International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Technique to Generate the Secure Symmetric Key Under Variable Keys Environment using Genetic Algorithm

Chukhu Chunka, Subhasish Banerjee, Rajat Subhra Goswami

Abstract: ***The purpose of information security is to protect information or data from misuse, unauthorized access and also to ensure the secured communication between transmitter and receiver. In this regard, one of the primary and foremost necessity is to protect the key by any means and should also be unbreakable. In this context, a mechanism namely, Automatic Variable Key (AVK) has been introduced to maintain the secrecy of the key. However, in this approach the initial key must be established earlier, then only it changes the keys and make unpredictable to guess, for every new block of data transmission. Thus, to overcome from this extra burden of initial distribution of key, we propose a new technique using Artificial Intelligent (AI) in order to generate the initial key automatically using the Genetic Algorithm (GA). We have demonstrated the importance of AI in the area of cryptography by our scheme. A comparative study is also carried out with the existing schemes to prove the proficiency of proposed scheme. Thus, in order to prove the unpredictability among the auto-generated keys, we have verified the randomness with the help of National Institute of Standards Technology (NIST) Test suite.***

Keywords: Randomness, Hamming distance, Genetic Algorithm, Automatic Variable Key, Artificial Intelligence.

I. INTRODUCTION

In the advancement of Internet technology, illegal data access, unwanted modification, disruption become a burning issue in data communication. Hence, it is necessary to protect the important information from the unauthorized user. In order to do so, various types of cryptographic schemes have already been proposed and implemented by many researchers. An AI system has capabilities to perform the intelligent task effectively and efficiently in the various field of cryptography and information security. Thus, many researchers [1]-[7], [23]-[25] have developed the algorithms to generate the keys using GA which are more complex and also difficult to predict. The most important part of the GA is selection, crossover and mutation operators where operators are used for generating very complex keys which are random in nature. The randomness of the key basically depends on the fitness function of GA which vary vastly in the key generation techniques under cryptography. In this paper, we have used the GA to resolve the issue of initial key (1st key) exchanged between sender and receiver. In AVK, the initial key requires to be exchanged between sender and receiver using various cryptographic algorithms like Rivest–Shamir -Adleman (RSA) or by the key distribution center or any other Internet Key Exchange (IKE's) protocols. Hence, to overcome the problem of distribution of 1st key, we have proposed a new mechanism to generate the initial key by using AI of GA.

The paper has been organized as follows. Section 2 describes working principle of AVK, section 3 consists of introduction of GA. The detailed of the proposed work is defined in section 4. the examples of key generation are discussed in section 5. Experimental results are given in section 6 and section 7 performance examination. Randomness verification with the help of NIST test suite is discussed in section 8 and finally the paper is concluded in section 9.

II. WORKING PRINCIPLE OF AUTOMATIC VARIABLE KEY

To fulfill the daily requirement whether for E-Commerce or online trading, the Internet is a primary necessity. From the last few decades, maintaining the security becomes a burning issue among the researchers [8]-[10]. Shannon [11][12] had acquainted the notion of perfect secrecy and demonstrated that the possible to change the key from session to session or time to time (better known as one time pad; also called as Vernam cipher) and the size of key is long as that of the plaintext and also, ciphertext only attack will be completely infeasible. The main purpose of AVK [13] [14] is to change the key automatically block by block of data transmission. In this section, we have described the working principle of standard AVK where the new keys are always generate based on the previous key and previous block of data. The detailed description is mentioned as below:

Key Generator (Initial Key)
{
$Key_i \leftarrow$ *Initial Key*
i ←2
while ($D_i \neq \emptyset$)
{

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Chukhu Chunka*, Department of Computer Science and Engineering, National Institute of Technology, Arunachal Pradesh, India, Email: chukhuchunka20@gmail.com
Subhasish Banerjee, Department of Computer Science and Engineering, National Institute of Technology, Arunachal Pradesh, India, Email: subhasishism@gmail.com
Rajat Subhra Goswami, Department of Computer Science and Engineering, National Institute of Technology, Arunachal Pradesh, India, Email: rajat@nitap.ac.in

$Key_i \leftarrow Key_{i-1} \oplus D_{i-1}$

i + +

}

}

Where:

D_i = i[th] block of data

In the continuation of these, several researchers like Chakraborty et al. [15], Goswami et al. [16][17], Dutta et al. [18], Singh et al. [19], and Banerjee et al. [20] have contributed by proposing many new protocols to make the key variable in nature. But deficiency lies in initial key distribution, that is through RSA or key distribution center. So, to overcome the key distribution GA is incorporated with AVK.

III. GENETIC ALGORITHM

GA is an optimization method dependent on mechanics of hereditary and evaluation of natural systems [22]. The three fundamental operators utilized in hereditary calculations are selection, crossover, and mutation. The GA generations are generated by the following cycle as shown in Fig.1 until the fitness value is reached or satisfied.

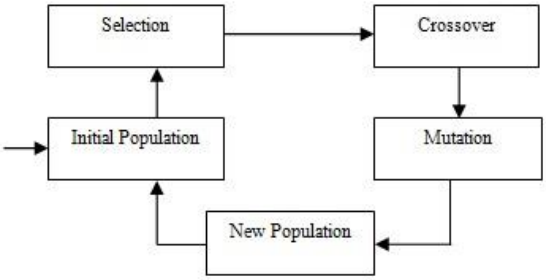

Fig.1 Basic Genetic Algorithm Process

The above Fig.1 shows how the GA works. The process starts with an initial population. From the initial population, individual with the most elevated wellness worth or fittest chromosome is chosen for further processing. The fitness value is determined from the population. The new population is generated using the crossover operation and mutation operation. The acquired recently population consist of best individual which is selected from population.

- **Selection:** chromosomes are chosen from population to imitate new chromosomes based on wellness worth or fitness value of chromosomes.
- **Crossover:** new (recently acquired) chromosomes are generated by performing crossover operation on two chromosomes. The current chromosomes consisting of some features of first chromosomes and the rest over from the subsequent chromosome. Example

Chromosome1: 1101**0011**	Offspring 1:1101**0101**
Chromosome2: 0101**0101**	Offspring 2: 0101**0011**

- **Mutation:** One generation to the next generation of population mutation is necessary to maintain genetic diversity. Example

Chromosome1: **1**0101011	Offspring 1:**1**1101011
Chromosome2:01110**1**1	Offspring 2:01110**0**1

IV. PROPOSED WORK

In this section, we have defined our proposed scheme "Genetic Algorithm for Automatic Variable Key" (GAAVK) where, GA is used to resolve the issue of initial key i.e. 1st key. The initial key needs to be exchanged between sender and receiver prior to AVK technique. In proposed scheme, two initial keys are generated, 1st initial key and 2nd round initial key. To generate initial key, perform the fitness function 1 *(FF1*), crossover, mutation and fitness function 2 *(FF2);* if *FF2* is satisfied then select 1st initial key else, perform *FF1*, crossover, mutation and *FF2* at rest of chromosomes keys. To select 2nd key for encryption, perform XORed previous key (initial key) with last dataset transmitted. After that, to generate 3rd key, performs the same steps *FF1*, crossover, mutation and *FF2* on initial population to select 2nd round key initial key. The 3rd key=2nd key ⊕ previous dataset transmitted ⊕ 2nd round initial key (best fit key). Later, the dynamic key or the rest of the keys are generated using the AVK mechanism. In this scheme only one generation is considered to select best key or initial key from population and initially population is selected randomly. The two best fit chromosome keys are chosen by applying *FF1* at initial population.

FF1 = If((count1<=n/4 &&(count1<=n/2)) and ((count2<=n/4)&&(count2<=n/2)) then perform one-point crossover operation at *n*/2 chromosome key and mutation operation at 100 bits place. Here, *count 1* and *count 2* is the number of set bit in key. To select best key (initial key) further perform fitness function 2 *(FF2).*

FF2= minimum counting pattern "110101" encountered in K_i *and* K_{i+1}.The best fit key is selected if the chromosome key is encountered with minimum pattern otherwise, it discards the two selected keys and repeat, the procedure of *FF1*, crossover, mutation and *FF2* till the minimum pattern encountered. The complete description of the proposed algorithm is described as below:

Population = P =[ ]nxm

Key_Generation (P, Dataset [ ][128])

{

do

{

$i \leftarrow 1$

$j \leftarrow 1$

$K_i \leftarrow P\ [1]\ [\ ]$

$K_{i+1} \leftarrow P\ [2]\ [\ ]$

Count1 ← no. of 1's in K_i *, P [1:64]*

Count2 ← no. of 1's in K_{i+1} *, P [1:64]*

// Select K_i *and* K_{i+1} *from initial population for further process, if the condition FF1 is satisfied then perform crossover operation.*

FF1 = If ((count1<=n/4 &&(count1<=n/2)) and ((count2<=n/4)&&(count2<=n/2))

// Crossover by swapping 1st Half of K_i *and* K_{i+1} *i.e.*

SWAP (K_i *[1:64];* K_{i+1} *[1:64])*

// Mutation of K_i *and* K_{i+1} *by flipping the binary bits at 100th position i.e.*

$K_i\ [100] \leftarrow K_i\ [100] \oplus 1$

$K_{i+1}\ [100] \leftarrow K_{i+1}\ [100] \oplus 1$

// Apply FF2 to select best fit chromosome key.

FF2= minimum counting pattern "110101" encountered at K_i *and* K_{i+1}.

$C_1 \leftarrow cmp\ (K_i, \text{“110101”})$
$C_2 \leftarrow cmp\ (K_{i+1}, \text{“110101”})$
If ($C_1 < C_2$)
{
Key[j][ ] ← K_i // assume as 1st best fit chromosome key for encryption.
else
Key[j] [ ] ← K_{i+1} // terminate and look for other possibilities (apply FF1) in initial population.
}
//to generate 2nd chromosome key= 1st best fit (initial) chromosome key transmitted is XORed with last transmitted dataset.
Key[j+1][ ] ← Key [j][ ] ⊕ Dataset [j:128] // 2nd key
i ← i+2
j ++
// to generate 3rd chromosome key apply FF1, crossover, mutation and FF2 as same steps are applied for selecting 1st best fit chromosome key and rest or dynamic key are generated by applying AVK mechanism
Key [j+1][3] ← Key[j+1][ ] ⊕ last Data block (Dataset [j:128]) ⊕ 2nd round (initial key or best fit chromosome key) key
// apply AVK mechanism to generate dynamic key and terminate the key generation when Dataset=0
Key [j+1][4] ← Key[j+1][3] ⊕ last Data block (Dataset [j:128])
} while (Dataset[j+1][ ] ≠ Ø)
}
Where: *n* = 128-bit length of the key
cmp = compare
m = Nos. of key

A. Explanation of proposed scheme of key generation technique

i. In this scheme, initial population is selected randomly and only one iteration is used to perform to select best key or initial key. The population size of chromosomes key is assumed to 34 and length of each keys are of 128 bits. The selection of two best chromosome keys or parents are selected based on the fitness function 1 *(FF1)*.
ii. *FF1:FF1*= Count the no. of 1's in first half of chromosome key (*n/2*), where *n*=128. If the numbers of 1's are between 32 to 64, then it will be selected as the parent for further process. Otherwise, it will be discarded. Thus, the crossover function is performed after the selection of two chromosomes.
iii. Crossover: Crossover originally gives the sexual genetic new combination. Two chromosome keys are randomly selected using the *FF1* and perform one-point crossover on two individual chromosome keys. The successor of the generated chromosome keys is very different from their initial parent chromosome keys.
iv. Mutation: After the crossover, mutation operation is performed on the selected chromosome keys of binary bit at 100th positions i.e. 1 is flip to 0 and vice versa.
v. *FF2:* To select fittest key between the individual chromosome key. *FF2*= chromosome key encountered with minimum pattern bits of {110101} is selected as the best fit chromosome key or best fitness value which is used for encryption.
vi. XOR Function: To generate 2nd chromosome key for encryption, 1st key is XORed with previous transmitted dataset. Similarly, in order to generate 3rd key, we need to generate 2nd round best fit chromosome key (generated using same process as the algorithm goes through *FF1*, crossover, mutation and *FF2*). If the condition is satisfied on two selected chromosome keys then, perform crossover, mutation, and *FF2* else, discard the selected chromosomes key and again it is required to consider the initial population for other two chromosome keys which satisfy the above *FF1* condition. 3rd chromosome key = 2nd chromosome key ⊕ last transmitted dataset ⊕ 2nd best fit (2nd round initial key) chromosome and to generate dynamic key or 4th key apply standard AVK mechanism. These processes will be continued till dataset =null.

V. EXAMPLES OF KEY GENERATION

Here, we have illustrated the examples how the initial key and rest of the keys are generated using GA and AVK mechanism. We have considered chromosome key in hexadecimal format of 128 bits long Key_1' = 6F7675656E657D774325492944EA5232 and chromosome Key_2' = E7FFE97FFD2BFFF97C956AB89C96A93F which are selected from the population using the *FF1*.

i. *FF1* = count numbers of 1's in the first half of the chromosome key. If the number of 1's in first half is between 32 to 64, then consider for further process. The numbers of 1's found in Key_1' is 41 and same as one more chromosome key is selected Key_2' which has the numbers 1's is 51 which satisfy the FF1.
ii. Crossover Operation: One-point crossover is performed in two individual chromosomes keys. chromosome key 1: Key_1' = 6E7675656E657D777**C956AB89C9 6A93F** chromosome key 2: Key_2' = E7FFE97FFD 2BFFF9**4325492944EA5232**
iii. Mutation: Mutation point is performed on selected individual chromosome key at bit position 100 is inverted. If 1's then flip to 0's and vice versa, chromosome key 1: Key_1' = 6E767565 6E657D777C956AB89**D**96A93F, where, 0 is flip to 1 (i.e. C to D) chromosome key 2: Key_2' = E7FFE97FFD2BFF F94325492945EA5232. 1 is flip to 0 (i.e. 4 to 5).
iv. *FF2:* To get 1st chromosome key finds the minimum pattern of "110101" in selected Key_1' and Key_2'. The chromosome key with the minimum pattern is selected as the 1st best fit chromosome key for encryption. For the given example, chromosome Key_2' has the minimum pattern of "110101" and select as the key for encryption, Key_2' is considered as the 1st best fit chromosome key (initial key) generated using the GA.
v. To generate 2nd chromosome key = E7FFE97FFD2B FFF94325492945EA5232 (previously transmitted key (Key_2'))⊕AB154EB 66827AACDDF30AAC 3C11B2A98 last transmitted dataset (D_1). Key_2 = 4CEAA7C99 50C55349C15E3E684F178AA as the 2nd chromosome key for encryption.
vi. To generate a 3rd key Key_3, similarly, apply *FF1*, crossover and mutation over the two selected chromosome which satisfies the *FF1* condition. After that, apply *FF2* to select 2nd best key (2nd round initial key), if the chromosome key satisfies the condition, consider as 2nd best key. And perform AVK technique.

Here, let consider 2nd best chromosomes key is A4BEE21459FFDABCC BDDEE58921ABDEC. Then, (Key_3) = 4CEAA7 C9950C55349C15E3E684F178AA (previous selected key Key_2) ⊕ 1EF3456FFADCB89FFB BCCDA43267FE23(D_2 ⊕ A4BEE21459FFDABC CBDDEE58921ABDEC (2nd best key selection) and rest of the key are generated as per the AVK mechanism.

VI. EXPERIMENTAL RESULTS

In this section, we have carried out three experiments for various data sets to demonstrate the differences in terms of Hamming distance. The hamming distance of two successive keys is estimated as the number of contrasts between the relating bits. Following are the list of the parameters which are characterized to show the experiments.

- Randomly selected initial population = 34
- Length of the chromosomes key = 128 bits
- Block size of the message is considered =128 bits
- For plotting a diagram, we have considered 20 keys only.
- HD stand for Hamming distance.

To perform the experiments 1 to 3 we consider the Data set which is given below.

Experiment-1: To check the hamming distance among the successive key, the Dataset as: "*Data Acquistion: Fingureprint data is aquired when subjects firmly press their fingure against a polycarbonate plate. The fingerprint image is not stored. Information on the relative location of the ridges, whorls, bifurcations, and intersections is stored as an enrolled user data base file and later compared with user input data.*" The result of variation among the auto-generated successive key for the following algorithm AVK, ASAVK, CSAVK and GAAVK are shown in Fig.2.

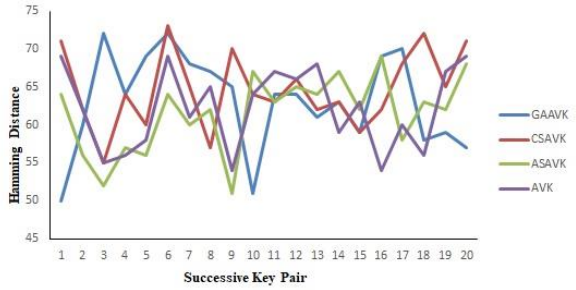

Fig.2 HD among the successive keys for Dataset-1

Experiment-2: The dataset considered for the experiment-2 is as follows: "*Unique System Aspects: To avoid the dirt build-up problem, a newly developed fingerprint acquires the fingerprint image with ultrasound. Claims are made that this system can acquire the fingerprint of a surgeon wearing latex gloves. A number of companies are producing fingerprint based biometric identification systems.*" The hamming distance of auto generated successive keys is shown in Fig. 3.

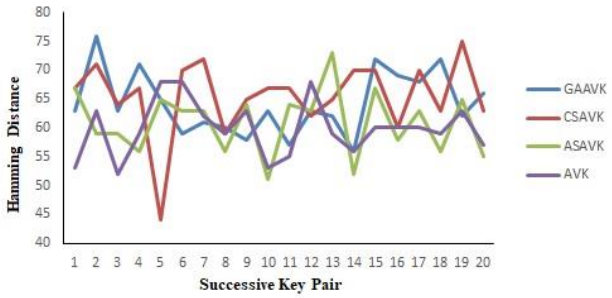

Fig.3 HD among the successive keys for Dataset-2

Experiment-3: Following dataset is considered for the experiment-3: *"User Action Required: Hand geometery system operates only as an identification verifier. The user is verified by entering a PIN on a keypad. When the place hand message appears on the display, the user places the hand flat on the platen against the pegs.If correct hand position the data are acquired and a remove hand message appears."* HD of auto generated graph is shown in Fig. 4.

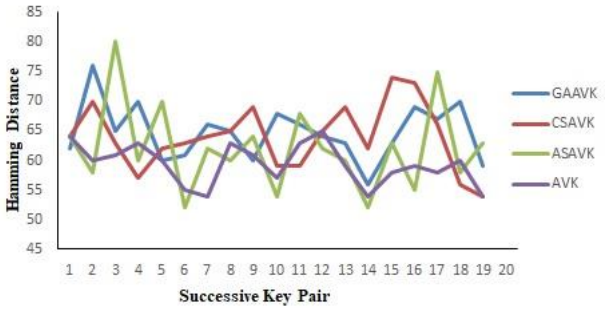

Fig.4 HD among the successive keys for Dataset-3

From Fig.2,3 and 4, graphs show that HD among the successive keys of GAAVK has higher variation than other existing schemes.

VII. PERFORMANCE EXAMINATION

In this section, we have estimated the efficiency of our scheme with other related existing schemes like AVK [13], Alternating and shifting Automatic variable key [16] (ASAVK), Computing shifting Automatic variable key [17] (CSAVK) with our scheme i.e. Genetic Algorithm for Automatic Variable Key (GAAVK) based on Average Hamming distance (AHD) and Standard Deviation (SD) which determined among the auto-generated keys. The two equations of AHD and SD are given below. In equation AHD stand for Average HD, SD standard deviation, HDKP as HD key pair, NHDKP as number of HD key pair. The experiments-1,2 and 3 are illustrated in Fig.5 and Fig.6.

$$AHD = \sum (HDKP/NHDKP) \dots\dots\dots\dots\dots (i)$$

$$SD = \sqrt{(\sum \frac{(HDKP - AHD)2}{NHDKP})} \dots\dots\dots\dots\dots (ii)$$

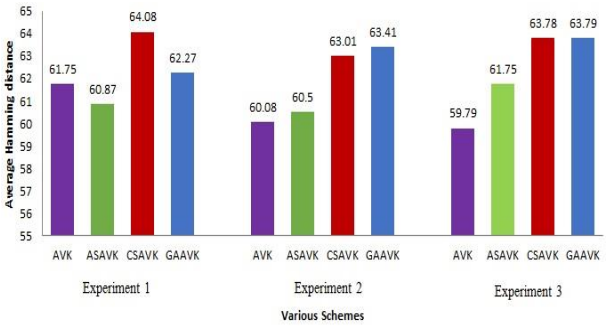

Fig.5 Average HD of GAAVK and numerous schemes

From above Fig. 5 and Fig. 6 we have seen that our scheme has a superior AHD and Standard Deviation (SD) than the different schemes.

Retrieval Number: A1178109119/2019©BEIESP
DOI: 10.35940/ijeat.A1178.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VIII. RANDOMNESS VERIFICATION

In this section, we have checked the randomness among the successive keys for experiment 1, 2 and 3. The NIST test suite version 2.1.2 is used to check the random and pseudorandom number of cryptography application [21].

In NIST test suite there are 15-different tests to verify the randomness. To prove the randomness of our scheme we have conducted four tests. List of statistical tests are:

a. Frequency Test (FT)
b. Block Frequency (BF)
c. Run Test (RT)
d. Cumulative Sums forward/backward (CSF/CSB)

To test the experiments, we consider the length as per the requirement. The above test a, b, c and d mainly focus on the P-Value which defines the strength of the randomness. As per the test suite if P-Value> 0.01 then it has the randomness otherwise, auto-generated keys are not random. To verify the randomness of auto-generated keys, we have tested all key sequences which are tabulated below.

Table I. Statistical Test for Experiment-1

Experiment-1			
Sl No.	Test	P-Values	Results
a.	FT	0.122325	Random
b.	BF	0.122325	Random
c.	RT	0.350485	Random
d.	CSF	0.350485	Random
e.	CSB	0.534146	Random

Table II. Statistical Test for Experiment-2

Experiment-2			
Sl No.	Test	P-Values	Results
a.	FT	0.739918	Random
b.	BF	0.534146	Random
c.	RT	0.534146	Random
d.	CSF	0.739918	Random
e.	CSB	0.350485	Random

Table III. Statistical Test for Experiment-3

Experiment-3			
Sl No.	Test	P-Values	Results
a.	FT	0.350485	Random
b.	BF	0.350485	Random
c.	RT	0.350485	Random
d.	CSF	0.534146	Random
e.	CSB	0.213309	Random

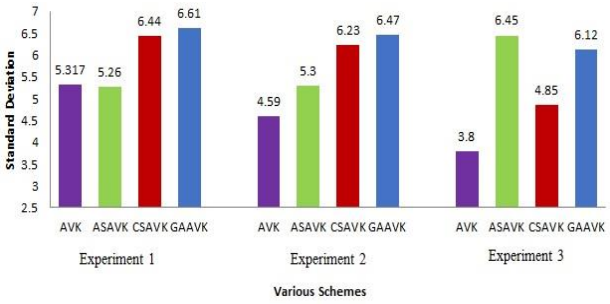

Fig.6 SD of GAAVK and numerous schemes

IX. CONCLUSION

In this paper, a new approach namely, GAAVK has been proposed using AI where the initial key is generated automatically on both side using the GA fitness function and then AVK is used to generate dynamic keys for encryption. The superiority, of our scheme relay on, key exchange i.e. it does not require key exchange between the sender and receiver. Secondly, comparative study with the other related existing schemes based on AHD and SD has been carried out. As per the experimental results, conclusion can be made:

- AHD: Based on the experimental results, our new proposed scheme could contribute the higher Hamming distance than others existing schemes.
- SD: Superior results over the other related existing schemes.
- NIST: Finally, we have done different NIST statistical tests namely Frequency Test, Block Frequency Test, Cumulative Sum Test, Run Test and has demonstrated that our proposed scheme could persist to maintain the required P-value>0.01.

Considering all the above points, we can conclude that our scheme is superior than other existing techniques of the AVK. Therefore, our scheme is more reliable and efficient. Thus, it can practically implemented in the cryptosystem to increase the security into next higher level.

REFERENCES

1. E. Rich, K. Knight, and S.B. Nair, "Artificial Intelligence," Third edition, McGraw Hill publication. 2008
2. A. Kumar, and M.K. Ghose.2009, "Overview of information security using genetic algorithm and chaos," *Information Security Journal: A Global Perspective*, vol.18, issue 6, pp.306-315.
3. A. Soni, and S. Agrawal. 2013, "Key generation using genetic algorithm for image encryption," *International Journal of Computer Science and Mobile Computing*, vol.2, issue 6, pp.376-383.
4. S. Jawaid, and A. Jamal.2014, "Generating the best fit key in cryptography using Genetic Algorithm," *International Journal of Computer Applications*, vol.98, issue 20, pp.33-39.
5. K. Sindhuja, and S.P. Devi. 2014, "A Symmetric Key Encryption Technique Using Genetic Algorithm," *Journal of Computer Science and Information Technologies*, vol.5, issue 1, pp.414-416, ISSN: 0975-964.
6. S. Bhowmik, and S. Acharyya. 2011, "Image cryptography: The genetic algorithm approach," *In Computer Science and Automation Engineering (CSAE), IEEE International Conference on IEEE*, vol. 2, pp. 223-227.
7. A. Kumar, and K. Chatterjee. 2016, "An efficient stream cipher using Genetic Algorithm," *In Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on IEEE*, pp. 2322-2326.
8. E. Hans. 1992, "A High Speed DES Implementation for Network Application," *Proc. Int. Conf. Cryptology*, pp.521-539.

9. B. Eli. 1997, "A fast new DES implementation in Software," *Proc. Int. Symp. Foundations of Software Engineering*, pp.260-273.
10. C.T. Li, C.C. Lee, C.J. Liu, C.W. Lee. 2011, "A robust remote user authentication scheme against smart card security breach," *In IFIP Annual Conference on Data and Applications Security and Privacy, Springer*, Berlin, Heidelberg, pp. 231-238.
11. C.E. Shannon. 1948, "Mathematical theory of communication," *The Bell System Tech J*, 27, pp. 379-423.
12. C.E. Shannon. 1949, "Communication Theory of Secrecy System," *The Bell System Tech Journal*, vol.28, pp.656–715, doi:10.1002/j.1538-7305. 1949. tb00928.x
13. C.T. Bhunia. 2006, "Application of AVK and selective encryption in improving performance of quantum cryptography and networks," *United Nations Educational Scientific and Cultural Organization and International Atomic Energy Agency, retrieved*, vol.10, issue 12, p.20010.
14. C.T. Bhunia, G. Mondal, and S. Samaddar. 2006, "Theory and application of time variant key in RSA and that with selective encryption in AES," *Proceedings of EAIT (Elsevier Publications, Calcutta CSI)*, pp.219-221.
15. P. Chakrabarti, B. Bhuyan, A. Chowdhuri, and C. Bhunia. 2008, "A novel approach towards realizing optimum data transfer and Automatic Variable Key (AVK) in cryptography," *International Journal Computer Science and Network Security*, vol.8, issue 5, pp.241.
16. R.S. Goswami, S.K. Chakraborty, A. Bhunia, and C.T. Bhunia. 2013, "New approach towards generation of Automatic Variable Key to achieve Perfect Security," *In Information Technology: New Generations (ITNG). Tenth International Conference on IEEE*, pp. 489-491.
17. R.S. Goswami, S.K. Chakraborty, A. Bhunia, and C.T. Bhunia. 2014, "New techniques for generating of automatic variable key in achieving perfect security," *Journal of The Institution of Engineers (India): Series B*, vol.95, issue 3, pp.197-201.
18. M. P. Dutta, S. Banerjee, and C.T. Bhunia. 2015, "An approach to generate 2-Dimensional AVK to enhance security of shared information," *International Journal of Security and Its Applications*, vol.9, issue 10, pp.147-154.
19. B. K. Singh, S. Banerjee, M.P. Dutta, and C.T. Bhunia. 2016, "Generation of automatic variable key to make secure communication," *In Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing Springer,"* pp. 317-323.
20. S. Banerjee, M.P. Dutta, and C.T. Bhunia. 2017, "A Novel Approach to Achieve the perfect security through AVK over Insecure Communication Channel," *Journal of The Institution of Engineers (India): Series* B. vol.98, issue 2, pp.155-159.
21. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," *Booz-Allen and Hamilton Inc Mclean Va*, 2001
22. G.A. Vijayalakshmi Pai, and S. Rajasekaran, "Neural Networks, Fuzzy Logic, and Genetic Algorithms," *Synthesis and Applications*, 2012.
23. S. Zhou, J. Xie, N. Du, and Y. Pang. 2018, "A random-key genetic algorithm for scheduling unrelated parallel batch processing machines with different capacities and arbitrary job sizes," *Applied Mathematics and Computation*, vol.334, pp.254-268.
24. G. J. Krishna, V. Ravi, and S. N. Bhattu. 2018, "Key generation for plain text in stream cipher via bio-objective evolutionary computing," *Journal of Applied Soft Computing*, vol. 70, pp. 301-317.
25. W. Jing, and K. Shimada.2018, "Model-based view planning for building inspection and surveillance using voxel dilation, Medial objects, and Random-Key Genetic Algorithm," *Journal of Computational Design and Engineering*, vol.5, pp.337-347.

AUTHORS PROFILE

Chukhu Chunka pursing Ph.D. in Computer Science and Engineering at NIT Arunachal Pradesh and completed her M.Tech degree in Computer Science and Engineering from NERIST (Deemed University) Arunachal Pradesh India in 2015. She did her B. Tech from Computer Science and Engineering from Central Institute of Technology Kokrajhar Assam 2013. Research Interests area are Cryptography and Information security.

Dr. Subhasish Banerjee received his Ph.D. in Computer Science and Engineering from National Institute of Technology, Arunachal Pradesh, in 2016 and M.Tech degree in Computer Application from Indian Institute of Technology (ISM), Dhanbad, India in 2012. Currently he is working as Assistant Professor in the Department of Computer Science and Engineering in National Institute of Technology, Arunachal Pradesh. His research activities are mainly focused on cryptography, networking and information security.

Dr. Rajat Subhra Goswami working as an Assistant Professor in the department of Computer science and Engineering in National Institute of Technology, Arunachal Pradesh, Govt. of India. He is having more than 8 years of teaching and research experience. His research areas are cryptography and Image Processing.