

GENETIC ALGORITHM GUIDED KEY GENERATION IN WIRELESS COMMUNICATION (GAKG)

Arindam Sarkar¹ and J. K. Mandal²

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

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

ABSTRACT

In this paper, the proposed technique use high speed stream cipher approach because this approach is useful where less memory and maximum speed is required for encryption process. In this proposed approach Self Acclimatize Genetic Algorithm based approach is exploits to generate the key stream for encrypt / decrypt the plaintext with the help of key stream. A widely practiced approach to identify a good set of parameters for a problem is through experimentation. For these reasons, proposed enhanced Self Acclimatize Genetic Algorithm (GAKG) offering the most appropriate exploration and exploitation behavior. Parametric tests are done and results are compared with some existing classical techniques, which shows comparable results for the proposed system.

KEYWORDS

Genetic Algorithm (GA), Key Generation, 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 GA 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 GA BASED KEY GENERATION TECHNIQUE

Ordinary version of GA suffers from many troubles such as getting stuck in a local minimum and parameters dependence. In this proposed Self Acclimatize Genetic Algorithm approach some useful improvements have been proposed to enhance the performance of the simple GA, by dynamically adjusts selected control parameters, such as population size and genetic operation rates, during the course of evolving a problem solution. That is because, one of the main problems

related to GA is to find the optimal control parameter values that it uses, when a poor parameter setting is made for an evolutionary computation algorithm, the performance of the algorithm will be seriously degraded. Thus, different values may be necessary during the course of a run. The key stream generators considered a function library used in this work.

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^SRlhkk&SRfmccddiphhcSRcgpjkgSRiechSRkhji

Chromosome: ^SRdcaeSRagojdfjfm

Chromosome: |&SRccga^SReehk&|SRpfdmingc^SRjeSRjmlidmbeSRhoSRmhfoh

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 our 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 keystream.
2. Keystream period length.
3. Chromosome length.

Following equation is used for the evaluation of keystream randomness using the frequency and serial tests, in which, n_w 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_0) = Probability (n_1), and
2. Probability (n_{01}) = Probability (n_{11}) = Probability (n_{10}) = Probability (n_{00})

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

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 n_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 Algorithm Parameters

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 of crossover=1.0 and mutation with probability of mutation=0.1. The selection strategy, used to select chromosomes for the genetic operations, is the binary 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. The run of SAGA is stopped after a fixed number of generations. The solution is the best chromosome of the last generation.

Algorithm: Self Acclimatize Genetic Algorithm (SAGA)

- 1: Input : Length of key
 - 2: Output : SAGA based key stream
 - 3: Generate the initial population (pop) randomly
 - 4: Evaluate pop
 - 5: while not Max Number of generations do
 - 6: Generate a new population (pop1) by applying crossover and mutation and Self Acclimatizing adjustment of the population size, crossover and mutation probabilities
 - 7: Evaluate the fitness of the new generated chromosomes of pop1
 - 8: Replace the old population by the new one, i.e., pop ← pop1
 - 9: end while
 - 10: Return the best chromosome of the last generation
-

The goals of Self Acclimatize Genetic Algorithm (SAGA) with adaptive probabilities of crossover and mutation are to maintain the genetic diversity in the population and prevent the GAs to converge prematurely to local minima. Crossover rate and Mutation rate get modified using the following proposed formula.

$$\begin{aligned}
 & \text{if}(\text{fitness}' \geq \text{max_fitness}) \text{then} \\
 & \text{Crossover_prob} = \text{Crossover_prob}_1 - \frac{(\text{Crossover_prob}_1 - \text{Crossover_prob}_2)(\text{fitness}' - \text{avg_fitness})}{(\text{max_fitness} - \text{avg_fitness})} \\
 & \text{else} \\
 & \text{Crossover_prob} = \text{Crossover_prob}_1 \\
 & \text{if}(\text{fitness}' \geq \text{max_fitness}) \text{then} \\
 & \text{Mutation_prob} = \text{Mutation_prob}_1 - \frac{(\text{Mutation_prob}_1 - \text{Mutation_prob}_2)(\text{fitness}' - \text{avg_fitness})}{(\text{max_fitness} - \text{avg_fitness})} \\
 & \text{else} \\
 & \text{Mutation_prob} = \text{Mutationr_prob}_1
 \end{aligned}$$

Where max_fitness is the highest fitness value in the population.

avg_fitness is the average fitness value in every population.

$\text{fitness}'$ is higher fitness value between two individuals.

$\text{Crossover_prob}_1 = 1.0$, $\text{Crossover_prob}_2 = 0.7$

and $\text{Mutation_prob}_1 = 0.2$, and $\text{Mutation_prob}_2 = 0.01$.

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. Single point crossover with probability of crossover=1.0 and mutation with probability of mutation=0.1. 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 $m = 2^M - 1$. 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 r_{i1} 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 SAGA based key generation algorithm will generate best fittest key stream having length of 128 bits.

```
1001101101011110110011011001011101010100110100011010101010110011
0100101001110001010101011001110011111001011011101101111100110101
```

Now, consider the plain text to be encrypted is “Network Security”

```
01001110/01100101/01110100/01110111/01101111/01110010/01101011/00100000/
01010011/01100101/01100011/01110101/01110010/01101001/01110100/01111001
Here “/” is used as the separator between successive bytes.
```

Now, perform XOR operation between plain text and SAGA based key stream.
So, level 1 i.e. SAGA based key stream encoded cipher text is

```
1101010100111011101110011110000000111011101000111100000110010011
0001100100010100001101101110100110001011000001111010101101001100
```

4. RESULTS

Table 2 represents 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 2Average of fitness values

Number of Generations	Average of fitness values
40	35.8350
60	36.2346
80	36.9535
100	38.5472

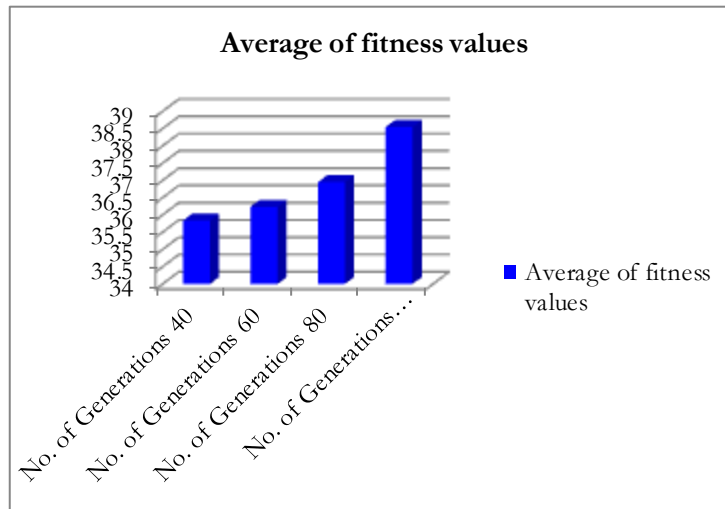


Figure 1 Number of Generation Vs Average of Fitness Values

Table 3 tabulated the fitness values of 50 numbers of iterations and the average fitness value of 50 iterations is 37.116146.

Table 3 List of fitness values in 50 iterations

Iteration	Fitness Value
1	34.1069
2	39.4297
3	32.9237
4	38.3263
5	29.5436
6	44.0764
7	43.3057
8	32.1490
9	31.4927
10	37.5192
11	29.4037
12	36.4917
13	40.1126
14	37.5078
15	28.0337
16	49.2654
17	37.4158
18	33.7915
19	29.4991
20	42.2783
21	37.0247
22	36.9162
23	38.4713
24	35.7240
25	36.8639

26	37.0257
27	37.8749
28	50.2952
29	37.0359
30	31.2819
31	27.3074
32	38.3809
33	39.5937
34	47.1729
35	42.2964
36	47.3062
37	37.4189
38	27.2190
39	26.1183
40	30.2973
41	43.0401
42	27.5291
43	49.6033
44	25.0072
45	39.3781
46	32.6194
47	40.2051
48	42.6397
49	37.1893
50	50.2985

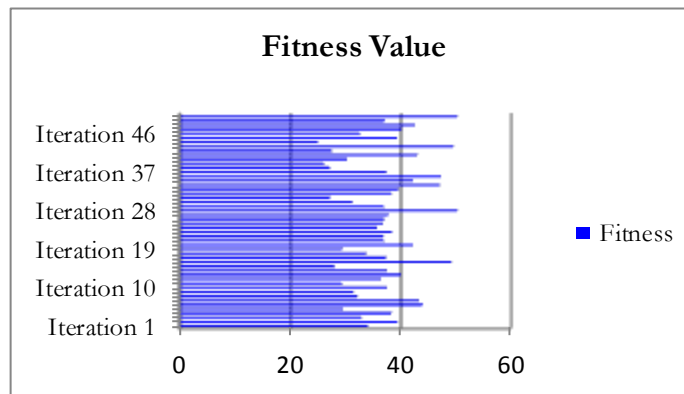


Figure 2 Graph Representation of No. of Iteration Vs Fitness Values

Table 4 shows the comparison results among GAKG, AES, RC4 and Vernam Cipher.

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

Length of Plain text	Key Storage Proposed (GAKG)	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

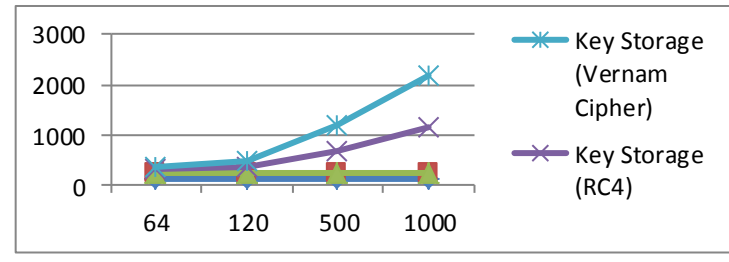


Figure 3 Comparison of key storage in Proposed GAKG, AES, RC5, RC4 and Vernam Cipher

5. CONCLUSION

In GAKG 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 GAKG key stream size is 128. If number of bits in a plain text is greater 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. 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

International Journal on Cybernetics & Informatics (IJCI) Vol.2, No.5, October 2013
(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 Conference 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.

Authors

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.

