOWLEDGEMENTS

The author expresses deep sense of gratitude to the DST, Govt. of India, for financial assistance through INSPIRE Fellowship leading for a PhD work under which this work has been carried out.

REFERENCES

[1] Mandal, J. K., Sarkar Arindam, "An Adaptive Neural Network Guided Secret Key Based Encryption Through Recursive Positional Modulo-2 Substitution For Online Wireless Communication (ANNRPMS)", *in Proc. International Conference on Recent Trends In Information Technology (ICRTIT 2011) Conf. BY IEEE*, 3-5 June 2011, Madras Institute of Technology, Anna University, Chennai, Tamil Nadu, India. 978-1-4577-0590-8/11

[2] Mandal, J. K., Sarkar Arindam, "An Adaptive Neural Network Guided Random Block Length Based Cryptosystem (ANNRBLC)", *in Proc. 2nd International Conference On Wireless Communications, Vehicular Technology, Information Theory And Aerospace & Electronic System Technology" (Wireless Vitae 2011) Conf By IEEE Societies*, February 28, 2011- March 03, 2011,Chennai, Tamil Nadu, India. ISBN 978-87-92329-61-5.

[3] Mandal, J. K., Sarkar Arindam "Neural Network Guided Secret Key based Encryption through Cascading Chaining of Recursive Positional Substitution of Prime Non-Prime (NNSKECC)" [TP-48][PID-63], *in Proc. of International Confrence of Computing and Systems-2010 Conf by ICCS-2010,* Novembar 19-20, 2010, The University of Burdwan,pp 291-297.

[4] Atul Kahate, *Cryptography and Network Security*, 2003, Tata McGraw-Hill publishing Company Limited, Eighth reprint 2006.

Arindam Sarkar

INSPIRE FELLOW (DST, Govt. of India), MCA (VISVA BHARATI, Santiniketan, University First Class First Rank Holder), M.Tech (CSE, K.U, University First Class First Rank Holder). Total number of publications 25.

Jyotsna Kumar Mandal

M. Tech.(Computer Science, University of Calcutta), Ph.D.(Engg., Jadavpur University) in the field of Data Compression and Error Correction Techniques, Professor in Computer Science and Engineering, University of Kalyani, India. Life Member of Computer Society of India since 1992 and life member of cryptology Research Society of India. Dean Faculty of Engineering, Technology & Management, working in the field of Network Security, Steganography, Remote Sensing & GIS Application, Image Processing. 25 years of teaching and research experiences. Eight Scholars awarded Ph.D. and 8 are pursuing. Total number of publications 267.