Biosc.Biotech.Res.Comm. Special Issue Vol 13 No 13 (2020) Pp-67-73

BBRC
Bioscience Biotechnology Research Communications

A Novel Symmetric Key Generation Technique for Securing Images in the Cloud: A Comparative Study

Pallavi Kulkarni[1]*, Rajashri Khanai[2] and Gururaj Bindagi[3]
[1]*Department of Electronics and Communication Engineering, KLE Dr. M.S. Sheshgiri College of Engineering and Technology, Belgaum, India*
[2]*Department of Electronics and Communication Engineering, KLE Dr. M.S. Sheshgiri College of Engineering and Technology, Belgaum, India*
[3]*Solution Architect, Bengaluru, India*

ABSTRACT

Cloud computing has revolutionized the world of computing. Cloud storage enables consumers to remotely store their data and enjoy the flexible, pay-as-you-go on-demand high quality cloud applications. Cloud computing gained popularity as users need not take the burden of hardware and software management. So the new trend is to outsource the multimedia information to cloud. Today, the rate at which the image data set produced is exponential in nature. Hosting of data including multimedia by third party gives rise to many questions related to the privacy and security since the cloud environment makes use of shared resources over internet. To address the above mentioned problem, we propose a new method to secure the images in the cloud environment. We are using Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithm for encryption and the symmetric key used by these algorithms is generated by Elliptic Curve Cryptography Diffie Hellman (ECCDH) /Neural network. The actual key used for encryption is different from the key shared between the two parties. If the key1 which is shared between the two parties is compromised, the actual key used for encryption will not be known to the unauthorized person. Hence the greater level of security is provided by the proposed technique. The comparative analysis of the existing system and proposed system is carried out by considering Performance and Security parameters.

KEY WORDS: AES, CLOUD COMPUTING, DES, ECC, IMAGE, TPM.

INTRODUCTION

Many new technologies have evolved over the time but cloud computing is the most promising one as it has changed the landscape of Information Technology. Cloud computing is able to address number of issues like software licensing/maintenance, platform upgradation etc. (Zissis and Lekkas, 2012). Although it has many advantages, it is operated by third party who is usually outside the data owner's trusted domain. Security is the topmost concern, out of many challenges the outsourcing of data/ multimedia is facing. In order to become truly successful, need of the hour is to address the underlying security issues. The recent way of exchanging the information is using multimedia data(Wang et al., 2014).

Images/photos are predominantly used as source of information. Example for Image dataset is medical images (Prabu, S et al. 2019), in which the diagnostic results for various patients are confidential. We need to embed the security feature from the very beginning, so that we can better protect owners' data privacy without sacrificing

ARTICLE INFORMATION

*Corresponding Author: pallavik15@gmail.com
Received 5th Oct 2020 Accepted after revision 29th Dec 2020
Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal

NAAS Journal Score 2020 (4.31)

Online Contents Available at: http//www.bbrc.in/
Doi: http://dx.doi.org/10.21786/bbrc/13.13/10

the availability and ease of access (Parameshachari, B. D et al. 2019). Cryptography plays an important role when we want to transfer sensitive/confidential information to third party, that's cloud (Kulkarni and Khanai, 2015). We have implemented Elliptic-Curve Diffie–Hellman (ECCDH) key agreement protocol. Basically it is Diffie–Hellman key agreement protocol with elliptic curve mathematics.

Elliptic Curve Discrete Logarithmic Problem involves the trap door function in which reversing a function is impossible. The protocol allows each party to have elliptic curve private-public key pair, using which the secret key is generated. One more technique we have implemented here to generate key is use of Tree Parity Machine. The concept of Neural Networks draws inspiration from the human nervous system. The neural networks consist of different layers that are analogous to the neurons of the human system. The first layer takes the input provided and transmits it to the succeeding layers in a manner similar to how neurons communicate with each other using synapses. The number of layers in a system determines its complexity.

Each synapse has a weight as a parameter that is included in the calculation of the input to the succeeding layer. The network has a learning process which aims to optimize the outputs by updating of weights for each layer. This method of learning is termed as a gradient descent mechanism. The process of encryption (Parameshachari, B. D et al. 2020) and cryptanalysis that uses stochastic algorithms in combination with neural networks gives rise to a branch of cryptography called Neural Cryptography. The Neural key exchange protocol is an important part of this domain, and is a protocol that allows the secure transfer of a shared key between the two parties. The basis for this lies in the usage and synchronization of two Tree parity machine (TPM) (Chourasia et al., 2019). In this paper we investigate the work carried out till now to understand the depth of the challenge and propose a novel method to secure images in the cloud.

MATERIAL AND METHODS

Chourasia Smruti, Bharadwaj Hrishikesh C,Queenie Das et al. present a novel vectorized TPM (vTPM) in order to develop a key. It also provides a system to detect any unwanted listeners, as one of the weakness of the TPM algorithm is Man in the middle attacks. Further this key is utilized for authentication between a sender and a receiver. The authentication is carried out by means of H-MAC with the SHA-512 hashing mechanism. Finally, a comparison is drawn out between the serial and vector implementation of the Tree Parity Machine. In this paper, authors Dr. Mahajan Prerna & Sachdeva Abhishek provide comparative analysis of three most popularly used encryption algorithms AES, DES and RSA. The analysis is performed on various factors as Key Size, Block Size, Ciphering & Deciphering key, Scalability, Power Consumption, No. of Rounds and Stimulation Speed etc. This helps us to understand the behavior of each algorithm.

Encryption is used to protect the sensitive data from the opponent. But in case of symmetric ciphers both sender and receiver need to have the same key. As Diffie Hellman key exchange protocol is prone to man in the middle attack, authors Dr. Singh Ajit, Nandal Aarti is proposing to use secret key generation and sharing by synchronization of Tree Parity Machine. This key can be further used as secret key for AES algorithm. Gupta Mayank, Gupta Manu & Deshmukh Maroti propose a new key generation technique using neural network. To generate secret key using neural networks many techniques are available like Tree Parity Machine (TPM) and many others. In TPM there are some flaws like less randomness, less time efficient. There are already three rules available i.e. Hebbian Rule, Anti Hebbian Rule and Random Walk, with same problems. So to overcome these issues, this paper proposes a new approach based on the same concept(TPM, as Tree-structured Neural Network's execution time is comparatively less than that of the other Neural Networks) which generate random and time-efficient secret key.

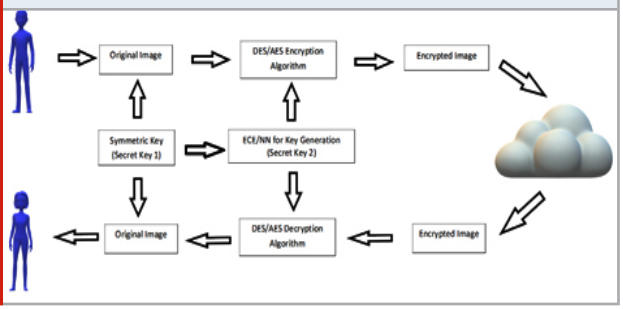
Kumar Mohit, Chahal Anju, 2014 in paper Effect of Encryption Technique and Size of Image on Correlation Coefficient in Encrypted Image tried to establish the relation between image encryption methods, correlation coefficient and size of the image. Security of the confidential images is ensured by several image encryption algorithms. In this paper, authors Singh Laiphrakpam Dolendro, Singh Khumanthem Manglem implement the Elliptic Curve cryptography to encrypt, decrypt and digitally sign the cipher image to provide authenticity and integrity. The histogram analysis of different size images is carried out to understand the strength of the proposed technique. Key sensitivity, Correlation coefficient and entropy analysis are the other parameters considered for the analysis.

The basic idea in the paper published by Wang Honggang, Wu Shaoen et al. is to protect the multimedia information stored in the mobile cloud by using secure sharing and scalable watermarking techniques. Here multimedia information like images are divided in to multiple pieces and stored in different clouds so that it is impossible to get information from the single cloud. Reed- Solomon code is used minimize the transmission errors. In this paper authors Zissis Dimitrios, Lekkas Dimitrios present a feasible solution to eliminate potential threats posed by cloud. This is achieved by evaluating cloud security requirements. Here Trusted Third Party is used with security characteristics. Authentication, integrity and confidentiality of involved data and communications are achieved by using Public key Infrastructure.

Proposed Model: A new security model has been proposed to enhance the security provided by traditional encryption approaches using AES, DES (Mahajan and Sachdeva, 2013). Symmetric Key algorithms rely on the secret key to provide the effective encryption. At the same time it is the weakest link in the security of the Symmetric key encryption technique. As a first step, it

was decided to overcome this weak link by introducing a key generated by ECC, which will eventually be compared with the NN based key generation. Approach is depicted pictorially as below:

Figure 1: Methodology



Model is derived by applying the key generation concepts by using ECC and Neural Network algorithm. The generated key is applied to AES and/or DES algorithms. Considering the better performance and security advantages offered by Elliptic Curve and Neural Network cryptography, we decided to use it for key generation and AES and DES for encryption of images. The proposed model helps us reap benefit of both. Comprehensive analysis and results are discussed in subsequent sections.

Elliptic Curve Cryptography: Neal Koblitz and Victor S. Miller developed ECC in the year 1985. ECC is a public key cryptography. Difficulty in solving an Elliptic Curve Discrete Logarithmic problem makes ECC a very good choice for encryption/decryption (Singh and Singh, 2015). Strength of ECC depends on the hardness of the discrete logarithm problem. Let X and Y be two points on an elliptic curve such that kX =Y, where k is a scalar. Given X and Y, it is hard to compute k. k is the discrete logarithm of Y to the base X (Nagaraj and Raju, 2015).

To make operations more efficient and accurate, the curve is defined over two finite fields
1. Prime field Fp and
2. Binary field F2m

The field is chosen with finitely large number of points suited for cryptographic operations.
Elliptic Curve on Prime field Fp is given by equation:

$$y^2 \bmod p = x^3 + ax + b \bmod p$$
$$\text{Where } 4a^3 + 27b^2 \bmod p \neq 0.$$

Elements of finite fields are integers between 0 and p-1. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.

Elliptic Curve on Binary field F2m is given by equation:

$$y^2 + xy = x^3 + ax^2 + b,$$
$$\text{Where } b \neq 0$$

ECCDH – Elliptic Curve Diffie-Hellman
Let A & B be the two parties generating the same shared secret key. Let dA & dB be private key, randomly selected integer less than n, where n is the order of the curve (an elliptic curve domain parameter).

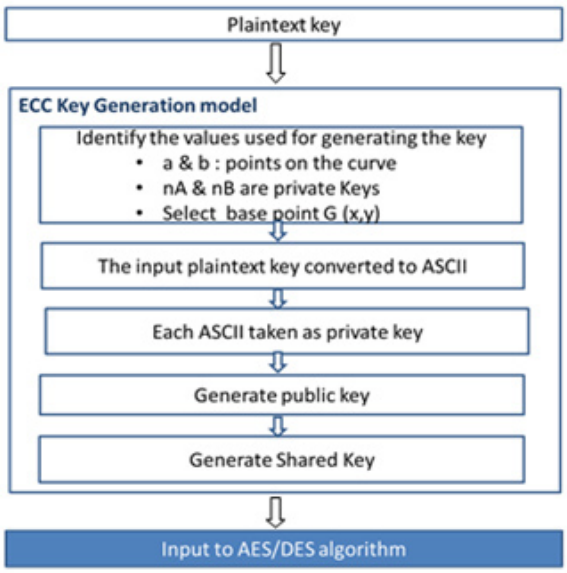
A (QA, dA) – Public, Private Key pair
Public key QA= dA * G
B (QB, dB) – Public, Private Key pair
Public key QB= dB * G

Where G is the generator point which is an elliptic curve domain parameter. Steps to generate shared secret key is as below:

1. The end A computes K = dA * QB
2. The end B computes L = dB * QA
3. Since dAQB = dAdBG = dBdAG = dBQA.
Therefore K = L

4. Hence the shared secret is K

Figure 2: Generating Shared key using ECCDH



We are using the ECCDH to generate the shared key. As depicted in [Fig. 2], plaintext key is converted in to its ASCII equivalent and each ASCII value is used as a private key. Public key is generated using this private key and generator point G. The shared key is generated by multiplying private key of A and public key of B. In turn this is used as key for DES and AES algorithm.

Neural Network: A simple neural network is developed by Rosenblatt in 1968.A simple neural network consist of an input vector X, a hidden layer Sigma S, weight coefficients W between the input vector and the hidden layer and an activation function.

Tree Parity Machine: TPM is a special type of multilayer feed-forward neural network. TPM relies on a neural network with a single hidden layer. The tree comprises

of input (N) neurons and hidden (K) neurons. This results the TPM to have K*N number of weights from the neural network. The Input vector Xij, ranges from is -1 to +1 as given (1). We restrict the weights between the bounds {-L... -2, -1, 0, 1, 2... L}, where L is a parameter of the TPM (Singh and Nandal,2013; Pal and Mishra,2019). Output value of each hidden neuron is shown in (3). The activation function is denoted by Signum as shown in (4).

Input Vector $X_{ij} = \{-1, 0, 1\}$ (1)

Weights $W_{ij} = \{-L,.....,0,.....,L\}$ (2)

$$\sigma i = sgn \sum_{j=1}^{n} Wij * Xij \quad (3)$$

$$\text{sgn}(x) = \begin{cases} -1 \text{ if } x < 0 \\ 0 \text{ if } x = 0 \\ 1 \text{ if } x > 0 \end{cases} \quad (4)$$

Where,

The final output (τ) is given by equation (5), which is product of output of each hidden neuron.

$$\tau = \prod_{k}^{i=1} \sigma i \quad (5)$$

Suppose that there are two machines Alice and Bob, which require the generation of a common key. The two machines, Alice and Bob are initialized with random weights and provided the same parameters (K, N, and L). Initially, the weights are different due to the random initialization. In order to exchange the key between Alice and Bob, we require the updation of weights in such a manner so as to synchronize the two machines, thereby, having the same weights for Alice and Bob. We use the following mentioned algorithm to update the weights in the TPM, each of them varying slightly to the other.

Learning Mechanism of Hebbian: The two neurons, pre-synaptic and post-synaptic are connected to each other by the synaptic weight. If they are active together, then they have similar excitation. Synaptic strengthening takes place due to positive correlation between pre-synaptic and post-synaptic neurons. Synaptic weakening happens due to negative correlation. According to Hebbian learning rule, positive correlation increases the strength between the synapses (Gupta et al., 2020). The change in weights wkj(t) associated with a neuron is shown in (6).

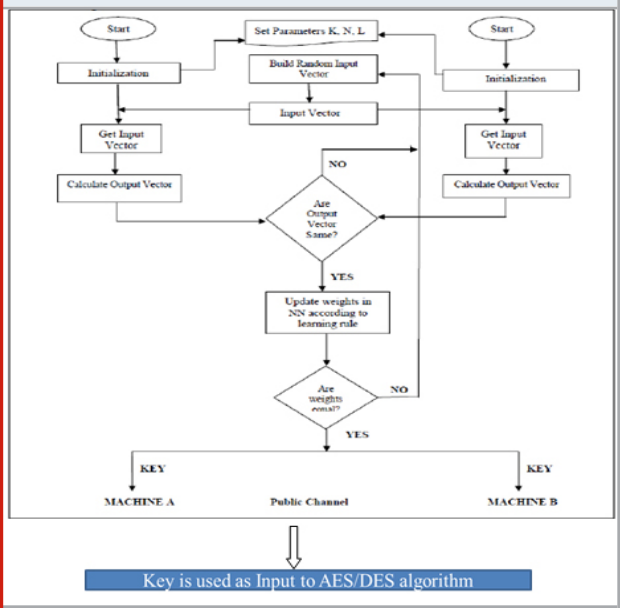
$$w_{kj}(t) = \eta \times y_k \times x_j \quad (6)$$

Our weights keep on increasing and we will never get a saturation point. To solve the problem the weights are confined within (–L, ..., 0, ..., +L). Now, we have two threshold points at which the weights can saturate. Thus, we can get the synchronization point (the point at which the two machines will have equal weights) and this weight is taken as key.

Flow Chart For Key Generation

Key Generation: Steps involved in the secret key generation based Tree Parity Machine for the flow chart shown in [Fig. 3] is:

Figure 3: Flow chart for key generation



1. Set the parameters k, n and l.
2. Initialize the network weights randomly.
3. Repeat the steps 4 through 8 till we achieve synchronization
4. The Key distribution Centre is third party responsible for generating the Inputs.
5. Calculate the inputs of the hidden units.
6. Find out the output vector using equation (3) and (4).
7. Use the Hebbian learning rule to update the weights if the output vectors of both the machines are same.
8. Check if synaptic weights are same for both the networks when synchronization occurs and this final weight is considered as secret key.

RESULTS AND DISCUSSION

In this segment, we define the parameters against which we are analyzing the results of existing and proposed system. The above said work is carried out on MATLAB platform on a virtual environment with 4GB memory. The gray scale images Baboon, Barbara and Lena are used as inputs.

Performance Analysis Parameters: The Performance parameters help us compare the different algorithms in terms of Execution time, CPU utilization and Throughput. Execution Time, measured in seconds is the measure CPU time required to execute encryption algorithm. CPU Utilization shows Percentage processor utilization. Throughput is number of bytes encrypted per second.

Security Analysis Parameters: To understand the strength

and weakness of the algorithms, the analysis is done against the security parameters.

Entropy: is a measure of randomness or uncertainty present in the data. Higher value of entropy is desired.

Table 2. Comparison of Security Parameters: A quantitative analysis of existing and proposed scheme (DES, DES with ECC & NN for key generation).

	Baboon			Barbara			Lena		
Parameter	DES	DES With ECC Key	DES With NN Key	DES	DES With ECC Key	DES With NN Key	DES	DES With ECC Key	DES With NN Key
Execution time (Sec)	10.81	11.4	10.59	10.84	11.31	10.37	10.78	11.35	10.39
CPU Usage(%)	0.10014	0.09937	0.1016	0.10104	0.09969	0.10004	0.1	0.10003	0.10002
File Size (Bytes)	1.11	1.11	1.11	1	1	1	0.99	0.99	0.99
Throughput(Bytes/Sec)	105.24	99.82	107.39	94.02	90.06	98.27	94.27	89.51	97.83

Table 2. Comparison of Security Parameters: A quantitative analysis of existing and proposed scheme (DES, DES with ECC & NN for key generation)

	Baboon			Barbara			Lena		
Parameter	DES	DES With ECC Key	DES With NN Key	DES	DES With ECC Key	DES With NN Key	DES	DES With ECC Key	DES With NN Key
Avg.Entropy	7.32	7.32	7.31	7.48	7.47	7.47	7.5	7.5	7.5
No. of bits changed	4068	4113	4079	4006	4059	4021	4038	4082	4108
Sensitivity	49.658	50.208	49.792	48.901	49.548	49.084	49.292	49.829	50.146
Hitcollision	0.488281	0.78125	0.126953	0.19531	0.292969	0.390625	0.58594	0.195313	0.215938
Correlation Coeff.	0.04565	0.040721	0.08521	0.0374	0.006937	0.023165	0.02749	0.001172	-0.00852

Table 3. Comparison of Performance Parameters: A quantitative analysis of existing and proposed scheme (AES, AES with ECC & NN for key generation)

	Baboon			Barbara			Lena		
Parameter	AES	AES With ECC Key	AES With NN Key	AES	AES With ECC Key	AES With NN Key	AES	AES With ECC Key	AES With NN Key
Execution time (Sec)	8.84	9.45	9.04	8.94	9.45	8.85	8.85	9.51	8.85
CPU Usage(%)	0.1004	0.10035	0.1018	0.10126	0.10054	0.0999	0.10026	0.10266	0.10009
File Size (Bytes)	1.11	1.11	1.11	1	1	1	0.99	0.99	0.99
Throughput(Bytes/Sec)	128.68	120.38	126.84	114.04	107.83	115.08	114.78	106.84	114.79

Table 4. Comparison of Security Parameters: A quantitative analysis of existing and proposed scheme (AES, AES with ECC & NN for key generation)

	Baboon			Barbara			Lena		
Parameter	AES	AES With ECC Key	AES With NN Key	AES	AES With ECC Key	AES With NN Key	AES	AES With ECC Key	AES With NN Key
Avg.Entropy	7.33	7.32	7.32	7.48	7.48	7.48	7.49	7.48	7.49
No. of bits changed	4041	4184	4243	4115	4040	4041	4148	4143	4164
Sensitivity	49.329	51.074	51.353	50.232	49.316	49.329	50.635	50.574	50.83
Hit collision	0.683594	0.390625	0.390625	0.585938	0.195313	0.183594	0.683594	0.292969	0.097656
Correlation Coeff.	-0.050831	-0.006866	0.037145	-0.022639	-0.010964	0.004167	-0.023875	-0.001868	-0.01596

Figure 4: No. of Bits changed in DES, DES with ECC & NN key for different images

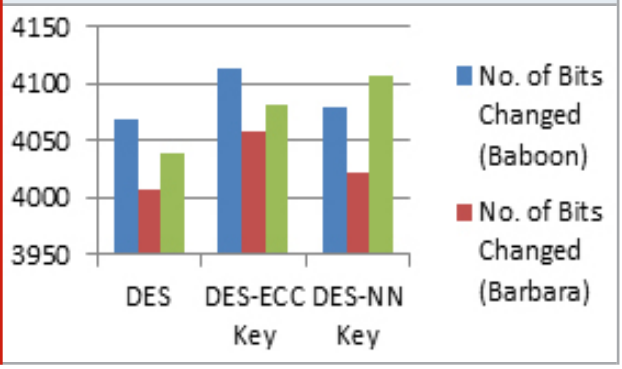


Figure 5: No. of Bits changed in AES, AES with ECC & NN key for different images

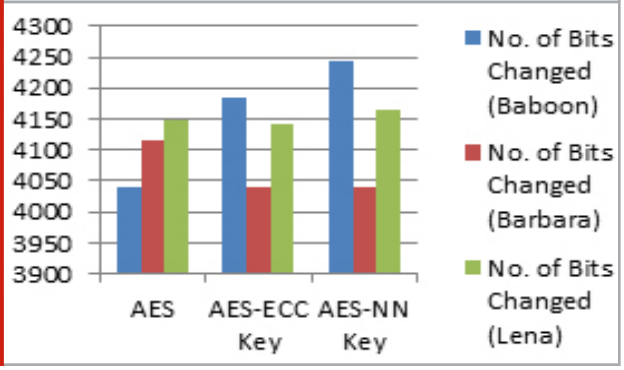


Figure 6: Hit collision in DES, DES with ECC & NN key for different images

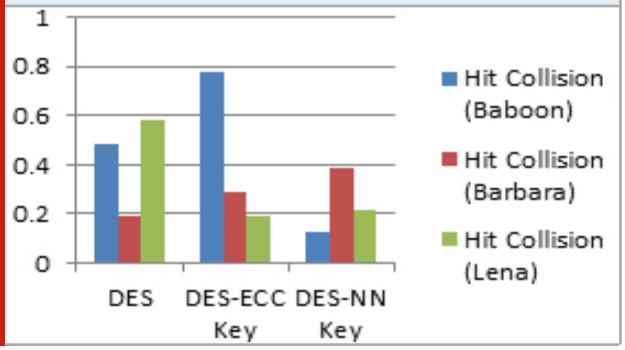


Figure 7: Hit collision in AES, AES with ECC & NN key for different images

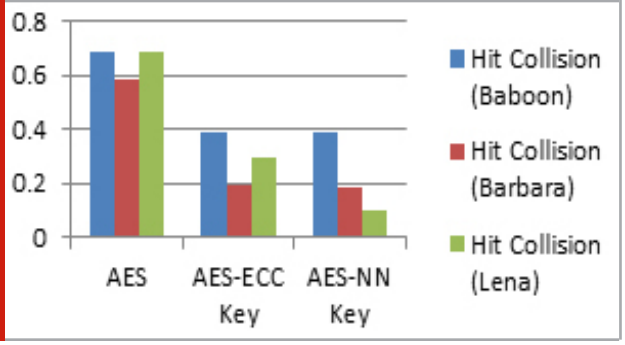
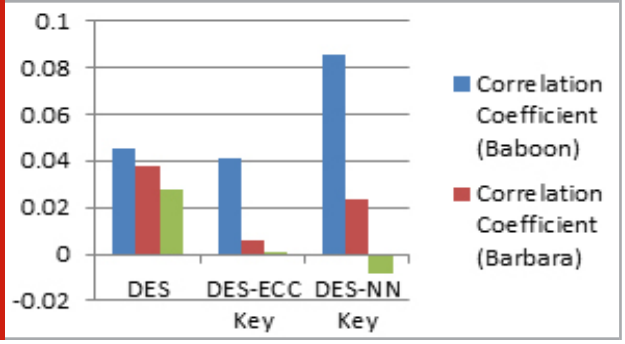


Figure 8: Correlation Coefficient in DES, DES with ECC & NN key for different images

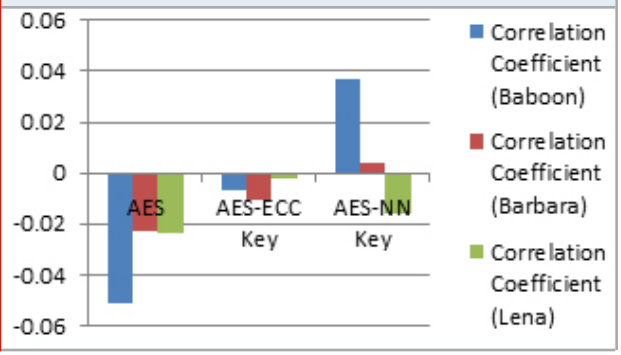


Bits Changed: This is the parameter which tells how many bits are changed between Original data and Encrypted data when we change 1 bit in key.
Plain text sensitivity: Bits changed/total bits

Hit Collision: If we make one bit change in the original key and generate new key using ECC/NN and then compare with original key bit by bit will give the Hit Collision. For better security, the hit collision should be lower.

Correlation Coefficient: It measures the connection between two adjoining pixels of an image (Kumar and Chahal, 2014). It is difficult to guess the image if correlation coefficient between adjoining pixels is small.

Figure 9: Correlation Coefficient in AES, AES with ECC & NN key for different images



CONCLUSION

Encryption is one of the important techniques to secure data/multimedia while using the cloud services. In this paper we have presented a technique to enhance the security of image by using a novel key generation technique. Detailed study is conducted to analyze the important Performance and Security parameters. A comparative study of existing and proposed model is carried out to understand the strength and weakness of each model. Quantitative analysis shows that there is a significant progress in security parameters i.e. 14% increase in Hit collision, 63% in Correlation coefficient and around 1.2% increase in Sensitivity for DES with ECC key when compared with traditional DES encryption method.

When we use NN key generation applied to DES, we see a considerable improvement in Execution time by 8%, CPU usage by 1% and throughput by 8.6% while security parameter Hit collision is increased by 13% in comparison with DES with ECC key. Correlation coefficient is better in the DES with ECC key. For AES with ECC implementation, it is observed that average improvement in CPU utilization is by 0.5%, throughput and execution time results are better in case of traditional AES encryption. Results show significant improvement in Hit collision (by 55%), correlation coefficient (by 76.7%) and Sensitivity (by 0.5%). AES encryption using NN key is showing better results in performance and security parameters except for the correlation coefficient (decreased by 8%) as compared to AES with ECC key. Considering the above results it can be concluded that the proposed AES implementation with NN key achieves the poise between performance and security.

REFERENCES

Chourasia, S., Bharadwaj, H.C., Das, Q., Agarwal, K. and Lavanya, K., 2019. Vectorized Neural Key Exchange Using Tree Parity Machine. Compusoft, 8(5), pp.3140-3145.

Gupta, M., Gupta, M. and Deshmukh, M., 2020. Single

secret image sharing scheme using neural cryptography. Multimedia Tools and Applications, pp.1-22.

Kulkarni, P. and Khanai, R., 2015, April. Addressing mobile Cloud Computing security issues: A survey. In 2015 International Conference on Communications and Signal Processing (ICCSP) (pp. 1463-1467). IEEE.

Kumar, M. and Chahal, A., 2014. Effect of encryption technique and size of image on correlation coefficient in encrypted image. International Journal of Computer Applications, 97(12).

Mahajan, P. and Sachdeva, A., 2013. A study of encryption algorithms AES, DES and RSA for security. Global Journal of Computer Science and Technology.

Nagaraj, S. and Raju, G.S.V.P., 2015. Image security using ECC approach. Indian Journal of Science and Technology, 8(26), pp.1-5.

Pal, S.K. and Mishra, S., 2019. An TPM Based Approach for Generation of Secret Key. International Journal of Computer Network & Information Security, 11(10).

Parameshachari, B. D., Rashmi P. Kiran, P. Rashmi, M. C. Supriya, Rajashekarappa, and H. T. Panduranga. "Controlled partial image encryption based on LSIC and chaotic map." In ICCSP, pp. 60-63. 2019.

Parameshachari, B. D., H. T. Panduranga, and Silvia liberata Ullo. "Analysis and Computation of Encryption Technique to Enhance Security of Medical Images." In IOP Conference Series: Materials Science and Engineering, vol. 925, no. 1, p. 012028. IOP Publishing, 2020.

Prabu, S., V. Balamurugan, and K. Vengatesan. "Design of cognitive image filters for suppression of noise level in medical images." Measurement 141 (2019): 296-301.

Singh, A. and Nandal, A., 2013. Neural cryptography for secret key exchange and encryption with AES. PDF. International Journal of Advanced Research in Computer Science and Software Engineering, 3(5), pp.376-381.

Singh, L.D. and Singh, K.M., 2015. Image encryption using elliptic curve cryptography. Procedia Computer Science, 54, pp.472-481.

Wang, H., Wu, S., Chen, M. and Wang, W., 2014. Security protection between users and the mobile media cloud. IEEE Communications Magazine, 52(3), pp.73-79.

Zissis, D. and Lekkas, D., 2012. Addressing cloud computing security issues. Future Generation computer systems, 28(3), pp.583-592.