PSEUDO RANDOM KEY GENERATOR USING FRACTAL BASED TRELLIS CODED GENETIC ALGORITHM FOR IMAGE ENCRYPTION

Anusha .T and Venkatesan R

Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

Cryptographic applications such as online banking, and securing medical and military data require the usage of random keys, which should remain unpredictable by adversaries. This paper focuses on the strengths and limitations of the techniques and algorithms that are used in the generation of random keys and a new method to generate random keys is proposed using fractals. Fractals are generated using the Sierpinski triangle and fed as input for Non-Deterministic Finite Automata (NDFA) to generate an Initial Vector (IV). Trellis Coded Genetic Algorithm (TCGA) code generator generates seed value using IV as input. Pseudo-Random Key Generator (PRKG) generates a Session Key matrix (SKM) using a seed value. Images are encrypted using SKM to generate cipher images. The randomness of the TCGA code is tested using entropy measure and efficiency based on NIST Tests. SKM with high entropy value is used for image encryption. The Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) values are used for calculating the randomness of cipher images.

KEYWORDS

Finite Automata, Pseudo Random Key Generator, Session Key Matrix, Trellis Coded Genetic Algorithm, Security

1. INTRODUCTION

Today, security has become a major social issue by all means. Cryptographic techniques and algorithms play a vital role in maintaining confidential information in an unintelligible form with the help of random keys. Random keys rely on high entropy sources such as an image, video, audio, and fractals. There are various types of random number generators viz., True Random Number Generator (TRNG), Pseudo Random Number Generator (PRNG), Pseudo Random Function (PRF), Uniform Random Number Generator (URNG), and Secure Uniform Random Number Generator (SURNG). Fractals are geometrically constructed figures using an initiator and generator. Part of the fractals is converted into a sequence of strings. Acceptance of strings is determined by NDFA, which is designed to accept a certain set of strings belonging to a language. NDFA generates an Initial Vector (IV) of variable length. TCGA code is generated using IV. The sequence of bits generated is converted to a decimal number which acts as the seed value for PRKG. PRKG generates Session Key Matrix (SKM) which is used for encrypting images. Trellis codes generated in our work are of 32-bits, which are then shuffled using a randomly chosen genetic algorithm technique. The Crossover phase of genetic algorithms plays a vital role in random key generation. One-point crossover, two-point crossover, uniform crossover, precedence preservative crossover, and ordered crossover techniques are chosen randomly during random key generation. The Number of Pixel Change Rate(NPCR) indicates the percentage of difference in pixels between two images. The Unified

DOI: 10.5121/ijnsa.2023.15302

Average Changing Intensity (UACI) measure is used to compute the average intensity difference in pixels between two images.

The rest of the paper is organized as follows: Section 2 discusses the existing random key generation techniques and their applications. Section 3 discusses the proposed method. Section 4 depicts the experimental results and Section 5 concludes the work. Our proposed method is executed using MATLAB 2014a with Intel Pentium (III) processor, 2.00 GHz speed, 4.00 GB RAM, and 64-bit Operating System.

2. LITERATURE REVIEW

Aarthi Soni, and Suyash Agrawal [1] say that the key can be generated using the fitness values calculated using genetic algorithms, a class of optimization algorithms. In this paper, the maximum fitness value calculated after selection, crossover, and mutation is taken as the random key. Images can be secured using various encryption methods.

Christian Rathgeb, and Andreas Uhl [2] present a survey on biometric crypto-systems and cancellable biometrics. The paper explores the dependency on biometrics for random key generation. The success of a biometric system depends on the initiation of a key-release mechanism. A private template scheme and a quantization scheme are the two approaches for biometric key Generation.

R.Divya, and V.Vijayalakshmi [3] developed an algorithm for session key generation and say that the key generated from a biometric template cannot be compromised from the stored hash value of the template. The session key shared by the client and server is used only for a particular session since the template used for its generation is cancellable.

Emeka Reginald Nwogu [4] proposed a three-level security method for Internet Banking systems and used biometrics for entry-level identification. Gholamreza Hesamian and Mohammad Ghasem Akbari [5] proposed a method to fit a non-parametric regression model for fuzzy random variables.

Kishore Golla and S. PallamSetty proposed a security technique by incorporating reinforcement learning-based ClonQlearnintegrated with ECC (ClonQlearn+ECC) for a random key generation with improved network performance [6]. The scheme proposed by Li Ma, and Tieniu Tan [7] uses spatial filters to capture local characteristics of the iris to produce discriminating texture features, which helps in iris recognition. Manju Kumari et.al [8] proposed a technique for securing the encrypted image using image compression done with the help of Huffman coding. Masahito Kayasi, and Shun Watanabe [9] give the importance of entropy in assuring the randomness inherent in the sequence generated. Here the relative entropy is used between the generated random number and the ideal random number.

Image is scrambled using Finite Field Cosine Transform (FFCT) to yield an image with a uniform histogram and fractals can be used to generate a one-time keypad stream that acts as a random key [10]. Performance evaluation is carried out using statistical and sensitivity analysis techniques. Keys can be exchanged using fractal functions [11].In this paper, the method generates a key using the Mandelbrot function and exchanges it using the Julia function.

Cipher images can be created using a hybrid method of image encryption using fractals and XOR operation [12]. The complexity of decryption gets increased based on the number of levels used in the encryption process. The method is based on Mandelbrot set fractals and XOR operation. In this paper randomness of the image is increased by reducing the correlation coefficient of vertical, horizontal, and diagonal pixels. Nikolaos G. Bardis [13] proposed a method based on an advanced key management system for secured military communications. W.Puech [14] proposed a method for compressing medical images and securely storing medical data. Various sources that are used in TRNG are physical patterns, disk electrical activity, mouse movements, and instantaneous values of the system clock. Optoelectronic Random Bit Generators make use of the non-deterministic property of laser chaos [15]. The photonics-based approach overcomes the disadvantage of electronic jitter that arises in random bits generated from laser chaos [16]. The paper reviews various tests that are used to assess the randomness of the generated code such as the PSNR test, key space, sensitivity analysis, and correlation coefficient.

The image encryption process can be carried out using fractal functions such as the Mandelbrot set, Julia set, Hilbert Curve, 3D fractal, multifractal, Iterated Function Systems (IFS), and chaotic functions [17]. Sunil.V.K.Gadam and Manohar Lal [18] proposed a method to extract features from fingerprint biometric data using the stages of minutiae point extraction from a fingerprint, secured feature matrix generation, and key generation. The analysis is carried out using cancellable transform, feature matrix security, and irreversibility.

Szcepanski. J et.al [19] puts forth the fact that biological data can be used for generating sequences of random bits. The data can act as a session key in the Secure Socket Layer (SSL), Secure Shell (SSH), Pretty Good Privacy (PGP), or Secure Electronic Transaction (SET) communication protocols. An entropy measure is taken to assure the randomness of the data. The last Digit Fluctuation method is used to extract a random bit sequence from the biological data. Thamizhchelvy. K and G. Geetha [20] say that a message authentication image can be generated using fractals and chaos theory.

The random key generator can be designed by extracting the chaos present within an image generated using fractals or the image by itself can be used as a random key. A non-uniform random number generated with probability distribution Px can be converted to a uniform random number with the conversion rate H (Px). It was extended to a general source by Vembu and Verdu [21]. Yasutada Oohama [22] analyses the performance of the interval algorithm using real numbers for random number generation. The interval algorithm makes use of a tree data structure for representation.

The strengths and limitations of existing techniques for random number generation are discussed in Table 1 and Table 2. The techniques that exist in the literature focus on random number generation using biometrics, genetic algorithms, interval algorithms, photonics & LASER chaos, and fractals.

Table 1. Strengths of existing techniques for random number generation

S. No.	Techniques	Strengths
1	Biometrics	Biometrics is the one every human possess. It cannot be forgotten or stolen. In using a cancellable biometrics technique based on random key generation stolen template of biometrics is neglected and a new template is generated based on a cancellable transform.
2	Genetic Algorithms	Generates random key based on the simple processes behind genetic algorithm viz., selection, crossover, and mutation
3	Interval Algorithms	Uses a simple data structure and simple tasks such as tossing a coin in the generation of the random key.

4	Photonics & LASER Chaos	Truly random sequence which remains unpredictable by adversaries can be generated
5	Fractals	Fractals are hard to break and hence fractal-based random key is unpredictable.

Table 2. Limitations of existing techniques for random number generation

S. No.	Techniques	Limitations
1	Biometrics	Possessing a biometrics-capturing device is sometimes practically not feasible.
2	Genetic Algorithms	Genetic Algorithms rely on high entropy sources or any other previously generated sequence of bits. It merely processes the bits already generated.
3	Interval Algorithms	The usage of floating-point arithmetic is complex.
4	Photonics & LASERChaos	There will be practical difficulty while generating photonics and LASER Chaos-based random keys.
5	Fractals	Generating unique fractals is complex

3. PROPOSED METHOD

Sierpinski triangles are generated and chosen randomly at angles 0^o, 90^o, 180^o, and 270^o, the size of which is random. Generated Sierpinski triangle is converted into a sequence of code strings which is fed sequentially into a Non-Deterministic Finite Automata (NDFA).NDFA returns a sequence of 1's and 0s which is taken as the initial value for the TCGA code generator. Trellis codes are generated by generating 4 bits for every single bit fed as input. Additional bits are padded to make if the initial value fed into the TCGA has a length of fewer than 8 bits. Part of a fractal is coded into a set of strings and fed as input to the NDFA. The sequence of strings is fed as input into the NDFA and a sequence of 0s and 1s can be generated. The generated value is used as an initial value for generating TCGA Code. TCGA code is used as a seed value to generate SKM to encrypt an image to get the cipher image. SKM is generated by considering the value generated from the TCGA code generator as a seed value. TCGA code is converted into a decimal number (n).

The initial n value is taken as shown in the formula (3.1)

$$n=n \% 255 \tag{3.1}$$

An initial p-value is taken as the next prime value to n.

SKM is generated using Pseudo Random Key Generator (PRKG) using the recursive formula (3.2)

$$n=((n * p) + c) \bmod 256 \tag{3.2}$$

where n is the value of each pixel of SKM, p is the next prime number of n (3.3)

$$c = n \% 7 \tag{3.3}$$

TCGA code uses five different genetic algorithm crossover techniques which are randomly chosen:

One-Point Crossover
Two-Point Crossover
Uniform Crossover
Precedence-Preservative Crossover
Ordered Crossover

The randomness of the TCGA code is assessed using NIST tests. And its efficiency is calculated using the equation (3.4)

Efficiency=(No. of NIST tests whose p-value is greater than or equal to 0.01)/11 (3.4)

4. Experimental Results

Table 3 lists the NIST Tests for randomness used in this work. Table 4 shows the generation of seed values for different angles. For the same angle, different seed values can be generated. Table 5a and Table 5b show the calculated p-values for NIST tests carried over the TCGA code. Table 6 shows the entropy and efficiency values of the random keys. Table 7 shows the entropy of the SKM. Table 8 shows the entropy of plain and cipher images, NPCR, and UACI values.

Table 3. NIST Tests for Randomness

S. No.	Test No.	Tests for Randomness
1	T1	Frequency test
2	T2	Frequency block test
3	T3	Longest runs test
4	T4	Spectral test
5	T5	Non-overlapping template matching test
6	T6	Overlapping template matching test
7	T7	Approximate entropy test
8	T8	Binary matrix rank test
9	T9	Runs Test
10	T10,T11	Serial Test

Table 4. Seed Value Generation

S. No.	Angle (Degree)	Initialization Vector	Trellis Code	TCGA Code	TCGA Code Identifier	Seed Value
1	0	[1,0,1]	'111001111'	'1001101'	C1	77
2	90	[0,0,1]	'1'	'100000000'	C2	256
3	180	[1,0,0]	'111001010'	'11001011'	C3	458
4	270	[1,0,1]	'111001111'	'111011011'	C4	475
5	0	[1,0,1]	'111001111'	'111001111'	C5	463
6	90	[0,0,1]	'1'	'10'	C6	2
7	180	[1,0,0]	'111001010'	'101011000001'	C7	2753
8	270	[1,0,1]	'111001111'	'1001101'	C8	77
9	0	[1,0,1]	'111001111'	'10111001011'	C9	1483
10	90	[0,0,1]	'1'	'100000000'	C10	256

11	180	[1,0,0]	'111001010'	'1000100'	C11	458
12	270	[1,0,1]	'111001111'	'1001101'	C12	77
13	0	[1,0,1]	'111001111'	'111011011'	C13	475
14	90	[0,0,1]	'1'	'1'	C14	1
15	180	[1,0,0]	'111001010'	'111001010'	C15	458
16	270	[1,0,1]	'111001111'	'1001101'	C16	77

Table 5a. Calculated p-values for NIST Tests

S. No.	**TCGA Code Identifier**	**Tests**						
		T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
1	C1	0.7054	0.7165	0.0011	4.3133e-06	1	NaN	0.5440
2	C2	1	0.0009	0.0054	3.0461e-134	0.0408	0.1005	4.8613 e-07
3	C3	0.7389	0.2998	0.0026	0.4002	1	NaN	0.2864
4	C4	0.0956	0.0965	0.0026	9.6166e-05	1	NaN	0.4866
5	C5	0.0956	0.0965	0.0026	9.6165e-05	1	NaN	0.4866
6	C6	1	0.0009	0.0054	3.0461e-134	0.0408	0.1005	4.8613 e-07
7	C7	0.5637	0.4060	0.0027	0.4268	0.3442	0.0344	0.5077
8	C8	0.7054	0.7165	0.0011	4.3133e-06	1	NaN	0.5440
9	C9	0.0956	0.0965	0.0026	9.6166e-05	1	NaN	0.4866
10	C10	1	0.0009	0.0054	3.0461e-134	0.0408	0.1005	4.8613e-07
11	C11	0.7389	0.2998	0.0026	0.4002	1	NaN	0.2864
12	C12	0.7055	0.7165	0.0011	4.3133e-06	1	NaN	0.5440
13	C13	0.0955	0.2997	0.0026	9.6165e-05	1	NaN	0.6601
14	C14	1	0.0009	0.0054	3.0461e-134	0.0408	0.1005	4.8613e-07
15	C15	0.7389	0.2998	0.0026	0.4002	1	NaN	0.2864
16	C16	0.7054	0.7165	0.0011	4.3133e-06	1	NaN	0.5440

Table 5b. Calculated p-values for NIST Tests

S. No.	**TCGA Code Identifier**	**Tests**			
		T8	**T9**	**T10**	**T11**
1	C1	0	0	0.7881	0.8669
2	C2	0	0	1	1
3	C3	0	0	0.6950	0.3679
4	C4	0	0	0.5394	0.5738
5	C5	0	0	0.5394	0.5738
6	C6	0	0	1	1
7	C7	0	0	0.7358	0.8465
8	C8	0	0	0.7881	0.8669
9	C9	0	0	0.5394	0.5738
10	C10	0	0	1	1
11	C11	0	0	0.6950	0.3679

12	C12	0	0	0.7881	0.8669
13	C13	0	0	0.4060	0.2359
14	C14	0	0	1	1
15	C15	0	0	0.6950	0.3679
16	C16	0	0	0.7881	0.8669

Table 6. Entropy and Efficiency of Random Key

S. No.	TCGA Code Identifier	Entropy	Efficiency	Efficiency %
1	C1	0.9852	0.5455	54.55
2	C2	0.2006	0.4545	45.55
3	C3	0.9911	0.6364	63.64
4	C4	0.7642	0.5455	54.55
5	C5	0.7642	0.5455	54.55
6	C6	0.2006	0.4545	45.45
7	C7	0.9799	0.7273	72.73
8	C8	0.9852	0.5455	54.55
9	C9	0.9457	0.7273	72.73
10	C10	0.2006	0.4545	45.45
11	C11	0.9911	0.6364	63.64
12	C12	0.9852	0.5455	54.55
13	C13	0.7642	0.5455	54.55
14	C14	0.2006	0.4545	45.45
15	C15	0.9911	0.6364	63.64
16	C16	0.9852	0.5455	54.55

Table 7. Entropy of Session Key Matrix

S. No.	TCGA Code Identifier	Entropy of Session Key Matrix (Sh)
1	C1	7.9952
2	C2	7.9956
3	C3	7.9951
4	C4	7.5794
5	C5	7.9945
6	C6	7.9956
7	C7	7.9951
8	C8	7.9952
9	C9	7.9945
10	C10	7.9945
11	C11	7.9945
12	C12	7.9951
13	C13	7.5794
14	C14	7.9956
15	C15	7.9951
16	C16	7.9952

Table 8. Entropy of plain and cipher images, NPCR, and UACI values

S. No.	TCGA Code Identifier	Entropy of Plain Image (Sh)	Entropy of Cipher Image(Sh)	NPCR (%)	UACI (%)
1	C1	7.6533	7.9977	99.6353	29.3977
2	C2		7.9971	99.6780	29.3995
3	C3		7.9971	99.6368	29.4493
4	C4		7.9972	99.4492	29.3174
5	C5	6.9228	7.9970	99.5697	34.2904
6	C6		7.9974	99.6780	34.4080
7	C7		7.9951	99.6368	34.4082
8	C8		7.9974	99.6353	34.4824
9	C9	6.8419	7.9973	100	34.2186
10	C10		7.9970	100	34.3769
11	C12		7.9970	100	34.3587
12	C13		7.9951	100	34.2463
13	C14	7.4809	7.9972	99.9908	28.5852
14	C15		7.9971	99.9863	28.6166
15	C16		7.9968	99.9863	28.5553
16	C17		7.9972	99.9924	28.4638

5. CONCLUSIONS

In this paper, a detailed survey is performed on the role of random keys in cryptographic applications. The strengths and limitations of techniques and algorithms used in a random key generation are studied. The need for high entropy sources in the field of random key generation is analyzed. From the survey, it is observed that there is a scope for research based on random key generation. The limitations of this work are that the entropy and the efficiency are less than 0.5 and hence there is a need for the regeneration of random keys which consumes time and energy. TCGA adds security by generating random seed values. Generated SKM is also efficient so that images encrypted using SKM generated by the proposed PRKG exhibits expected NPCR and UACI values. The maximum value for entropy for an SKM is 8 Sh since it is an image whose individual pixel values can be represented with 8 bits. If the entropy of SKM is less than 6 Sh, it will not be used as a random key. The future enhancement is to identify the reason for obtaining keys that are not random. The key generated from certain IVs are not random and the seed value generation from such IVs can be prevented. Our future work focuses on the secure sharing of generated keys.

ACKNOWLEDGMENTS

The authors would like to thank the Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, and Anna University, Chennai for supporting this research work.

REFERENCES

[1] Aarthi Soni, Suyash Agrawal, (2013) "Key Generation using Genetic Algorithm for Image Encryption", *International Journal of Computer Science and Mobile Computing*, Vol. 20, No. 6.

[2] Christian Rathgeb, Andreas Uhl (2011), "A Survey on Biometric Cryptosystems and Cancellable Biometrics", Eurasip Journal on Information Security, No.3.

[3] R.Divya, V.Vijayalakshmi, (2015) "Analysis of Multimodal Biometric Fusion Based Authentication techniques for Network Security", *International Journal of Security and its Applications*, Vol. 9, No. 6, pp.239-246.

[4] Emeka Reginald Nwogu, (2014) "Improving the Security of the Internet Banking System Using Three-Level Security Implementation", *International Journal of Computer Science and Information Technology Security (IJCSITS),* ISSN:2249-9555, Vol.4, No.6.

[5] Gholamreza Hesamian and Mohammad Ghasem Akbari,(2017) "Nonparametric Kernel Estimation Based on Fuzzy Random Variables", *IEEE Transactions on Fuzzy Systems*, Vol. 25, No. 1.

[6] Kishore Golla and S. PallamSetty (2022), "An Efficient Secure Cryptography Scheme For New Ml-Based Rpl Routing Protocol In Mobile Iot Environment", International Journal of Network Security & Its Applications (IJNSA), Vol.14, No.2.

[7] Li Ma, Tieniu Tan (2003) "Personal Identification Based on Iris Texture Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No.12.

[8] Manju Kumari, Vipin Pawar, and Pawan Kumar (2019), "A Novel Image Encryption Scheme With Huffman Encoding And Steganography Technique", International Journal of Network Security & Its Applications (IJNSA), Vol. 11, No.4.

[9] Masahito Kayasi, Shun Watanabe (2016), "Uniform Random Number Generation from Markov Chains-Non Asymptotic and Asymptotic Analysis", *IEEE Transactions on Information Theory,* Vol 62, No.4.

[10] Mervat Mikhail, Yasmine Abouleseoud, and Galal Elkobrosy (2017), "Two-Phase Image Encryption Scheme based on FFCT and Fractals", *Hindawi, Security and Communication Networks*, Article 1D 736751.

[11] Mohammed Ahmad Alia, Azman Bin Samsudin (2007), "New Key Exchange Protocol Based on Mandelbrot and Julia Fractal Sets", *IJCSNS International Journal of Computer Science and Network Security*, Vol.7, No.2.

[12] Nevart A.Minas, Faten.H.Mohammed Sediq, Adnan Ibrahem Salih, "Color Image Encryption Using Method of Fractal Based Key and Private XOR Key", Kirkuk University Journal/Scientific Studies(KUISS), Vol. 13,Iss.1,October,pp(104-117), ISSN 1992-0849.

[13] Nikolaos G. Bardis (2008), "Design and Development of a Secure Military Communication based on AES Prototype Crypto Algorithm and Advanced Key Management Scheme*", WSEAS Transactions on Information Science and Applications*.

[14] W. Puech, "Image Encryption and Compression for Medical Image Security", *IPTA'08: 1st International Workshops on Image Processing Theory, Tools and Applications.*

[15] Pu Li, Yuanyuan Sun, Xianglian Liu, XiaogangYi, Jianguo Zhang, Xiaomin Guo, Yanqiang, Guo, Yuncai Wang(2016), "Fully Photonics-based physical random bit generator", *Article in Optics Letter.*Vol 41. No.15,pp1-4.

[16] Pu Li, Yuanyuan Sun, Xianglian Liu, XiaogangYi, Jianguo Zhang, Xiaomin Guo, Yanqiang Guo, Yuncai Wang(2016), "Brownian motion properties of optoelectronic random bit generators based on laser chaos", *Opt Express*.

[17] Shafali Agarwal (2017), "Image Encryption Techniques using Fractal Function: A Review", *International Journal of Computer Science & Information Technology (IJCSIT),* Vol 9, No 2.

[18] Sunil.V.K.Gadam, Manohar Lal (2010), "Efficient Cancellable Biometric Key Generation Scheme for Cryptography", *International Journal of Network Security*, Vol. 11, No.2, pp-61-69.

[19] J. Szcepanski, E. Wajnryb, J.M. Amigo,Maria,V. Sanchez-Vives, M. Slater(2004), "Biometric Random Number Generators", Computers & Security 23, 77e84.

[20] K.Thamilzhchelvy, G.Geetha, "A Novel Approach to generate fractal images using Chaos theory", Indian Journal of Computer Science and Engineering (IJCSE).

[21] S. Vembu and S. Verdu (1995), "Generating random bits from an arbitrary source: Fundamental limits," IEEE Trans. Inf. Theory, vol. 41, no. 5, pp. 1322–1332.

[22] Yasutada Oohama (2011), "Performance Analysis of the Interval Algorithm for Random Number Generation Based on Number Systems", IEEE Transactions on Information Theory, Vol 57, No.3.

AUTHORS

Ms.T.Anusha completed her B.E degree in Computer Science and Engineering from Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, and her M.E degree in Computer Science and Engineering from Manonmaniam Sundaranar University, Tirunelveli. She is designated as Assistant Professor (Sel. Grade) in the Department of Computer Science and Engineering, PSG College of Technology, Coimbatore.

Dr.R.Venkatesan completed his B.E degree in Mechanical Engineering from Coimbatore Institute of Technology, M.E degree in Industrial Engineering from PSG College of Technology, M.S in Computer Science from the University of Michigan, and Ph.D. in Information and Communication Engineering from PSG College of Technology, Coimbatore. He is designated as a Professor in the Department of Computer Science and Engineering, PSG College of Technology, Coimbatore.