International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Symmetric Cryptographic Framework for Network Security

Kanagaraj Narayanasamy, Padmapriya Arumugam

Abstract*: *In this rapidly developing digital environment, a single cryptographic algorithm becomes inefficient and incapable to hold the secrecy of data. A symmetric cryptographic framework is proposed which provides the platform for using the existing and future algorithms in a secured manner. In this research paper, totally six algorithms have been taken into the framework, two algorithms for text, three algorithms for image, and one algorithm for video. The algorithms are grouped into the proposed symmetric encryption framework which provides better network security for the adopted environment. Cryptanalysis and brute force attack have been done to assess the strength of the algorithms incorporated in the framework. Character repetition frequency and brute force attack are analyzed for text encryption algorithms. Mean values, Entropy measure, Differential attack and brute force attack are analyzed and used to assess the reliability of the image and video encryption algorithms. The framework is designed in such a way to adopt the existing and future algorithms. The proposed framework provides a bridge to achieve quality, upgradability, maintainability, and longer usability in applied applications..

Keywords* : *cryptanalysis, framework, randomized framework, symmetric algorithms.

I. INTRODUCTION

A framework provides a bridge to achieve quality, upgradability, maintainability, and longer usability in applications. It allows cryptographers to save time by reusing generic modules which-in-turns help them to focus on other works. The encryption algorithms for text, image and video have been proposed early in [8 -14]. In this latter section, these algorithms are taken to cryptanalysis. Even though these algorithms provide promising results, it's better to keep the algorithms within a framework. A framework is desirable but not essential. A dynamic and robust framework is developed which adopts proposed symmetric encryption algorithms, as well as, the existing and future symmetric encryption algorithms. In this paper, the symmetric encryption algorithms based framework has been proposed and the usefulness of the proposed framework is summarized.

This paper has six sections. In the first section, the necessity to use the framework is discussed. The proposed framework is explained with the help of an overall figure in the next section. In the third section, the working scenarios have been discussed in a detailed manner. A detailed explanation on Cryptanalysis and brute force attack on the algorithms have been given in the fourth section. In the fifth section, the results and discussion have been given. The conclusion and future work of the research is given in the last section.

Manuscript published on 30 August 2019.
***Correspondence Author(s)
Kanagaraj Narayanasamy, Alagappa University, Karaikudi,India. Email: kanagaraj.n.in@ieee.org
Padmapriya Arumugam, Alagappa University, Karaikudi, India. Email: mailtopadhu@yahoo.co.in

II. PROPOSED FRAMEWORK

The proposed framework is developed to handle the user data efficiently by allocating appropriate algorithms to encrypt the user data to disguised form, with the help of a randomizer. Randomizer avoids the immediate repetition of using same algorithms.

The framework consists of four phases. They are

i) Set of algorithms,
ii) Randomizer,
iii) Key Generation, and
iv) Key Distribution Central (KDC).

i) Set of algorithms – The proposed algorithms are the set of algorithms which are used during encrypting the user data. This set can be easily modified to inherit existing and future symmetric based encryption algorithm. This algorithm set provides the facility to encrypt the data which might be text, image or video for the framework.

ii) Randomizer– Randomizer is responsible for choosing a particular algorithm that will be used to encrypt data from the set of available algorithms. This phase rely on the KDC, which provides the data about previously used algorithms. Eventually, it helps the randomizer to choose the next best algorithm for the user data.

iii) Key Generation – The key generation is the key phase of this symmetric framework. This phase will generate the keys as per the prerequisites of the algorithms in the framework. If the user data holds text and images in the same file; then, the key generation will provide keys for text as well as image data. The two key files will be sent individually to the receiver side from the KDC.

iv) Key Distribution Central (KDC) – This phase acts as the storage space for keeping the generated keys, and also used to send the keys to concern receivers. After the generation of keys, the keys are stored till the acknowledgement for the key sent. It acts like key distribution authority to keep and share the keys using an authentication mechanism, that is, One Time Password (OTP). According to the nature of data, an algorithm will be chosen by the randomizer from the set of available algorithms. The key will be generated according to the chosen algorithm and it will be used at sender side. The key file holds the key along with sender address, receiver address, algorithm ID, time of key generation, number of times the key can be used by the receiver (only if the sender requests to keep the key for the later purpose), and key time to live. These data will be saved at KDC (server side) for future usage. These communications between phases in framework are represented as in the below figure 1.

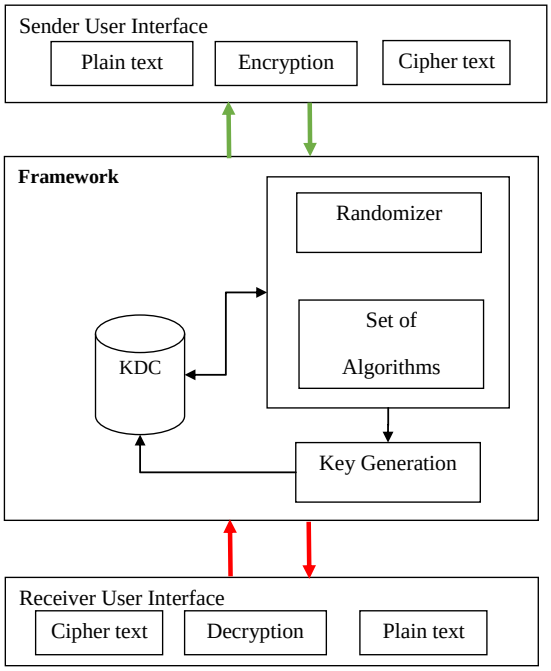

Fig. 1: Overview of Symmetric Cryptographic Framework

From the receiver side, a request is send to server asking for the key. The request is placed with the receiver's address and time of key generation. The OTP will be sent to the receiver's IP address to know the authenticity of the user. Depending on the above two values with OTP, the key will be shared with the receiver and then the encrypted data will be decrypted by the respective algorithm and key.

III. WORKING SCENARIO OF PROPOSED FRAMEWORK

If '*A*' wants to communicate with '*B*', '*A*' will request for the key from *KDC* residing at the server by providing the sender's and receiver's IP address through the User Interface. The request is placed with analyzed report of the input data. Now the KDC chose the algorithm using randomizer and the relevant key and time of key generation will be dispatched to the sender. The sender now sends the encrypted data along with time of key generation; and the key will be given only by the KDC to the receiver.

The receiver will ask for the appropriate key by providing sender and receiver details along with time of key generation. Now the KDC generates OTP for authentication purpose. After the authentication, the key will be shared with receiver.

The process of key generation varies with the input data and the following cases arise.

Case (i) – Text data – If '*A*' wants to send text data to '*B*', the text data are analyzed in the sender's side by the framework itself. The input type, number of words, and blocks are determined and the analyzed report will be sent to the server's side framework. According to the report, the randomizer chose an algorithm from the available list of text encryption algorithm. The key related to the chosen algorithm will be generated. The number of keys will be depending upon the number of blocks which is provided in the analyzed report.

Case (ii) – Image data – If '*A*' wants to send image data to '*B*', the input image is analyzed and the report will hold type of input, the size of the height and width, and number of color channels. If the height is not equal to width, the input image will be cropped to square with the sender's opinion. The analyzed content will be sent to server side to choose algorithm report by the randomizer. The key will be generated according to the chosen algorithm. For *i*-TSS and *i*-TEE algorithms, the keys are generated according to the size of the input image and color channels. If the randomizer chose *i*-S^{2}SLE, key will be generated based on the Chosen Value (*CV*) and linear equation.

Case (iii) – Video data – If '*A*' wants to send video data to '*B*', the input video is analyzed and the following data will be in the report, they are type of input, number of frames per second (*fps*) and frame size. The analyzed report used to fetch an algorithm from the server's framework using randomizer. If *v*-CSS is chosen, then a single channel (color band) based key file will be generated in random fashion.

IV. CRYPTANALYSIS AND BRUTE FORCE ATTACK

In [4], hackers get to know about security pitfalls of the cryptosystem in order to break the system partially or completely. Cryptanalysis reveals the difference between expectations during development phase and reality before deployment of the cryptosystem. Cryptanalysis is not only used to reveal the decrypted message without key but is also used to reveal the strength of the algorithms [4]. The cryptosystem must not reveal any information about plain text, even partially. If the cryptosystem did not maintain the secrecy, then there is no use in choosing the system to send the message securely. The brute force attack is another method used by attackers to crack the ciphered content. In this exhaustive key search the attacker tries different combinations of keys to find the exact key. In this section, the algorithms incorporated in the framework are cryptanalysed and tested for brute force attack to find its usability in the real world.

A. Cryptanalysis on Text Encryption Algorithms

In S^{2}SE and S^{2}SLE, the characters and the words are shuffled, so that the exact words cannot be retrieved in a proper way without the exact key. Consider the number of keys as '*Kn*' for '*n*' number of parts in input data which are generated in random. It makes difficult for adversaries to determine the exact keys. Even though in case of these keys are determined by intruders, the set of keys used once won't be repeated. This provides the stability for the cryptosystem.

The distribution of the characters in a message can be analysed using "frequency analysis". Analysis of frequencies help cryptographers to assess the proposed algorithms using the fact that some characters are repetitively used in English, like, E, T or A. Frequency analysis allows decrypting a cipher text by comparing frequency of letters in a plain text message with letter frequencies in a ciphered message.

The analysis can be done basically by creating bigrams and trigrams with blocks mode and sliding window mode. In table I, the sample plain texts used are tabulated with total number of words and characters. The plain texts are used in both S^{2}SE and S^{2}SLE.

Table I: Total number of words and characters in input text

S. No	Total number of words	Total number of characters	Input text
1	9	40	America is going to attack on Cyria this weekend
2	6	49	Symmetric cryptographic framework for network security
3	11	51	Girl slips from third floor balcony, dangles by neck in China
4	24	131	The Income Tax Department conducted raids at 74 locations in Chennai and Coimbatore in connection with a tax evasion probe against realty business holders

The plain texts are encrypted using S^2SE and S^2SLE algorithms and the corresponding cipher texts have been tabulated in table II and V. From the tables II and V, the characters in the cipher text are analyzed using bigrams and trigrams with blocks analysis mode and sliding window mode. The count of occurrence of repeated trigrams and bigrams in plain texts and cipher texts are calculated and tabulated in table III, and table VI. The count of occurrence of repeated characters (one time to greater than five times) along with the total number of individual characters in plain texts and cipher texts are tabulated in the table IV and table VII.

Table II: *S^2SE*'s Cipher text for the respective plain text in table I

S. No	Cipher text
1	?H@MD>R <CD^ NJDIO || FGGvx< >v q{tkv cE pggmgpq
2	y5ttl{ypm jlj|yp{Z 5yhtl3vyz bipgkf^iXg_`] Z\knfiZ bfe
3	airC lG nroi mlipf sloos I?@I= ;8E>C<K J8C:FEP; M<:9 BE P?@E9
4	kZdWa^VWdFeSZj`Ua_FWa`VgUfW;VWbSdf_W`Uf6fS[ VSeaUSf[a`de^&ZW``S)[545/3(':58/+4+*544+):/5'4)41/{/:'. <'9/5=4'+>-'/49::85('+;9/4+969+'2:(

Table III: Frequency analysis on S^2SE's cipher text against plain text

S. No	Text	Count of occurrence of repeated trigrams		Count of occurrence of repeated bigrams	
		Blocks analysis	Sliding window	Blocks analysis	Sliding window
1	PT	00/13	00/38	01/19	03/36
	CT	00/13	00/38	00/20	00/39
2	PT	00/16	02/45	01/22	08/39
	CT	00/16	00/47	00/24	02/46
3	PT	00/17	00/49	01/24	03/47
	CT	00/17	00/49	00/25	02/48
4	PT	00/43	05/123	06/57	20/99
	CT	00/44	000/131	03/63	13/118

Table IV: Frequency analysis on S^2SE's cipher text against plain text

S. No	Text	Count of occurrence of repeated characters						Total number of individual characters
		>5x	5x	4x	3x	2x	1x	
1	PT	-	1	3	2	5	7	18
	CT	-	-	-	3	6	19	28
2	PT	-	-	-	1	7	31	39
	CT	-	1	-	6	6	14	27
3	PT	-	3	2	2	7	8	22
	CT	-	-	1	4	11	13	29
4	PT	9	1	1	3	5	8	27
	CT	9	3	3	5	6	15	41

Table V: *S^2SLE*'s Cipher text for the respective plain text in table I

S. No	Cipher text
1	F"G'KE, CJKt ($K#) :: FGG46< >4 Q[TKV C% PGGMGPQ
2	8?33+:8/,)+);8/:w ?8'3+=589 qx!vzumxgvnol ikz}uxi qut
3	/7@g :k <@=7 ;:7>4 A:==A 4*+4(|y0).}6 5y.{10;| D}{z -0 ;*+0z
4	C2< /96./<S =+2 B8-97S /98.?->/H ./:+<>7/8- >C >+3.+ =9-+>398< =6 |2/88+6 3B ;<6:/.A<?6 2;E 1<;;20A6<. ;0 ;86 96A. 5C.@6<D ;.2 E4.6;@A A?</. 2B@6;2@= @2.9A/

Table VI: Frequency analysis on *S^2SLE*'s cipher text against plain text

S. No	Text	Count of occurrence of repeated trigrams		Count of occurrence of repeated bigrams	
		Blocks analysis	Sliding window	Blocks analysis	Sliding window
1	PT	00/13	00/38	01/19	03/36
	CT	00/13	00/38	00/20	01/38
2	PT	00/16	02/45	01/22	08/39
	CT	00/16	00/47	00/24	02/46
3	PT	00/17	00/49	01/24	03/47
	CT	00/17	00/49	00/25	02/48
4	PT	00/43	05/123	06/57	20/99
	CT	00/44	00/131	04/62	15/117

Table VII: Frequency analysis on S^2SE's cipher text against plain text

S. No	Text	Count of occurrence of repeated characters						Total number of individual characters
		>5x	5x	4x	3x	2x	1x	
1	PT	-	1	3	2	5	7	18
	CT	1	-	1	-	6	18	26
2	PT	-	-	-	1	7	31	39
	CT	-	1	-	4	8	16	29
3	PT	-	3	2	2	7	8	22
	CT	-	-	1	5	10	12	28
4	PT	9	1	1	3	5	8	27
	CT	10	2	2	4	5	6	29

The analysis show the deviation from the plain text and cipher text as good enough to withstand the cryptanalysis attack using frequency analysis. The extended analysis is available at [7].

The attacks like 'known plain text', 'chosen plain text', 'chosen cipher text', and other plain and cipher text based attacks assumes a key based on what they actually got.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

It can be reliable or unreliable source for them, even though they got the original message, still that deciphered message considered as assumption only. They apply the same key to other messages to get confirmed about the key.

The proposed *S²SE* and *S²SLE* algorithms come under the Symmetric encryption scheme with randomly generated key. The keys are randomly generated, and there will be 'n' keys that will be used for 'n' blocks within the plain text. So these kinds of attacks can't be applied to the randomly generated key algorithms.

Everyone knows the algorithm's procedure, because it has to be available for all. The secrecy lies on the key used [3]. These two algorithms work on the range of 64-112 bits and 80-128 bits key size. Numbers of alternative keys needed for the S²SE and S²SLE algorithms are 1.8*1019 (min) and 3.4*1038 (max). These values reveal that these algorithms are secured enough to withstand for many years. Yet, this will not be declared these algorithms as unconditionally secure algorithms; these algorithms can become vulnerable, if and only if, the intruder has enormous resource.

Still these encryption algorithms can be said to be computationally secure. Because it satisfies the following two criteria:

i. The cost of breaking the cipher exceeds the value of the encrypted information, and
ii. The time required to break the cipher exceeds the useful lifetime of the information. It is very difficult to estimate the amount of effort required to cryptanlayze cipher text successfully.

In S²SLE, even the key file is breached or hacked, the random key cannot be used to reveal the cipher text; because of the fact that the original key (*Kn*) is actually determined from the appropriate linear algebraic equation which is used during encryption.

B. Cryptanalysis on Image and Video Encryption Algorithms

Image encryption algorithms (*i*-TEE, *i*-S²SLE, and *i*-TSS) are already discussed in [8], [9] and [12]. Video encryption algorithm (*v*-CSS) has been discussed in [10].

Barbara, Goldhill and lenna images are used in image encryption algorithms. Video is a collection of frames (images), for better understanding and analysis purpose, the same images which are used in image encryption algorithms are passed into *v*-CSS algorithm too, so that the results from all the proposed algorithms can be tabulated in one.

For assessment purposes the mean values are also provided in table VIII. The table VIII reveals that the mean value of Red, Green and Blue components is shattered in an almost equal manner which exhibits that the encrypted image won't revealed during any statistical attacks.

Table VIII: Mean values of various encrypted images from the proposed encryption algorithms

Algorithms	Baboon	Lena	Jet
***i*-TEE**	Red : 126.88 Green : 127.11 Blue : 127.06	Red : 126.84 Green : 127.01 Blue : 126.98	Red : 127.13 Green : 127.18 Blue : 127.12
***i*-S²SLE**	Red : 127.45 Green : 127.36 Blue : 127.21	Red : 126.95 Green : 126.74 Blue : 127.10	Red : 127.63 Green : 127.32 Blue : 127.36
***i*-TSS**	Red : 127.56 Green : 127.79 Blue : 127.53	Red : 127.57 Green : 127.63 Blue : 127.38	Red : 127.98 Green : 127.73 Blue : 127.99
***v*-CSS**	Red : 127.61 Green : 127.60 Blue : 127.22	Red : 127.78 Green : 127.42 Blue : 127.20	Red : 127.66 Green : 127.59 Blue : 127.63

Entropy is a statistical measure that deals with the randomness of a bundle of data. Theoretically, if the entropy measure of the encrypted images nearly equal to 8 (sh); then the image encryption algorithm is highly robust against entropy attack. Entropy value for the original image (Lena – 512x512) is 7.7502.

Table IX: Entropy values of the encrypted image from the proposed encryption algorithms

Algorithm	Entropy value for Encrypted Image
i-TSS	7.9998
i-S2SLE	7.9973
i-TEE	7.9941
v-CSS	7.9998
Z.Lin etal [16]	7.9890
S.S.Askar et al [15]	7.9961
Zhang et al. [17]	7.9854

From table IX, it is possible to justify the leakage of information of the proposed algorithms against entropy attack to be negligible.

C. Brute Force Attack

The *i*-TEE and *i*-TSS image encryption methodologies uses Randomly Generated bitmap image (*RGbmp*) which is actually an image that created with random values (Pixels). This image (*RGbmp*) is XOR-ed with the encryption-in-need image, which eventually results in an absolute disguised image. *RGbmp* acts as the key. Due to the nature of the origin of *RGbmp*, that is random, the adversaries cannot recreate the original key. For example, if the plain image size is 128 x 128, then the *RGbmp* is created with the same size.

The number of pixels in an image can be found by (l x b), where l and b represents length and breadth of image respectively. Then the number of pixels in the image is equal to 16384. The randomness (combination) to recreate the *RGbmp* will be as follows, (16384 + 16384 + 16384) x 3. This can be rewritten as 16384 x 3^2. From the above calculation, it is possible to derive a generalized formula for various combinations as (l x b) x 3^2. This enormous randomness will avoids the brute force attack easily.

Just like S²SLE, the *i*-S²SLE's key is also derived from the chosen linear equation, so even in the case of exposed key, there is no possibility to acquire the original message at any cost through brute force attack.

V. RESULTS AND DISCUSSION

The previous section evaluates the security of the algorithms used in framework for text, image and video data which was primarily designed with the help of Transposition, Shuffling and Substitution techniques to avoid the unintentional access of data. Mean values and Entropy measure are carried out to quantify the encryption quality and robustness of the proposed algorithms. In this section, the Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE),

Number of Pixels Change Ratio (NPCR) and Unified Averaged Changed Intensity (UACI) results show that the algorithms have strong security and high robustness and the results are also compared with some related encryption methodologies. All these algorithms fall under Symmetric key algorithms. In this type of cryptography method, an intruder cannot be able to compute any information about a plain text from its cipher text. This may be posited as an adversary, given two plain texts of equal length and their two respective cipher texts, cannot determine which cipher text belongs to which plain text. Hence, it shows that these algorithms can said to be perfect secrecy.

PSNR is used to compute the ratio between the maximum possible value of a signal and the power of distorting noise that changes the representation quality [6]. PSNR is expressed in decibel (dB) unit. PSNR is based on the MSE value. MSE is used to calculate the amount of deviation between the original and its disguised image. If the comparison images are identical, then the MSE value will be zero and PSNR would be infinity. If the PSNR value is less; then, the quality of the image encryption is better. PSNR and MSE values for all proposed algorithms are calculated between different original image and its encrypted image; and the values are tabulated in table IX. The results show that the values falls between 19 and little higher than 20; these values reveal that the image is encrypted in a better way and assure that the encrypted images won't disclose any information to the intruders while transferring.

Table IX: PSNR and MSE values between original and respective encrypted images

Algorithm	PSNR	MSE
i-TSS	20.5113	4.461
i-S2SLE	19.7036	5.161
i-TEE	20.1267	4.822
v-CSS	20.2478	4.226

In differential attack, an attacker tries to find the plain image by changing a specific pixel in the image and traces the differences in the respective output image. A general consideration for all encryption algorithms is that the encrypted image must be different from its original image. This deviation can be measured by means of two criteria: NPCR and UACI. The NPCR is used to measure the rate of change in an encrypted image when a bit is changed in the plain image. The UACI is used to calculate the unified average changing intensity between two encrypted images with a deviation in only one bit in respective plain images. In table X, NPCR and UACI values are tabled for proposed algorithms along with comparison with similar works.

Table X: Values of NPCR and UACI tests of Proposed algorithms

Algorithms	NPCR (%)	UACI (%)
i-TSS	99.608	33.43
i-S2SLE	99.6111	30.452
i-TEE	99.6147	33.36
v-CSS	99.6196	33.50
Zhang. J. et al [17]	99.7017	28.7051
Diaconu et al. [5]	99.489	29.006
Dascalescu et al [1]	99.431	25.032
A.L.A. Dalhoum et al [2]	90.126	NaN

In the previous section, the text frequency analysis reveals that the cipher text won't reveal any information during analysis attack. In this section, PSNR and MSE values are tabulated and the results show that the proposed algorithms can withstand the differential attacks at any cost. The NPCR and UACI tests are also carried out and compared with some similar works. These comparative values provide the assurance to stand against differential attacks than other similar works.

VI. CONCLUSION AND FUTUREWORK

A symmetric cryptographic framework is presented in this paper which uses set of symmetric key based encryption algorithms. Even though the encryption algorithm provides security for data, the framework is also used to keep the data intact during communication through the following ways:

i) The data-to-be-encrypted is not sent to the framework; the analyzed reports only sent to fetch an appropriate algorithm.
ii) The algorithm is chosen by the framework's randomizer. Randomizer helps to avoid the repetitive usage of same algorithm which provides stronger reliability in secure transmission of data.
iii) The generated keys are kept in KDC and it is not shared instantly to the receiver; this provides more security.
iv) One Time Password (OTP) provides authenticated access to the key.
v) After receiving the acknowledgement for the key already sent to the receiver; the key will be deleted in KDC. This provides data security even at the moment of KDC compromised.

The set of algorithms can be updated in regular intervals, which extends usability of the framework in terms of time as well as reliability. The authentication can be improvised by providing framework identification number (FID), eventually; it will also provide non-repudiation service.

Nowadays, the top performing information technology companies are not sharing the encryption algorithms to the staff itself. If the adopted algorithm exposed to the world, there may be possibilities for an adversary to sneak in. In future, instead of using individual algorithms, the algorithms can be generated or formed in randomize manner. Each time, a different set of methodologies can be clubbed to frame an algorithm. The randomized generation of algorithm improves more reliability.

ACKNOWLEDGMENT

This article has been written with the financial support of RUSA – Phase 2.0 grant sanctioned vide Letter No.F.24-51/ 2014-U, Policy(TNMulti-Gen), Dept. of Edn. Govt. of India, Dt.09.10.2018.

REFERENCES

1. A. C. Dascalescu, R. E. Boriga, "A novel fast chaos-based algorithm for generating random permutations with high shift factor suitable for image scrambling. Nonlinear Dynamics", Vol.74, pp. 307–318, 2013.

2. A. L. A. Dalhoum, B. A. Mahafzah, A. A. Awwad, I. Aldamari, A. Ortega, M. Alfonseca, "Digital image scrambling using 2D cellular automata", IEEE Transactions on Multimedia, Vol. 19, pp. 28–36, 2012.
3. Auguste Kerckhoffs, " La cryptographie militaire" *Journal des sciences militaires*, vol. IX, pp. 5–83, January 1883, pp. 161–191, February 1883.
4. Bruce Schneier, Applied Cryptography, Second edition, Wiley, 1996, ISBN 0-471-11709-9
5. Diaconu, A. V., Costea, A., Costea, M. A.,"Color image scrambling technique based on transposition of pixels between RGB channels using Knight's moving rules and digital chaotic map", Mathematical Problems in Engineering, 2014.
6. http://in.mathworks.com/help/images/image-quality-metrics.html & http://in.mathworks.com/help/images/image-quality.html [Last Accessed on: 13-07-2019]
7. https://www.researchgate.net/profile/Kanagaraj_Narayanasamy [Last Accessed on: 20-02-2019]
8. Kanagaraj Narayanasamy and Padmapriya Arumugam, "*i*-TEE – An Image Encryption Algorithm based on Multilevel Encryption using a Randomly Generated Bitmap Image". Aust. J. Basic & Appl. Sci., vol 10(2), pp. 150-155, 2016.
9. Kanagaraj Narayanasamy and Padmapriya Arumugam, "*i*-S2SLE: An Encryption methodology for Securing Image using Linear Algebraic Equation", International Journal of Advanced Research Trends in Engineering and Technology, ISSN (online): 2394-3785. DOI: 10.20247/IJARTET.2016.S20040060.
10. Kanagaraj Narayanasamy, and Padmapriya Arumugam, "*v*-CSS: A video Encryption Algorithm based on Conversion, Shuffling and Substitution using Randomly Generated Grayscale Image", International Journal for Research in Engineering Application and management, IJREAM publishing house, Vol 04, Issue no.:12, March 2019. ISSN: 2454 – 9150.
11. N. Kanagaraj and A. Padmapriya, "Cryptanalysis of S^2SE and S^2SLE", NCDSE - National Conference on Data Science and Engineering, South Travancore Hindu college, Nagercoil-629 002, Tamilnadu, India, published by Elsevier, August 2014. pp. 295-301. ISBN: 978-93-5107-294-2
12. N. Kanagaraj and A. Padmapriya, "*i*-TSS: An Image Encryption Algorithm Based on Transposition, Shuffling and Substitution Using Randomly Generated Bitmap Image". In: Bjørner N., Prasad S., Parida L. (eds) Distributed Computing and Internet Technology. Lecture Notes in Computer Science, Springer. vol 9581, pp.148-156, 2016. DOI: 10.1007/978-3-319-28034-9_20
13. N. Kanagaraj and A. Padmapriya, "S^2SE: An Encryption Methodology", in IJCA Proceedings on International Conference on Computing and Information Technology 2013 IC2IT(2), December 2013. pp. 13-15. ISSN: 0975 – 8887
14. N. Kanagaraj and A. Padmapriya, "S^2SLE: An Encryption Methodology for Securing Data using Linear Algebraic Equations", in Journal of Computer Science and Applications, Vol. 6 Number 1 (2014). pp. 309-312. ISSN : 2231-1270
15. S. S. Askar, A. A. Karawia, A. Alshamrani, "Image Encryption Algorithm Based on Chaotic Economic Model", Mathematical Problems in Engineering, 2015.
16. Z. Lin and H. Wang, "Efficient image encryption using a chaos-based PWL memristor", IETE Technical Review, Vol. 27,pp. 318–325, 2010
17. Zhang, J., Fang, D., & Ren, H., "Image Encryption Algorithm Based on DNA Encoding and Chaotic Maps", Mathematical Problems in Engineering, 2015.

AUTHORS PROFILE

Kanagaraj Narayanasamy graduated M.C.A from Anna university, Triuchirappalli and M.Phil from Bharathidasan University, Tiruchirappalli. Now he is purusing Ph.D Part time in Alagappa University, Karaikudi and also working as Assistant Professor in the Department of Computer Applications, J.J College of Arts and Science (Autonomous), Pudukkottai, Tamilnadu, India. His research interest is Cryptography, and Cyber Security. He is an IEEE student member since late 2013.

Dr. A. Padmapriya, is working as Associate Professor in the Department of Computer Science, Alagappa University, Karaikudi – 630 003, Tamil Nadu, India. She has 15 years of teaching experience and 11 years of research experience. She has published many papers in reputed journals and conferences. Her research areas include Data analytics, Communication Networks and Information security.