DNA and Bernoulli Random Number Generator Based Security Key Generation Algorithm

Sodhi, G. K. and Gaba, G. S.*

Discipline of Electronics and Communication Engineering, Lovely Professional University, Jalandhar 144411, India

ABSTRACT

Security is a major concern for the communication sector. The technique presented in this paper provides a novel security key generation mechanism. The proposed technique aims to generate a security key using the biological characteristics of the human body and the mathematically generated pseudo random sequences, thus producing different keys for different individuals. The final key is produced through the fusion of deoxyribonucleic acid (DNA) sequence of 1024 characters and Bernoulli Random Number Generator sequence of 256 bits. The performance of produced keys is evaluated using National Institute of Standards and Technology (NIST) tests and uniqueness is verified through avalanche test.

Keywords: Authentication, Bernoulli random number generator, biometrics, communication, confidentiality, DNA, integrity, security

INTRODUCTION

Security is critical in communication networks following the growth of the internet. Biometrics and integrity are two mechanisms which can improve security considerations. Biometrics ensures the identification and authentication of individuals by observing their personal unique features (Hao, Anderson, & Daugman, 2006). A practical system that integrates the iris biometrics into cryptographic applications can be found in Hao et al. 2006. A system that works using audio fingerprints has also been proposed by (Brown & Seberry, 2001; Chouakri, Bereksi, Ahmaidi, Fokapu, 2005; Covell & Baluja, 2007. Ktata, Ouni, Ellouze, 2009) studied the use of electrocardiogram (ECG) signals for enhanced security. Baluja, Covell, 2007 proposed to create personal signatures for authentication. Chen & Chandran, 2007) did a study on identification based on image processing is carried out by various researchers.

DNA has been used in many cryptographic algorithms to provide confidentiality (Chang, Kuo, Lo, & Wei, 2012). Bernoulli random

number generator (BRNG) is used to make the technique more efficient and effective. The BRNG works on a secret seed value which ensures the generation of a different sequence for each seed value provided. The work reported in this paper is based on the idea of unique and random attributes of DNA and Bernoulli random number generator sequence.

The approach to generate a security key is implemented through numeric coding of the DNA sequence and the Bernoulli random sequence generated with the secret seed value. The output of key generating algorithm is tested using NIST tests of randomness as well as the strict avalanche criterion, the results of which are formulated in Table 4 and Table 5 respectively. Results point out the applicability of the proposed approach in areas where security is a concern.

The paper is organized as follows; characteristics of DNA and Bernoulli random number generator are described in Section 2. In Section 3, proposed algorithm for the 256-bit key generation is presented where the DNA data is taken from the MIT-BIH database, followed by result and analysis in Section 4. Finally, conclusions and future work are reported in Section 5.

Characteristics of DNA and BRNG

Progress in the field of biotechnology has made the DNA sequencing more effective. DNA sequencing for various organisms has been done with higher accuracy (Goldberger, Amaral, Glass, Hausdorff, Ivanov, Mark, Mietus, Moody, Peng & Stanley, 2000). Investigating the biological relationships of different species is known as analysing the DNA sequence.

The internet provides various databases of DNA sequences which can be easily accessed from the World Wide Web (Ensembl Genome Browser, NCBI databases). The DNA is composed of two polymeric strands made of monomers that include a nitrogenous base (A-adenine, C-cytosine, G-guanine, and T-thymine), deoxyribose sugar and a phosphate group. According to most of the techniques a DNA sequence is taken as symbolic data that is composed of four characters A, C, G and T corresponding to the four types of nucleic acids present. The DNA samples from a genome are sequenced using a Genome Sequencer, the signals created in the sequencing process are then analysed by the software to generate millions of sequenced bases.

The backbone of each strand located on the surface of the DNA is formed by the sugar and phosphate groups, while the inside of the structure is made of bases. There exists a weak hydrogen bond between the complementary bases of each strand (i.e. between A and T and between C and G) giving rise to pairing of bases, this pairing holds the two strands together. DNA sequences are unique for every individual, including the identical twins. The pattern formed by a DNA sequence specifically represents an individual and its characteristics. Hence, there is no possibility of duplicity in the DNA sequences.

To strengthen the bond of security, a random sequence is generated by Bernoulli random number generator. This sequence is generated with a seed value that is secretly given by the user. BRNG computes n^{th} Bernoulli number for a given integer n. Bernoulli numbers are a sequence B_n of rational numbers defined by the Taylor expansion.

Bernoulli numbers have a prominent place in mathematics, for instance they appear in Taylor expansion of tangent and in Euler-Maclaurin formula. Equation (1) represents the BRNG method for generation of random sequences.

$$\sum_{n=0}^{\infty} \frac{B_n}{n!} x^n = \frac{x}{e^x - 1} \tag{1}$$

The fusion of BRNG and DNA sequence forms a very strong 256-bit key which is less susceptible to attacks and thus provides higher level of security.

The Key Generation Mechanism

The proposed key is generated by combining DNA sequence of an individual and the random sequence obtained through BRNG using a secret seed value. The method of the generation of these keys is devised into three subsequent subsections and the hierarchical structure is portrayed in Figure 1.

DNA sequence formulation

1) Obtaining a DNA sequence of 1024 characters from the DNA database [9]:
    The DNA sequence consists of 1024 characters forming base pairs 'agct'.

2) Obtaining the binary sequence from DNA characters:
    Each character of the DNA sequence is represented by 8 bit ASCII code. Hence, resulting in a DNA sequence of length 8192 bits.

3) Framing a DNA sequence of 256 bits:
    (i)   The DNA sequence is then divided into equal halves.
    (ii)   Apply exclusive-or operation on the obtained sequences.
    (iii)   The result is further divided into two equal parts and exclusive-or operation is applied again.
    The step (iii) is repeated till a sequence is obtained whose length is 256 bits. The whole procedure is summarized in the flow chart (Figure 1).
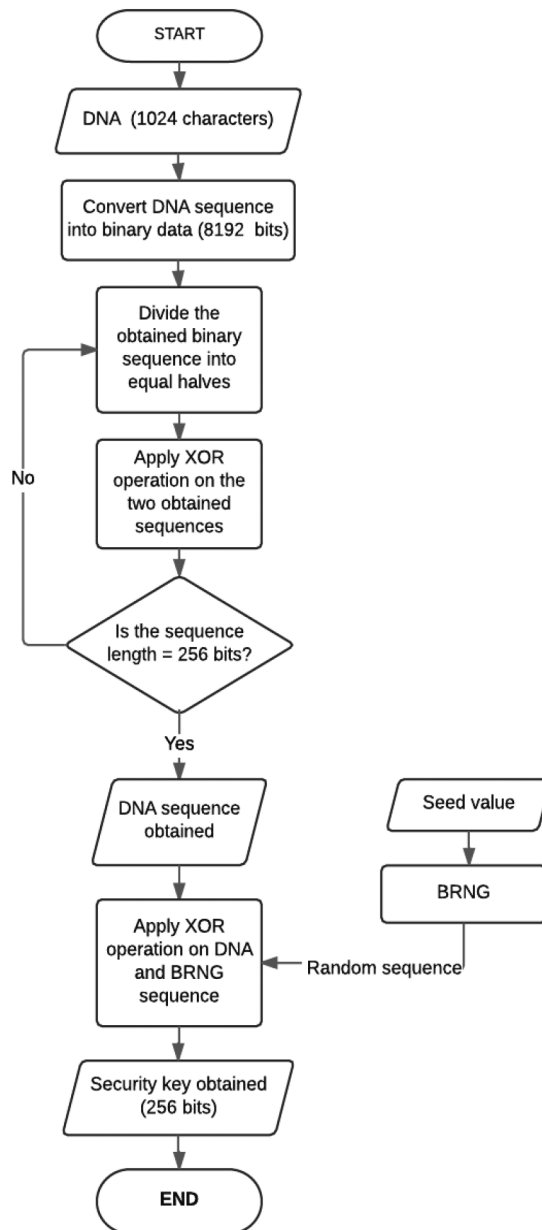
Figure 1. Key generation process

The conversion from 'agct' sequence of 1024 characters to 256-bit binary DNA sequence is given in Table 1.

Table 1
DNA sequence formulation

	DNA sequence (1024 characters)	DNA sequence (8192 bits)	DNA sequence (256 bits)
D_1	gcacaatcagaagcaggcggaggagacggcg gccttcgaggaggtcatgaaggacctgagcctg ac.. ....gcaca gaggcaaggcgtcagcaggcatcgcccaccct gtctccgctgtcacccatcactcaggctgtagcc atg.	011001110110001101100001 011000110110000101100001 0111010001100011............ ……………………………… ……………………………… ……………………………… ……………...…………....01 110100011000010110011101 100011011000110110000101 11010001100111	000001000000000000001010100 010111000000100000011000001 010100000100000000110000001 100000011000000110000010011 000000000000000000001001100 000000000000010000001100000 000000000110000001100000001 000000000100001001100000110 000000100000000000000010000 000000000010010000000000
D_2	ccacgcgtccgggcgagaagatggcgacttcg aacaatccgcggaaattcagcgagaagatcgc gct………………………………… ………………………………..... ....ggcgtcagcc ccctgtccctgagcacagaggcaaggcgtcag caggcatcgcccgccctgtccccgctgtcaccc at.	011000110110001101100001 011000110110011101100011 0110011101110100................ ……………………………… ……………………………… ……………………………… …………………………..,……0 111010001100011011000010 110001011000110110001101 10000101110100	000000100000000100000001000 000010000001100001011100000 01 000001011100000100000000 000000000000000000000000110 000000100000011000001010100 010011000000100000000000000 000000000010000100110000010 100000000000000001000000110 000011000000001000000000000 010111000001000000000100
D_3	agcccttaggggaagagtcctgctctggctgttg atgctccagctccagaaatcccagtacctgcaac tg………………………………… ………………………………… ………………………………… ………………………………… …………..…………tctggagcagcagctg ccctacgccttcttcacccaggcgggctcccagc agccaccgccgcagccccagcccccgccg	011000010110011101100011 011000110110001101110100 0111010001100001............ ……………………………… ……………………………… …………..………………… …………..…………..…….1 000110110001101100011011 000110110011101100011011 0001101100111	000101010000000000000001000 010111000001000001011100000 000000000010000001000000001 100000000000000010000010101 000000100000010000000000000 000000000101010000000100000 001000000000000000100000000 000001011100000010000100 11 000100110000011000000000000 010111000100110001001 1

Random sequence generation through BRNG. Three sequences are generated using three different seed values. A seed value is a secret input given by the user at the beginning of generator operation. The obtained random sequences along with the corresponding seed value are summarized as in Table 2.

Table 2
BRNG sequence

	Seed value	Bernoulli random sequence (256 bits)
B_1	1231	1011001100000011010110100110000101010010000101110111101110011101000 0000110000011010101101000101011011101010001101110101010101100000001 1001110110100000110110011001011110111001010001000101100100000000101 11000011110110101110111100000110111100001100100010010110
B_2	101355	0001011010010001010100011001011001110000001101011111100111101010110 0010010110010100001110110111001110000100110011001101101110100101110 1101110100011000001101111100111111111001101000001111100110010010011000 0100010001100110111001001001001010101101011010100001
B_3	2114	0111010100001101001100000101010100000001000100001010101010101000001 1010111101010101001101001101001010000011000100010101110000000100011100 01110001000011001001011111000101110001010011010111011111101011000011 00011000101110110010110100100100001111011110001101011

*B_1, B_2, B_3: Three BRNG sequences

Fusion of DNA sequence and BRNG sequence. To enhance the strength of security and randomness in the key, exclusive-or logic is applied between each DNA and BRNG sequence. The resultant key has length of 256 bits and is strong enough to survive critical attacks on networks. The process is repeated for two other DNA and BRNG sequences. Therefore, three 256 bit keys are obtained as shown in Table 3.

Table 3
Security keys

KEY$_1$	000100101001000101000100100000010111001000110011111011001110111011001000101111101000101101100010010101101001100110011011010100101011100110111110001111000110111110010011111111101001001111101110000001011001000010000100110111001000001001010101111000010100001
KEY$_2$	101100010000000101011000011000101010101111000111001011100111011100110001010011000011010101011010001010101101000101000010111001101000011000100000011100101000001101100110010110101100001100011101011001000000001101110110111100001011100111000001101010110011011001001111
KEY$_3$	011000000000110100110010010000100000001010000011110101010101010101000110001111011001010010011010010011110010000101011000000010111110000000010001110011001001000000111010010101110001011100000100110101011100100010101110011100000000001000110001110100100100100001010111101111000

The final key is a combination of true random source (DNA) and pseudo random source (BRNG). The keys obtained have unique and random characteristics and thus can be useful in maintaining security.

RESULTS AND DISCUSSION

A security key is said to be efficient if it is random and unique. The National Institute of Standards and Technology (NIST) tests discuss some aspects of selecting and testing random number generators. The outputs of these generators may be used in most of the security applications for the generation of security keys. The generators that are to be used for security applications should meet stronger requirements than those to be used for other applications. To be precise, their outputs need to be unpredictable for unknown inputs. These tests may be useful for determining if a generator is suitable for a particular security application. The randomness of a key is evaluated on the bases of its P-value, which must be greater than 0.01 for a random sequence.

NIST Tests

The keys obtained are evaluated on the basis of seven NIST randomness verification tests and the P-value is calculated for each test with respect to the security key. An overview of the NIST tests used for evaluating the sequences is given as:

Runs test. The purpose of applying this test is to calculate the number of runs in an entire sequence, where run specifies the number of uninterrupted sequence of identical bits (Rukhin et al., 2010). The results in Table 4 clearly depict that the proposed algorithm has higher rate of interruptions.

Frequency Test. Frequency test is used for determining the proportion of number of ones and zeros in an entire sequence. It is used to check the closeness between the number of ones and number of zeros. A sequence is said to be random if the proportion of zeros and ones are close to each other (Rukhin et al., 2010). The results in Table 4 clearly depict that the proposed algorithm produces better proximity between the count of ones and zeros as compared to the traditional techniques.

Approximate Entropy Test. This test deals with the frequency of all the overlapping bit patterns across the entire sequence. The aim of this test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths with the expected result for a random sequence.

Discrete Fourier Transform Test (DFT). The DFT test is used to find the peak heights in the Discrete Fourier Transform of a sequence. The aim of this test is to detect periodic features (i.e. repetitive patterns) in the tested sequence that indicates a deviation from the assumed randomness. The goal is to detect if the number of peaks exceeding the 95 % threshold are significantly different than 5%.

Binary Derivative Test. The test is performed using exclusive-or operation between successive bits until only one bit is left. Afterwards, the ratio of number of ones to the length of entire sequence in each case is calculated. Finally, the average of the ratio of all the sequences is observed, and where the value lies near to 0.5, then the sequence is considered as a random sequence (Rukhin et al., 2010). The results in Table 4 show that the output of proposed algorithm is random.

Maurer's "Universal Statistical" Test. The aim of this test is to detect if a sequence can be significantly compressed without any loss of information. The number of bits between matching patterns are calculated. A sequence that is significantly compressible is considered to be non-random. This test is also known as Universal test (Rukhin et al., 2010).

Random Excursion Variant Test. The test focuses on the total number of times a particular state occurs in a cumulative sum random walk. It detects the deviations from the expected number of visits. The P-value specifies if the sequence is random or not. This test considers successive sums of the binary bits as a one-dimensional random walk (Rukhin et al., 2010).

$$P_{value} = erfc \times \frac{(|\xi(x) - j|)}{\sqrt{(2 \times j \times ((4 \times |x|) - 2))}}$$

(1)

Where,

      erfc: the error function
      ξ : the total number of times the state *x* occurs
      x : the state occurred
      j : the total number of cycles

The P-value must be greater than 0.01 for a sequence to be random (Rukhin et al., 2010).

It is observed that the proposed technique produces better results in terms of randomness of the keys. The efficiency of the proposed technique is evaluated by comparing it with other traditional biometric techniques used for authentication and security key generation. The tests have been performed on KEY_1 and results are tabulated in Table 4.

Table 4
Comparison and analysis

S. No	Input Source of random number generator	Key Length (bits)	Runs Test	Frequency Test	Approximate Entropy Test	DFT Test	Binary Derivative Test	Maurer's Test	Random Excursion Variant Test
			P-value	P-value	P-value	P-value	P-value	P-value	
1.	ECG	128	0.1262	0.2487	0.5468	0.0294	0.5039	0.9428	Random
2.	Image	256	0.0809	0.8026	0.9759	0.4220	0.4887	0.9780	Random
3.	Iris sequence	128	0.1254	0.3768	0.9409	0.3304	0.5021	0.9062	Random
4.	Finger print	128	0.3345	0.3041	0.3345	0.7597	-	0.2757	Random
5.	**DNA & BRNG**	256	0.0438	0.9005	**0.9340**	**0.8185**	0.5090	0.9920	Random

It can be observed that the P-value obtained for the keys generated for all the seven tests is significantly greater than 0.01. Thus, it can be concluded that the keys generated are random in nature and hence fulfil the basic criteria required for security keys.

Avalanche test has also been performed on the keys obtained. The purpose of this test is to check the avalanche effect, which is a desirable property of the security keys. Wherein if the input is changed slightly the output changes significantly. It gives the percentage of bits flipped with a change in the input. It is a desirable property of security keys.

The test is performed on three sets of DNA and BRNG sequences:

Case 1: In the initial set, two security keys are generated through two DNA sequences while keeping the same BRNG sequence.

Case 2: The second set involves generation of two security keys through the same DNA sequence and two BRNG sequences.

Case 3: In the third set, two security keys are generated through two DNA and BRNG sequences.

Further, the avalanche effect is calculated for each of the three sets and results are tabulated in Table 5, Table 6 and Table 7 respectively, this is done to know the amount of randomness the proposed technique produces on changing the input.

The avalanche effect can be calculated using the formula given in equation (3).

$$Avalanche\ effect = \frac{No.of\ bits\ flipped\ in\ the\ sequence}{Total\ no.of\ bits\ in\ the\ sequence} \times 100 \tag{3}$$

The result of avalanche effect on changing the inputs is summarized in tabular form, where D_1, D_2, D_3 represent the DNA sequences as taken from Table 1 and B_1, B_2 B_3 represent the BRNG sequences characterized by the seed values, as taken from Table 2.

Table 5
Avalanche test analysis: Case 1

DNA Sequences (D_n)	Seed Value	Bernoulli Random Sequence (B_n)	Key Generated K= D_n xor B_n	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
D_1			1011011100000011010011110111011001 0100000001000101101110100110010010000 0011110000101010100001000110011000 1100100011011101010101111111000001 1001110100100001010110011001011000 0111000110001010101100110000100001 1110011111101111011101111000011101 1110000111011110010110		
	1231	B_1	1011000100000001010110000110001100 1010100000000000111111111000101000 0001011000001101010110100010101100 1101101000100111011001011100100010 0101011101001000001101100110010111 0011110110000101110110010000000000 0011100111111010110111010110000011 0110111101100000010010010	58	22.65 %
D_2					

* Refer Table 1 for D_1, D_2, D_3 and Table 2 for B_1, B_2, B_3
** Different DNA sequences - Same BRNG sequence

Table 6
Avalanche test analysis: Case 2

DNA Sequence (D_n)	Seed Value	Bernoulli Random Sequence (B_n)	Key Generated K= D_n xor B_n	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche effect
D₁	1231	B₁	10110111000000110100111101110110010100000001000101101111010011001000001111000010101010000100011001100111001000110111010101011111100000110011010010000101011001100101100011100011000101010110011000010011110011111101111011110111100001110111100001110111100101100	124	48.43 %
	101355	B₂	00010010100100010100010010000001011100100011001111101100111011101110010000101111101000101101100010010101101001100110011011010100100101011001101111100011110001101111100100111111111101001001111101110000001011001000010000100110111001000001001010101111000010100001		

* Refer Table 1 for D_1, D_2, D_3 and Table 2 for B_1, B_2, B_3
** Same DNA sequences - Different BRNG sequences

Table 7
Avalanche test analysis: Case 3

DNA Sequence (D_n)	Seed Value	Bernoulli Random Sequence (B_n)	Key Generated K= D_n xor B_n	Avalanche Result of Key (K)	
				No. of bits flipped	Avalanche Effect
D₁	1231	B₁	10110111000000110100111101110110010100000001000101101110100110010000011110000101010100001000110011001110010001101110101010101111110000011001110100100001010110011001011000111000110001010101100110000100011110011111101111011101111000011101111000011101111001011011	58	22.65 %
D₂	101355	B₂	00010100100100110101001110010100011101100010001011111101111111101110011001011001010000111011011100111110010011011001011101011110100111111011111000110000011011111001101111010101010011011111100110010000011001000010010100110101001001001000001001101111010101010		

*Refer Table 1 for D_1, D_2, D_3 and Table 2 for B_1, B_2, B_3
**Different DNA sequences – Different BRNG sequences

The avalanche test results indicate that a slight change in the inputs leads to a significant change in the output. The higher the percentage value of avalanche effect, the more is the uniqueness of the key. As the seed value may vary according to the user's choice and the DNA being unique to an individual, it can be assumed that the security system built has higher efficiency and is less susceptible to attacks.

CONCLUSION

In this work, a novel algorithm is proposed to generate biometric security keys for authentication using an individual's DNA sequence and BRNG sequence. DNA being the most unique characteristic of an individual when collaborated with the Bernoulli seed is more efficient. The proposed key finds its application in various areas where maintaining the sender's confidentiality as well as retaining the original message are important. Most of the security system researches have been carried out on fingerprint, facial, iris and voice recognition; while those focused on the DNA are rare. A system that computes a 256-bit biometric security key through the fusion of DNA and BRNG is presented and analysed using NIST Tests. Results indicate this method is superior to more traditional techniques.

REFERENCES

Baluja, S., & Covell, M. (2007, April). Audio fingerprinting: Combining computer vision & data stream processing. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, ICASSP 2007* (Vol. 2, pp. II-213). IEEE.

Brown, L., & Seberry, J. (1990). On the design of permutation P in DES type cryptosystems. In G. Brassard (Ed.), *Advances in Cryptology—EUROCRYPT'89* (pp. 696-705). Germany: Springer Berlin/Heidelberg.

Chang, H. T., Kuo, C. J., Lo, N. W., & Lv, W. Z. (2012). DNA sequence representation and comparison based on quaternion number system. *DNA Sequence*, *3*(11), 39-46.

Chen, B., & Chandran, V. (2007, December). Biometric based cryptographic key generation from faces. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications* (pp. 394-401). IEEE.

Chouakri, S. A., Bereksi-Reguig, F., Ahmaldi, S., & Fokapu, O. (2005, September). Wavelet denoising of the electrocardiogram signal based on the corrupted noise estimation. *Computers in Cardiology, 2005* (pp. 1021-1024). IEEE.

Covell, M., & Baluja, S. (2007, April). Known-audio detection using waveprint: spectrogram fingerprinting by wavelet hashing. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, ICASSP 2007* (Vol. 1, pp. I-237). IEEE.

Ensembl. (n.d.). *Ensembl Genome Browser*. Retrieved from *http://www.ensembl.org/index.html*

Garcia-Baleon, H. A., Alarcon-Aquino, V., & Starostenko, O. (2009, August). A wavelet-based 128-bit key generator using electrocardiogram signals. *52nd IEEE International Midwest Symposium on Circuits and Systems, 2009, MWSCAS'09* (pp. 644-647). IEEE.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., ... & Stanley, H. E. (2000). Physio bank, physio toolkit, and physio net. *Circulation*, *101*(23), e215-e220.

Hao, F., Anderson, R., & Daugman, J. (2006). Combining crypto with biometrics effectively. *IEEE transactions on computers*, *55*(9), 1081-1088.

Hedayatpour, S., & Chuprat, S. (2011, September). Hash functions-based random number generator with image data source. *IEEE Conference on Open Systems (ICOS), 2011* (pp. 69-73). IEEE.

Khokher, R., & Singh, R. C. (2015, May). Generation of security key using ECG signal. *International Conference on Computing, Communication and Automation (ICCCA), 2015* (pp. 895-900). IEEE.

Ktata, S., Ouni, K., & Ellouze, N. (2009). A novel compression algorithm for electrocardiogram signals based on wavelet transform and SPIHT. *International Journal of Signal Processing*, *5*(4), 32-37.

NCBI. (n.d.). *National Center for Biotechnology Information*. Retrieved from *http://www.ncbi.nlm.nih.gov/Entrez*

NCBI. (n.d.). *National Center for Biotechnology Information*. Retrieved from http://www.ncbi.nlm.nih.gov/nuccore/3327045?report=genbank

NCBI. (n.d.). *National Center for Biotechnology Information*. Retrieved From http://www.ncbi.nlm.nih.gov/nuccore/20380066?report=genbank

NCBI. (n.d.). *National Center for Biotechnology Information*. Retrieved from http://www.ncbi.nlm.nih.gov/nuccore/33874586?report=genbank

Rukhin, A., Soto, J., Nechvatal, J., Barker, E., Leigh, S., Levenson, M., ... & Smid, M. (2010). *Statistical test suite for random and pseudorandom number generators for cryptographic applications*. Gaithersburg: NIST Special Publication.

Wei, W., & Jun, Z. (2013, November). Image encryption algorithm based on the key extracted from iris characteristics. In *IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI), 2013* (pp. 169-172). IEEE.

Ying, L., Shu, W., Jing, Y., & Xiao, L. (2010, December). Design of a random number generator from fingerprint. In *International Conference on Computational and Information Sciences (ICCIS), 2010* (pp. 278-280). IEEE.