Defence Science Journal, Vol. 70, No. 4, July 2020, pp. 425-439, DOI : 10.14429/dsj.70.15467

Cryptanalysis of an Image Cipher using Multi-entropy Measures and the Countermeasures

Ram Ratan[*] and Arvind Yadav[#]

[*]*DRDO-Scientific Analysis Group, Delhi - 110 054, India*
[#]*Hansraj College, University of Delhi, Delhi - 110 007, India*
[*]*E-mail: ramratan_sag@hotmail.com*

ABSTRACT

The use of same keys or equivalent keys should not be occurred in cryptographic communications because a cipher system utilising such keys to secure messages can be attacked even it possesses excellent cryptographic characteristics for extracting intelligible information from encrypted messages. Identification of crypts formed with such keys is an important task of traffic analysis of cryptographic communications to check the applicability of two-messages-on-same-key (TMSK) attack. To avoid its applicability, adequate safeguards are required. In the paper, we cryptanalyze stream encryption based cipher system and propose an intelligent identification methodology using multi-entropy measures and soft decision criteria for identification of encrypted images of same or equivalent keys. Experimental test results show that the crypts formed with same keys can be identified successfully with high precision. We also present the countermeasures against TMSK attack.

Keywords: Countermeasures; Cryptography; Fuzzy classification; Multi-entropy measures; Stream cipher; TMSK attack; Traffic analysis

1. INTRODUCTION

The advancement in information technology has increased the use of computer and mobile communication networks as well as multimedia data in text, audio, and visual form to exchange information. Such communication networks are open and an adversary may attack to extract vital information. Security of data is an important requirement to safeguard our vital information. The security of data can be achieved by the techniques of cryptography[1] to conceal the contents, steganography to conceal the existence[2,3], secret sharing to decompose data into different parts[4], and spread spectrum communication to spread data over available bandwidth[5,6]. Cryptography based cipher system consists of encryption algorithm which transforms plain-messages into encrypted messages using encryption keys and decryption algorithm decrypts encrypted messages using decryption keys to obtain plain messages. An encryption algorithm may be based on symmetric-key-cryptography or asymmetric-key-cryptography. Symmetric-key-cryptography uses encryption key and decryption key same and it is kept confidential. Asymmetric-key-cryptography uses encryption key and decryption key different where encryption key is kept open and decryption key is kept confidential. This paper concerns to symmetric-key-cryptography to analyse stream enciphering based cipher system for cryptographic communications. A stream cipher[7,8] consists a pseudo random number generator (PRNG) to generate random binary key sequences are used to encrypt plain messages. The cryptographic primitives and parameters of an encryption algorithm of a stream cipher should be strong enough and the keys sequences should be random to avoid the applicability of cryptanalytic attacks. A stream cipher with appropriate cryptographic parameters possesses excellent cryptographic characteristics and immune to cryptanalytic attacks is called a cryptographically strong cipher system. Many pseudo random number generators[9-14] are reported to design stream ciphers. Boolean functions[15,16], hash functions[17,18] and chaotic functions[18-22], have also been reported to design pseudo random number generators.

In cryptographic communications, an adequate management of keys is necessary to avoid the repetition of keys and the applicability of attacks even an encryption algorithm is strong enough. The repetition of same keys may arrive due to weakness in key generation and inadequate use and handling of secret keys. Key management caters to prevent the leakage keys at any stage starting from its generation and finally loading into the crypto system. Key management practices[23-26] should have an adequate and secure chain of actions including key generation, key loading into storage media, key transportation, transfer of specific key into specific device and loading into the crypto systems securely. It also includes to maintain the record of actions assuring change keys and erasing of used keys timely.

A cipher system can be attacked for extracting message (even partial or distorted message), identifying key (even few consecutive or non-consecutive key bits) and reducing bruit force complexity of encryption algorithm. An attack can

Received : 04 February 2020, Revised : 12 May 2020
Accepted : 19 June 2020, Online published : 13 July 2020

be applied under different conditions as known plain data, chosen plain data, chosen cipher data and known cipher data. Cryptanalysis is attempted for image encryption schemes based on pixel scrambling[27], pixel inversion[28], chaotic[29-31] and auto-locking and electrocardiography[32]. An attack applied under only known cipher data is very difficult. The cryptanalysis of image stream cipher system is attempted under a condition when only cipher data is known to identify encrypted images with same keys.

For a cipher system, the applicability of approximately equivalent keys may arrive if random key sequences differ slightly from each other for different keys due to weak cryptographic algorithm and its weak initial bits. The use of equivalent keys is not safe because encrypted messages can be decrypted by some of its equivalent keys. Hence, encryption keys should be unique for every message otherwise a TMSK attack can be applied. A stream cipher become vulnerable to attack if it encrypts two or more messages with same key. The construction of plain text is formed based on the knowledge of language characteristics. The words are guessed in one message, and these words are *xor* bit-by-bit with the *xor* of encrypted messages. The meaningful words of other messages are formed at places corresponding to correct guess words, otherwise a garbled text is formed. In the case of image ciphers, the guessing of image portion is very difficult to construct plain images until we guess (know) the exact portion of images. Even if we know the portion of an image, knowing of its exact gray level detail is extremely difficult. For encrypted images with same encryption key, the *xor* of these encrypted images gives intelligible information. In TMSK attack for encrypted text messages, probable words are guessed in one message based on intelligibility and meaningful words are formed in other message to solve crypts[33-34]. Finding the applicability of TMSK attack can be found by identifying crypts formed with same encryption keys.

Analysis of huge traffic is a challenging task to check lapses and leakage of cryptographic system, and to find meaningful information from adversary's traffic of cryptographic communications. It is the process to intercept and examine communications and requires huge data collection and rigorous processing. Traffic analysis caters the analysis and monitoring of data communicated and communication activities over communication networks. It is an important Task and lot of relevant work is reported in traffic analysis[35-39] which helps in future planning of networks[36], formulation of practices[37], network selection[38], traffic prediction[39], identifying and classifying communications such as watermarked and noisy images[40], multimedia and text classification[41-43], engineering of communication networks[44], and classification of encryption algorithms[45-46]. Identification of cipher data of same key is one of the task of traffic analysis to identify and segregate for extraction of vital intelligible information of adversaries. For classification of objects of interest, various techniques of statistical, structural, and nature inspired pattern recognition[47-51] can be applied. Fuzzy computing is one of the soft computing techniques which classifies objects using the degree of belongingness and the similarity criteria for ambiguous and ill-defined class boundaries. Fuzzy decision criteria gives better classification compared to other conventional and binary decision criteria. Fuzzy concepts are applied successfully with other methods in a hybrid manner to improve their performance[50,51]. It is also reported that a fuzzy classification approach[52] performs better compared to the support vector machine (SVM) to identify messages encrypted with same keys from the traffic of cryptographic communications. The classification performance is analysed by utilising the criteria of Wilkinson test[53]. Various statistical measures[54-59] such as histogram, correlation, and co-occurrence entropy are reported to use as features of the patterns. We use the multi-entropy measures computed globally and locally as features of the images to classify them in classes as plain image or encrypted image and images encrypted with same keys or different keys.

We present an identification methodology using multi-entropy measures to characterize plain and crypt messages and apply these collectively in fuzzy decision to identify crypts formed with same/equivalent encryption keys. We use asymmetric membership functions[60-61] to cater the variations in entropy measures of plain messages for computing degree of belongingness and maximum mean similarity scores to identify the class. The methodology identifies crypts of same/equivalent encryption keys with high precision as shown in the results. For having a cryptanalytic immune cipher system, we present some countermeasures to safeguard cryptographic communications.

2. CIPHER SYSTEM

A cipher system[62] is a cryptography based software/hardware system which provides security to sensitive data. The main components of a cipher system are input-output modules, random source module, key scheduling module, and encryption algorithm module. When a call is initiated by a sender, random source module generates random bits. These bits go to key scheduling module which generates initial bits. These initial bits go to encryption algorithm module to initialize it. The encryption algorithm module of stream based cipher system after initialising with unique initial bits for every message, generates pseudo random key sequence which is added bit-by-bit using *xor* operation with input message comes from input module to produce encrypted message. The encrypted message goes to the output module which send it to the receiver using secure communication protocols. At receiving end, the encrypted data is decrypted to get plain message.

Apart from strong cipher modules with its adequate security strength, a cipher system should also have the following various other provisions:

- Access control mechanism to assure authenticated usages for operation.
- Integrity mechanism to check alteration/corruption of codes and data.
- Maintenance and administrative activities of cipher system for audit.
- Erase mechanism to delete sensitive data.
- Tamper detection and response mechanism to prevent leakage of sensitive system information.
- Key change mechanism to prevent repetition of keys.

- Keyboard and display to access and operate system with ease and comfortably.
- Health check of system.
- Error display and alarm indication in case of corruption and system failure.
- Updating of cryptographic algorithm, key and access parameter whenever required.

A cipher system should be analysed carefully for ensuring its cryptographic strength against cryptanalytic attacks. The strength of cipher system depends on the design of encryption algorithm and cryptographic primitives used. For a secure system, the cryptographic primitives and modules should be mathematically and statistically strong and should meet expected characteristics of cryptographic parameters such as brute force attack complexity, periodicity, linear complexity, unpredictability, and randomness. An attacker tries to obtain information about algorithm, keys, sequences and messages. According to Shannon criteria, a cipher system should not leak such information even with the availability of unlimited computing resources and encrypted data[57]. According to Kerckhoffs criteria, the security of information should lie entirely on the keys but not on the obscurity of the cryptographic algorithm[63], i.e., the cryptographic algorithm except keys is public. Hence, if keys are not utilised and handled properly to avoid the repetition of keys and its improper use in initialisation of cryptographic algorithm then the cipher system becomes insecure and it leaves the scope of the applicability of TMSK attack.

A stream cipher based cipher system generates random binary sequences using cryptographic algorithm which consists shift registers. For maximal length binary key sequences with high non-linearity and randomness, proper feedback connections and suitable filtering and combining functions are chosen in the design of encryption algorithm. A key sequence is added with plain data bit-by-bit under modulo two to get encrypted data. Let K_1 and K_2 be the key sequences, P_1 and P_2 be the plain messages and, C_1 and C_2 be the encrypted messages, the encrypted images formed for different plain messages with different/same key sequences are given by following equations:

$$C_1 = P_1 \oplus K_1, C_2 = P_2 \oplus K_2 \quad (1)$$

$$C_1 = P_1 \oplus K_1, C_2 = P_2 \oplus K_1 \quad (2)$$

$$C_1 = P_1 \oplus K_2, C_2 = P_2 \oplus K_2 \quad (3)$$

Symbol $\oplus$ denotes modulo two addition (*xor* operation) which gives output '1' when two input are different otherwise '0'.

Applying *xor* operation on encrypted messages C_1 and C_2 for which same key (K_1 or K_2) is used to encrypt P_1 and P_2, we get

$$C_1 \oplus C_2 = P_1 \oplus K_1 \oplus P_2 \oplus K_1 = P_1 \oplus K_2 \oplus P_2 \oplus K_2 = P_1 \oplus P_2 \quad (4)$$

In Eqns. (1) - (4), we see that P_1, P_2 are not random, K_1, K_2 are random, $K_1 \oplus K_2$ is random, $P_1 \oplus P_2$ is not random, C_1, C_2 are random and $P_1 \oplus K_1$, $P_1 \oplus K_2$, $P_2 \oplus K_1$, $P_2 \oplus K_2$ are random. In Eqn. (4), the effect of *xor* of same keys, i.e., $K_1 \oplus K_1$ and $K_2 \oplus K_2$ gets cancelled.

Hence, we get the *xor* of plain messages P_1 and P_2 when we *xor* C_1 and C_2 which are formed with same encryption keys.

For messages of image type, plain images appear random and intelligible but encrypted images appear random and unintelligible. The output images after *xor* of encrypted images with same keys look non-random and such images exhibit meaningful information which depends on the content and gray level variations of plain image. Such characteristics are exploited in identifying images encrypted with same keys.

Two keys are said to be same if sequences for these keys are bit-by-bit same, i.e., if K_1, K_2 are two sequences then $K_1(i) = K_2(i)$, $i = 1$ to n where n is the length of sequence.

Two keys are said to be equivalent if sequences for these keys are approximately same and few bits differ, i.e., $K_1(i) \neq K_2(i)$, for some i, $i = 1$ to n. If k is the number of bits differ in sequences K_1 and K_2 then $n-k$ is the number of bits remain same in these sequences. For equivalent keys, the value of k may be considered different for different kinds of data. Say, the keys are said to be equivalent with 10% dissimilarity if there are 10% mismatch of bits in key sequences. For example, the value of k may be higher for image or speech data because these data are highly redundant and slight change in such data is unnoticeable. Also, the value of k may be lesser for text data because it is less redundant and a slight change in text is significantly noticeable.

In identification of encrypted data of same keys or equivalent keys, we treat both the cases as of same keys.

3. MULTI-ENTROPY MEASURES AND CHARACTERISTICS

The randomness or non-randomness of messages can be found using a number of statistical measures such as grey level distribution, adjacent correlation and co-occurrence measures. In this paper, we consider an entropy measures[58,59] obtained globally and locally for sub-block and bit-plane of images to identify crypts of stream ciphers formed with same keys. An entropy measure proposed by Claude Shannon is associated to random variables and uses as a measure of information.

3.1 Global Entropy

Entropy of a random variable *X* is given by the equation

$$H(X) = -\sum_{i=1}^{L} p_i \log_2 p_i \quad (5)$$

where $p_i = prob(X = x_i), x_i$ is the i^{th} possible value of X from L symbols and p_i is the possibility of $X = x_i$. The value of $H(X)$ lies between 1 to $\log_2 L$, i.e., $1 \le H(X) \le \log_2 L$. For a random message of L symbols in which each symbols occurs equally ($p_1 = p_2 = ... = p_{L-1} = p_L = 1/L$), the value of entropy is given by $\log_2 L$. The entropy measure computed for a message S is called as the global entropy of S.

Global entropy varies from 0 to $\log_2 L$ for non-random data. It may not be able to exhibit the local non-randomness of S for which $H(S)$ is near to $\log_2 L$ which is the highest value of $H(S)$ for a random data. To see the local non-randomness of S, local entropies can be considered.

3.2 Sub-block Entropy

Sub-block entropy (k, T_B) with respect to local k number of non-overlapping sub-blocks $S_1, S_2, ...S_K$, each of size T_B within S, are obtained by finding entropy $H(S_k)$ over all the sub-blocks. The size of an image should be reasonable to get the fair estimate of Shannon entropy. Also, the value of k and T_B should be adequate in number and size respectively to get fair values of local Shannon entropy. A k number of sub-blocks of S can be taken in non-overlapping manner randomly instead of all the sub-blocks of S. Average sub-block local entropy is taken as the average of local entropies computed for a number of different sub-blocks to fairly cater the local randomness of S.

As the finding of global entropy requires entire S and finding of local entropy requires only the portion of S, the local entropy is computed efficiently. In view of inaccuracy of global entropy, it is preferable to use local entropy to see the local non-randomness of S.

3.3 Bit-plane Entropy

When global and sub-block entropy measures are not giving information of local non-randomness of S, we can compute local entropy for bit-planes of S in a non-overlapping manner as similar to sub-blocks local entropy to see the non-randomness of S at bit-plane level.

For an image $S(x, y)$ with 8 bit/pixel, we have 8 different bit-planes, $b(x, y, k), k = 1$ to 8 where k indicates the number of bit-planes. Bit-plane $b(x, y, 1)$ is known as most significant bit (MSB) plane and $b(x, y, 8)$ is known as least significant bit (LSB) plane.

To see the adequacy of the global and local entropy measures, we consider an image $I(x, y)$ of size 256×256 as shown in Fig. 1, where $I(x, y) = \text{mod}(x \times y, 256)$ and $1 \le x, y \le 256$.

Values of global, sub-blocks local and bit-pixel local entropies computed for an image of Fig. 1 are given as:

- Global Entropy = 7.7117
- Sub-blocks local entropies (from upper-left to lower-right sub-blocks) = 7.7037, 7.7037, 7.7037, 7.7011
- Bit-planes local entropies (from MSB to LSB bit-planes) = 0.8113, 0.9544, 0.9887, 0.9972, 0.9993, 0.9998, 1.0000, 1.0000

The global and local entropy measures have different values. The values of global entropy and block entropies are on higher side and all appears approximately equal, and it is difficult to differentiate such images from random data. The values of bit-plane local entropies are increasing from MSB plane to LSB plane. The individual use of these entropies may not give good detail of non-randomness. Applying these entropies together can give good measures of non-randomness of data. The mean values of sub-blocks entropies and bit-planes entropies can also be computed by measuring their averages.

3.4 Entropy Specific Image Characteristics

To study the characteristics of plain images and encrypted images with respect to entropy measures, we consider a number of images and compute global entropy and local entropies for sub-blocks and bit-planes of images.

For illustration, we present the values of multi-entropies measures for three different images Baboon (I1), Monalisa (I2) and Cameraman (I3) each of size 256×256 and their encrypted images C1 = I1 xor K1, C2 = I2 xor K2, C3 = I3 xor K3, and *xor* form of images C1 xor C2, C1 xor C3, C xor C3, C1 xor C4, C1 xor C5, C4 xor C5 as shown in Figs. 2, 3, 4.

The key sequences K1, K2, K3 are obtained through a random number generator form Matlab. One can consider any generator to get random sequences for such image stream encryption. Figure 2 shows plain images I1, I2, I3 and key sequence K1, K2, K3 along with their histograms. Figure 3 shows encrypted images along with their histograms. Figure 4 shows images from xor of encrypted images along with their histograms. The values of global and local entropies for images of Figs. 2, 3, 4 are given in Table 1, 2 where Table 1 shows the global entropy and sub-block entropies and Table 2 shows the bit-plane entropies.

From Fig. 2, we see that the plain images appear intelligible and their histograms appear non-uniform. The keys shown as images appear unintelligible and their histograms appear uniformly distributed. From Fig. 3, we see that encrypted

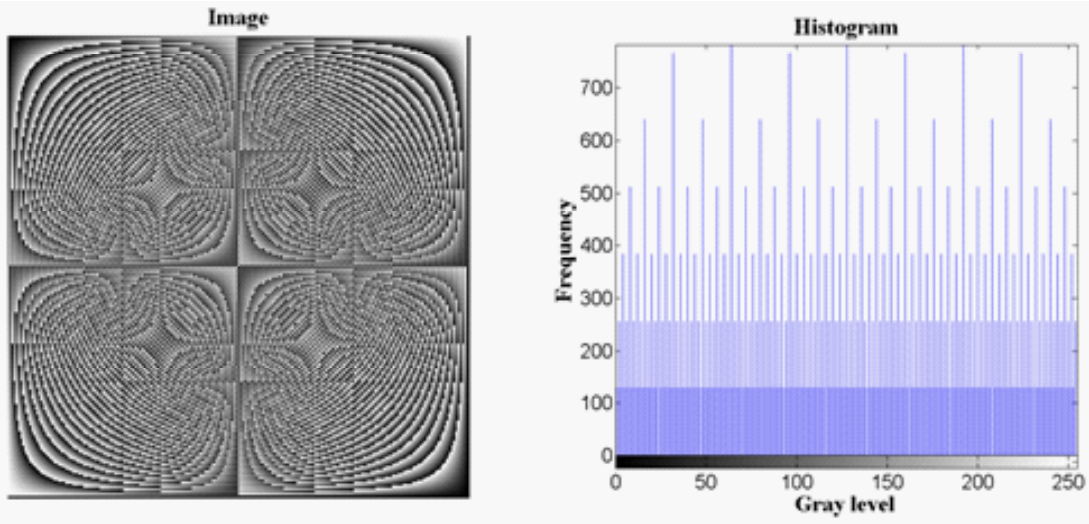

Figure 1. Simulated image and its histogram.

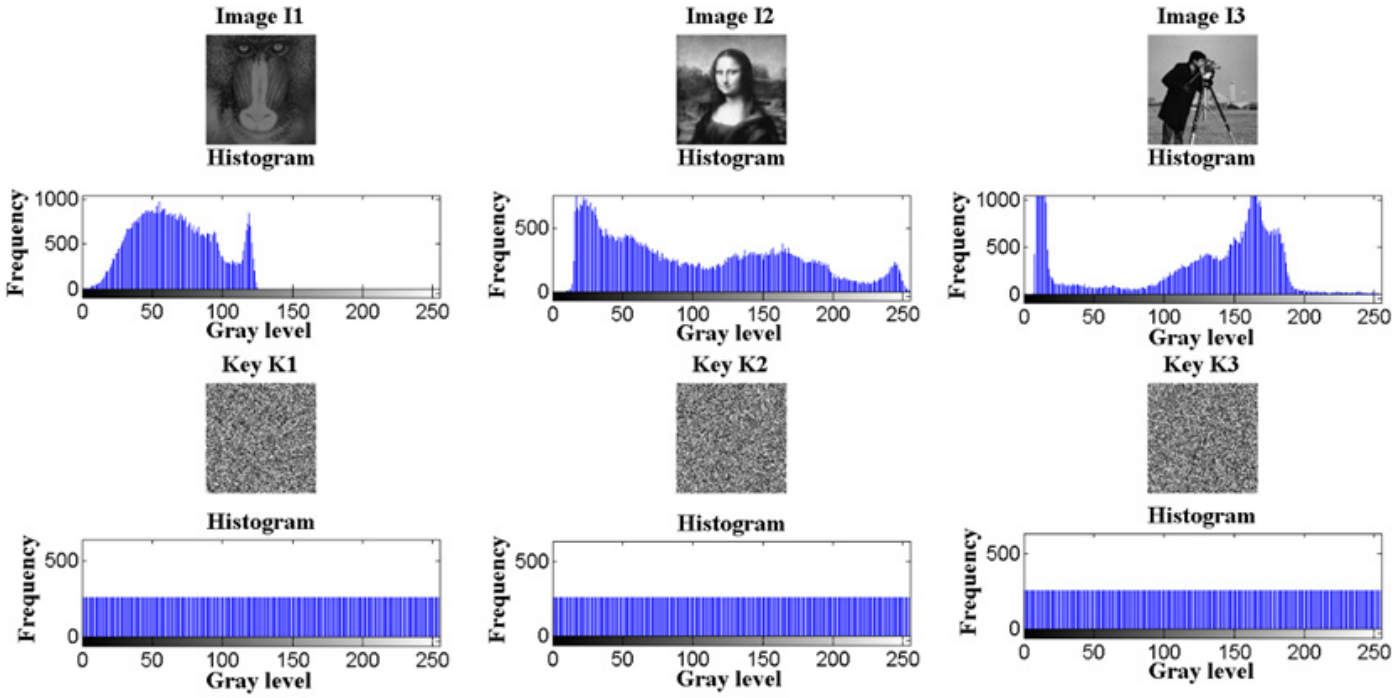

Figure 2. Plain images (Baboon, Monalisa, Cameraman), key sequences and their histograms.

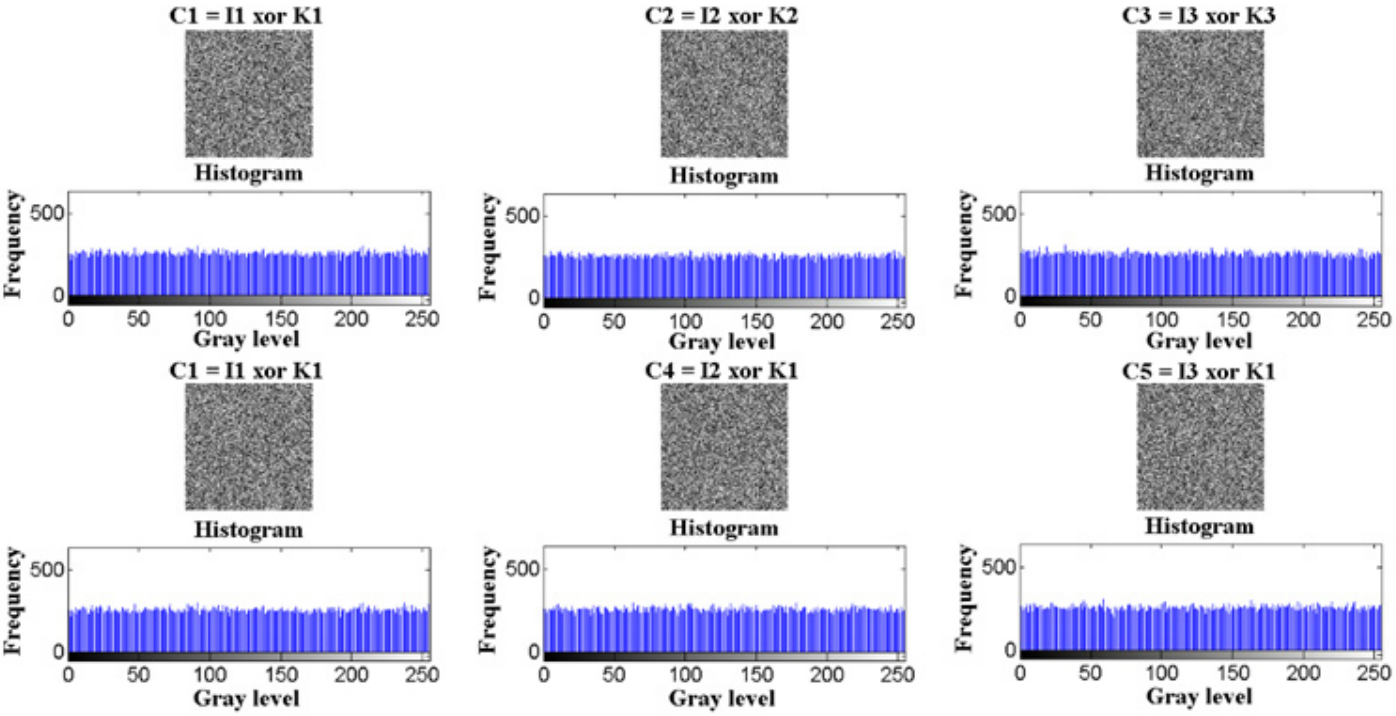

Figure 3. Encrypted images with different keys and same keys along with their histograms.

images appear unintelligible and their histograms appear flattened and uniformly distributed. From Fig. 4, we see that images from *xor* of encrypted images with different keys appear unintelligible and their histograms appear flattened like the histograms of encrypted images. The images from *xor* of encrypted images with same keys appear with some intelligibility and some meaningful information is visible in distorted form and their histograms appear non-uniformly distributed.

From Table 1 and 2 it is observed that the values of global, sub-block and bit-plane entropies of plain images and images from *xor* of encrypted images with same keys are lesser to the entropies of encrypted images and also to the entropies of images from *xor* of encrypted images with different keys. The bit-plane entropies of MSB planes of plain images are lesser than to the bit-plane entropies of LSB planes of plain images. The bit-plane entropy of encrypted images and *xor* encrypted images with different keys has high values and is almost

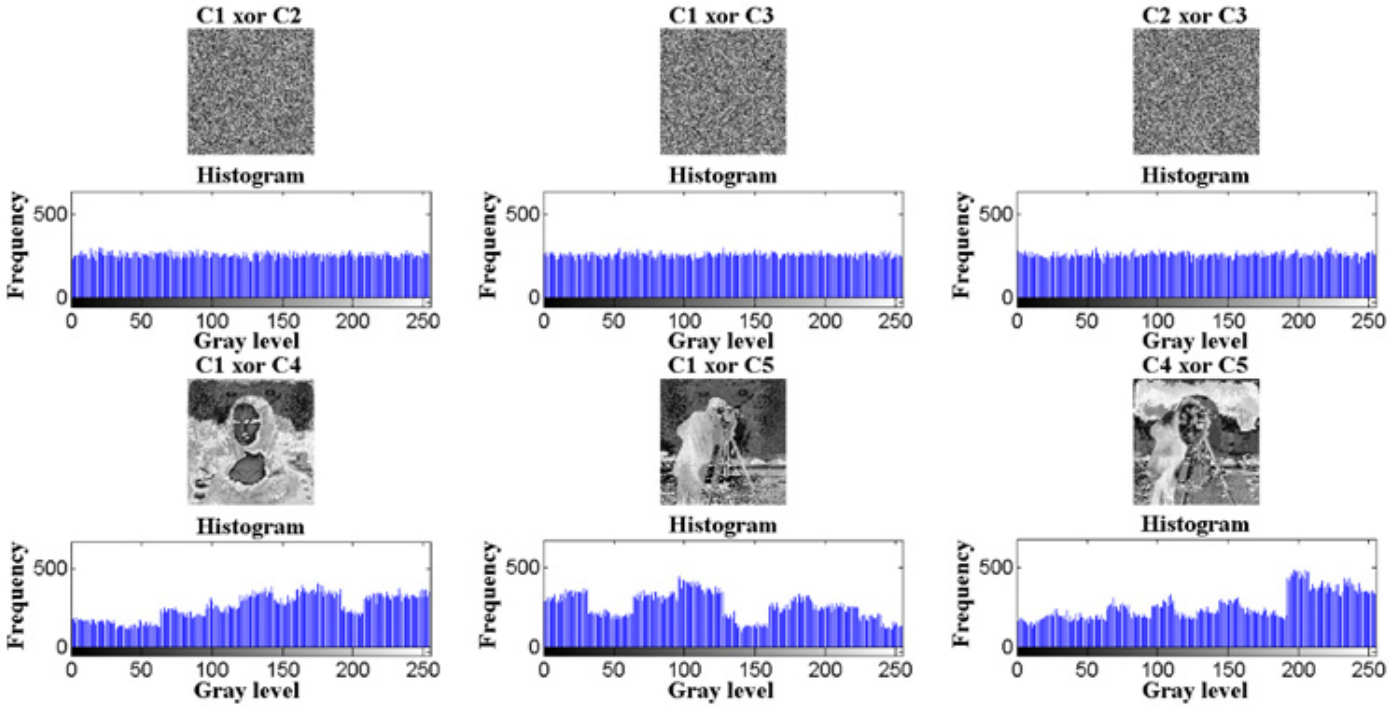

Figure 4. Images after xor of encrypted images with different keys and same keys and their histograms.

same for all the bit-planes starting from MSB to LSB planes. The entropy values of *xor* encrypted images with same keys have lesser values but different from entropy values of plain images. The values global entropy and sub-block entropy may vary from 0 to 8 and the values of bit-plane entropy may vary from 0 to 1 depending on the content of plain images.

These entropy measures can be used as features to identify the encrypted images formed with same keys or different messages. From the analysis of these entropies for different images we come to know that the entropy measures vary in wider range. The conventional approaches of pattern recognition for classification of such images are not suited and we consider a fuzzy decision approach to identify images encrypted with same or different keys. The detailed analysis of these entropies for different images of each class helps in establishing the values of parameters of fuzzy membership function.

Table 1. Global entropy and sub-block entropy measures for plain images, key sequences, encrypted images and xor of images

Image	Global entropy	Local entropy				
		Block 1	Block 2	Block 3	Block 4	Avg.
I1 = Baboon	6.6962	6.6299	6.4968	6.6610	6.4956	6.5708
I2 = Monalisa	7.6916	7.6756	6.4412	7.2327	7.0098	7.0898
I3 = Cameraman	7.0097	5.8926	6.1224	6.5959	7.1177	6.4322
K1 = Random Seq.	8.0000	7.9953	7.9953	7.9942	7.9942	7.9948
K2 = Random Seq.	8.0000	7.9946	7.9946	7.9933	7.9934	7.9940
K3 = Random Seq.	8.0000	7.9940	7.9940	7.9942	7.9942	7.9941
C1 = I1 ⊕ K1	7.9977	7.9894	7.9898	7.9885	7.9896	7.9893
C2 = I2 ⊕ K2	7.9975	7.9885	7.9885	7.9888	7.9876	7.9883
C3 = I3 ⊕ K3	7.9973	7.9898	7.9888	7.9888	7.9874	7.9889
C4 = I2 ⊕ K1	7.9974	7.9893	7.9897	7.9889	7.9890	7.9892
C5 = I3 ⊕ K1	7.9976	7.9906	7.9894	7.9898	7.9825	7.9886
C1 ⊕ C2	7.9971	7.9873	7.9907	7.9871	7.9891	7.9886
C1 ⊕ C3	7.9970	7.9889	7.9906	7.9893	7.9878	7.9892
C2 ⊕ C3	7.9969	7.9889	7.9881	7.9885	7.9882	7.9884
C1 ⊕ C4	7.9273	7.8743	7.9279	7.5852	7.5106	7.7244
C1 ⊕ C5	7.9229	7.7117	7.3550	7.7251	7.8759	7.6669
C4 ⊕ C5	7.9204	7.7745	7.6534	7.8665	7.8479	7.7856

Table 2. Bit-plane entropy measures for plain images, key sequences, encrypted images and xor of images

Image	Bit-plane Entropy								
	BP 1	BP 2	BP 3	BP 4	BP 5	BP 6	BP 7	BP 8	Avg.
I1 = Baboon	0.0000	0.9999	0.9868	0.9961	0.9999	1.0000	0.9999	0.9999	0.8728
I2 = Monalisa	0.9584	0.9285	1.0000	0.9909	0.9987	0.9998	1.0000	1.0000	0.9845
I3 = Cameraman	0.9732	0.6741	0.9994	0.9844	0.9957	1.0000	1.0000	1.0000	0.9534
K1=Random Seq.	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
K2=Random Seq.	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
K3=Random Seq.	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C1 = I1 ⊕ K1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C2 = I2 ⊕ K2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C3 = I3 ⊕ K3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C4 = I2 ⊕ K1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C5 = I3 ⊕ K1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C1 ⊕ C2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C1 ⊕ C3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C2 ⊕ C3	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
C1 ⊕ C4	0.9584	0.9978	0.9979	0.9999	1.0000	1.0000	1.0000	1.0000	0.9945
C1 ⊕ C5	0.9732	0.9956	0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	0.9960
C4 ⊕ C5	0.9680	0.9698	0.9989	0.9987	1.0000	1.0000	1.0000	1.0000	0.9919

4. MULTI-ENTROPY BASED IDENTIFICATION METHODOLOGY

Traffic of communications may include plain data in addition to encrypted data which might be of same keys or different keys. Identifying images encrypted with same keys directly prior to separating out plain images is very difficult because the features values of plain images and encrypted images with same keys are not able to distinguish such images. The methodology of identification of images encrypted with same keys or equivalent keys takes the task as a two class problem and resolves it in following two stages:

- Segregation of plain images and encrypted images and
- Segregation of encrypted images with same keys or equivalent keys and images encrypted with different keys.

The block diagram of identification methodology is given in Fig. 5. The traffic data is segregated in four different classes. In identification methodology, the finding of prominent features of data plays a vital role to classify data accurately. Here, we consider multi-entropy measures to discriminate different data and classify these precisely. The combined use of all these local and global entropy measures can give better classification compared to its individual use for classification of encrypted images. Based on the analysis of the characteristics of different images as plain, encrypted, and *xor* images of encrypted images with same/equivalent or different keys, we consider ten features f_1 to f_{10} given by following equations:

$$f_1 = GlobalEntropy(GE) \tag{6}$$

$$f_2 = SubBlockEntropy(SBLE_2) \tag{7}$$

$$f_3 = SubBlockEntropy(SBLE_3) \tag{8}$$

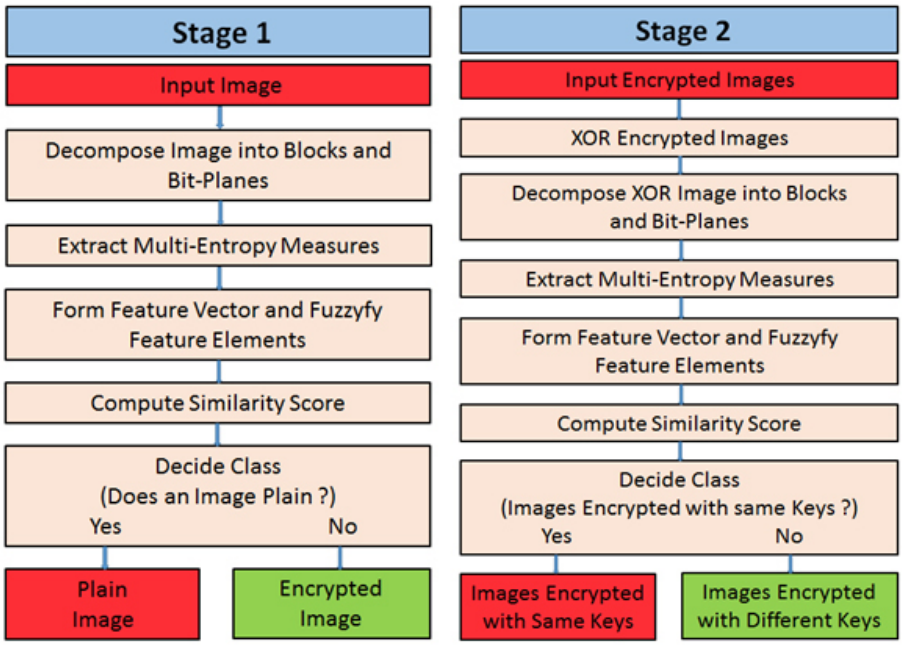

Figure 5. Block diagram of methodology of class identification.

$$f_4 = SubBlockEntropy(SBLE_4) \quad (9)$$

$$f_5 = SubBlockEntropy(SBLE_5) \quad (10)$$

$$f_6 = (1/k1)\sum_{i=1}^{k1} SBLE_i \quad (11)$$

$$f_7 = BitPlaneLocalEntropy(BPLE_1) \quad (12)$$

$$f_8 = BitPlaneLocalEntropy(BPLE_2) \quad (13)$$

$$f_9 = BitplaneLocalEntropy(BPLE_3) \quad (14)$$

$$f_{10} = (1/k2)\sum_{i=1}^{k2} BPLE_i \quad (15)$$

In Eqns. (11) and (15), $k1$ and $k2$ are the number of sub-blocks and number of bit-planes and these are taken as 4 and 8 respectively. These features constitute a feature vector of a pattern.

Fuzzy decision criteria resolves ambiguous problems and gives better classification results as compared to conventional decision criteria. A crisp set based criteria takes binary decision whereas fuzzy set based criteria takes a fuzzy decision in terms of degree of belongingness to perform classification of objects.

For an element x_i of set A, the membership value of x_i to set A, is denoted as $\mu(x_i)$ which is computed using a membership function. Classification of unknown pattern is obtained among reference pattern classes using average similarity scores. A given pattern is categorised to a particular reference class for which it has high similarity value.

For computing the membership value for feature elements, we use a quantitative asymmetric kind of triangular fuzzy membership functions similar to Mamdani-type fuzzy function[60-61].

For rc^{th} reference pattern, let $T_{rc}(i)$ be the feature value and $Th1_{rc}(i)$, $Th2_{rc}(i)$ be the thresholds for i^{th} element of feature vector. Let $P_{\mu}(i)$ be the value of i^{th} element of feature vector of given pattern to be classified. Similarity value $P_{\mu}(rc,i)$ is obtained as $\mu P_{\mu}(rc,i)$ using triangular fuzzy membership function. The parameters $Th1_{rc}(i)$ and $Th2_{rc}(i)$ of membership function are taken carefully based on the knowledge and analysis of measures for different reference class patterns and based on the classification results obtained by varying threshold values. The establishment of parameters of membership function plays a vital role in making decisions and the precise selection of parameter values compute membership values appropriately for performing correct classification. Fuzzy membership function can take symmetric or asymmetric shape depending on values of $Th1_{rc}(i)$ and $Th2_{rc}(i)$. Mathematically, triangular fuzzy membership functions compute the membership values with respect to reference class pattern as given in following equations:

$$\begin{aligned} &If\ P_{\mu}(i) \geq T_{RC}(i)\ then\ \mu(P_{\mu}(i)) = 1 \\ &elseif\ P_{\mu}(i) \leq T_{RC}(i) - Th1_{RC}(i)\ then\ \mu(P_{\mu}(i)) = 0 \\ &else\ \mu(P_{\mu}(i)) = (Th1_{RC}(i) - T_{RC}(i) - P_{\mu}(i))/Th1_{RC}(i) \end{aligned} \quad (16)$$

$$\begin{aligned} &If\ P_{\mu}(i) \leq T_{RC}(i)\ then\ \mu(P_{\mu}(i)) = 1 \\ &elseif\ P_{\mu}(i) \geq T_{RC}(i) + Th2_{RC}(i)\ then\ \mu(P_{\mu}(i)) = 0 \\ &else\ \mu(P_{\mu}(i)) = (Th2_{RC}(i) - T_{RC}(i) - P_{\mu}(i))/Th2_{RC}(i) \end{aligned} \quad (17)$$

Equation (16) computes membership values of feature element for crypt class or crypts with different encryption keys class and Eqn. (17) computes membership values of feature element for plain class or crypts with same encryption keys.

A given image is classified as a plain/encrypted image or encrypted images are classified as images encrypted with same/different keys on the basis of similarity score S_{rc} which is given following equation:

$$S_{rc} = (1/n)\sum_{i=1}^{n} \mu(P_{\mu}(rc,i)) \quad (18)$$

where n is the number of features. An unknown image belongs to the rc reference image class for which the value of S_{rc} computed for features of unknown image is maximum.

5. TEST RESULTS AND PERFORMANCE

Identification methodology is implemented in Matlab programming and applied on a number of plain and encrypted images. For an illustration, we demonstrate experimental tests on different images each of size 256×256 as shown in Fig. 6. In the experiments, we have taken a number of encrypted images obtained by using a number of random binary key sequences and xor with plain images as stream enciphering. Some of the encrypted images are obtained with same key sequences and some are obtained with different key sequences.

Features for each given image as mentioned above and the membership values as well as the similarity scores with respect to reference class patterns are computed to classify a given image. The values of feature vectors of reference patterns are kept same to compute fuzzy membership values for plain/encrypted images and for images from *xor* of encrypted images with same/different keys. Threshold values are also kept same for reference classes.

As the class decision depends on the establishment of parameters of triangular fuzzy membership function, the values of parameters for reference classes and threshold parameters need to be chosen carefully. Appropriate establishment of parameters values of triangular fuzzy membership functions performs correct classification of given patterns. In the experiment for the problem of identification, we perform the analysis of entropy for different images of each reference class to establish their reference values for each feature and threshold values by varying these to attain successful classification. The establishment of parameters of triangular fuzzy membership function requires once only. There is a larger variation in entropy for plain images, slightly lesser variation for *xor* images encrypted with same keys and very less variation for encrypted images and *xor* images encrypted with different keys.

Based on above analysis, the parameter values of triangular fuzzy membership function for reference class plain $T_{plain}(i)$ and *xor* of crypts with same key $T_{sk}(i)$, crypt $T_{crypt}(i)$ and *xor* of crypts with different keys $T_{dk}(i)$ are established as:

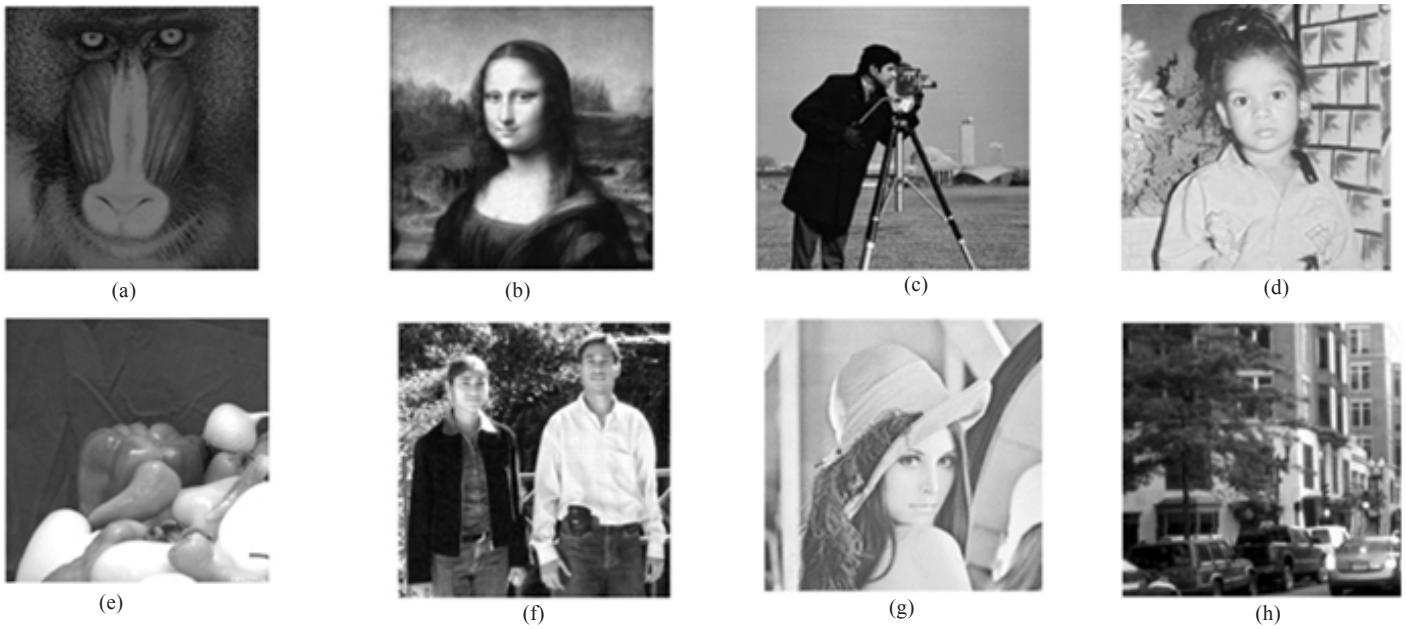

Figure 6. Plain images: (a) Baboon, (b) Monalisa, (c) Cameraman, (d) Baby, (e) Pepper, (f) Children, (g) Lena, and (h) Street.

$T_{plain}(i) = T_{sk}(i)$ = 7.98, 7.95, 7.95, 7.95, 7.95, 7.95, 0.99, 0.99, 0.99, 0.99

$T_{crypt}(i) = T_{dk}(i)$ = 7.99, 7.99, 7.99, 7.99, 7.99, 7.99, 1.00, 1.00, 1.00, 1.00

Based on varying threshold analysis, the values of thresholds for triangular fuzzy membership function $Th1_{rc}(i)$ and $Th2_{rc}(i)$ for reference classes plain $Th1_{plain}(i), Th2_{plain}(i)$ and *xor* of crypts with same key $Th1_{sk}(i), Th2_{sk}(i)$ crypt $Th1_{crypt}(i), Th2_{crypt}(i)$ and *xor* of crypts with different keys $Th1_{dk}(i), Th2_{dk}(i)$ are established as:

$Th1_{plain}(i) = Th1_{sk}(i)$ = 0.002, 0.010, 0.010, 0.010, 0.010, 0.010, 0.0001, 0.0001, 0.0001, 0.0001

$Th2_{plain}(i) = Th2_{sk}(i)$ = 7.600, 7.900, 7.900, 7.900, 7.900, 7.900, 0.010, 0.010, 0.010, 0.010

$Th1_{crypt}(i) = Th1_{dk}(i)$ = 0.010, 0.100, 0.100, 0.100, 0.100, 0.100, 1.000, 1.000, 1.000, 1.000

$Th2_{crypt}(i) = Th2_{dk}(i)$ = 0.299, 0.050, 0.050, 0.050, 0.000, 0.050, 0.050, 0.050, 0.050, 0.050

5.1 Classification Results

Identification of plain/encrypted images and encrypted images with same/different keys:

Plain Images

(i) $P_{\mu}(i)$ = 6.6962 6.6299 6.4968 6.6610 6.4956 6.5708 0.0000 0.9999 0.9868 0.8728
$\mu P_{\mu}(crypt,i)$ = 0.0000 0.8210 0.0000 0.8318 0.0000 0.8204 0.0000 1.0000 0.0000 0.9839
$\mu P_{\mu}(plain,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.8021 1.0000 1.0000
S_{crypt} = 0.4457, S_{plain} = 0.9802; Class identified: Plain image

(ii) $P_{\mu}(i)$ = 7.6916 7.6756 7.4412 7.2327 7.0098 7.3398 0.9584 0.9285 1.0000 0.9845
$\mu P_{\mu}(crypt,i)$ = 0.0000 0.9586 0.0000 0.9041 0.0000 0.9177 0.0000 0.9909 0.9959 0.9980
$\mu P_{\mu}(plain,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9004 1.0000
S_{crypt} = 0.5765, S_{plain} = 0.9900; Class identified: Plain image

(iii) $P_{\mu}(i)$ = 0.0000 0.7240 0.0000 0.8235 0.0000 0.8028 0.0000 0.9587 0.9352 0.9941
$\mu P_{\mu}(crypt,i)$ = 0.0000 0.0744 0.0000 0.7636 0.0000 0.6466 0.0000 0.9587 0.9352 0.9941
$\mu P_{\mu}(plain,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9065 1.0000
S_{crypt} = 0.5238, S_{plain} = 0.9906; Class identified: Plain image

Encrypted Images

(i) $P_{\mu}(i)$ = 7.9976 7.9886 7.9880 7.9892 7.9886 7.9886 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(crypt,i)$ = 1.0000 0.9998 0.8033 0.9999 0.8581 0.9998 1.0000 1.0000 0.9994 1.0000
$\mu P_{\mu}(plain,i)$ = 0.0000 0.8709 0.6197 0.2150 0.6142 0.2276 0.9000 0.8001 0.9001 0.8001
S_{crypt} = 0.9660, S_{plain} = 0.5948; Class identified: Encrypted image

(ii) $P_{\mu}(i)$ = 7.9970 7.9887 7.9890 7.9899 7.9885 7.9890 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(crypt,i)$ = 1.0000 0.9998 0.9025 1.0000 0.8460 0.9999 0.9975 1.0000 0.9987 1.0000
$\mu P_{\mu}(plain,i)$ = 0.0000 0.8707 0.6098 0.2017 0.6154 0.2197 0.9003 0.8000 0.9001 0.8002
S_{crypt} = 0.9744, S_{plain} = 0.5918; Class identified: Encrypted image

(iii) $P_{\mu}(i)$ = 7.9972 7.9877 7.9900 7.9877 7.9893 7.9887 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(crypt,i)$ = 1.0000 0.9997 1.0000 0.9997 0.9281 0.9998 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(plain,i)$ = 0.0000 0.8741 0.6000 0.2457 0.6072 0.2267 0.9000 0.8001 0.9000 0.8001
S_{crypt} = 0.9927, S_{plain} = 0.5954; Class identified: Encrypted image

xor of Encrypted Images with Same/Equivalent Keys

(i) $P_{\mu}(i)$ = 7.9273 7.8743 7.9279 7.5852 7.5106 7.7245 0.9584 0.9978 0.9979 0.9943
$\mu P_{\mu}(dk,i)$ = 0.0000 0.9848 0.0000 0.9488 0.0000 0.9664 0.0000 0.9997 0.7882 0.9993
$\mu P_{\mu}(sk,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.8432 0.9212 0.9150
S_{dk} = 0.5687, S_{sk} = 0.9679; Class identified: Images encrypted with same keys

(ii) $P_{\mu}(i)$ = 7.9229 7.7117 7.3550 7.7251 7.8759 7.6669 0.9732 0.9956 0.9999 0.9961
$\mu P_{\mu}(dk,i)$ = 0.0000 0.9634 0.0000 0.9665 0.0000 0.9591 0.0000 0.9994 0.9908 0.9995
$\mu P_{\mu}(sk,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.8875 0.9009 0.8784
S_{dk} = 0.5879, S_{sk} = 0.9667; Class identified: Images encrypted with same keys

(iii) $P_{\mu}(i)$ = 7.9204 7.7745 7.6534 7.8665 7.8479 7.7856 0.9680 0.9698 0.9989 0.9919
$\mu P_{\mu}(dk,i)$ = 0.0000 0.9716 0.0000 0.9844 0.0000 0.9741 0.0000 0.9962 0.8926 0.9990
$\mu P_{\mu}(sk,i)$ = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9107 0.9613
S_{dk} = 0.5818, S_{sk} = 0.9872; Class identified: Images encrypted with same keys

xor of Encrypted Images with Different Keys

(i) $P_{\mu}(i)$ = 7.9974 7.9876 7.9900 7.9890 7.9881 7.9887 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(dk,i)$ = 1.0000 0.9997 1.0000 0.9999 0.8099 0.9998 0.9996 1.0000 1.0000 1.0000
$\mu P_{\mu}(sk,i)$ = 0.0000 0.8742 0.5995 0.2209 0.6190 0.2263 0.9000 0.8000 0.9000 0.8001
S_{dk} = 0.9809, S_{sk} = 0.5940; Class identified: Images encrypted with different keys

(ii) $P_{\mu}(i)$ = 7.9969 7.9895 7.9897 7.9875 7.9895 7.9891 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(dk,i)$ = 1.0000 0.9999 0.9713 0.9997 0.9505 0.9999 0.9999 1.0000 1.0000 1.0000
$\mu P_{\mu}(sk,i)$ = 0.0000 0.8677 0.6029 0.2499 0.6050 0.2187 0.9000 0.8001 0.9000 0.8002
S_{dk} = 0.9821, S_{sk} = 0.5944; Class identified: Images encrypted with different keys

(iii) $P_{\mu}(i)$ = 7.9970 7.9883 7.9889 7.9879 7.9884 7.9884 1.0000 1.0000 1.0000 1.0000
$\mu P_{\mu}(dk,i)$ = 1.0000 0.9998 0.8891 0.9997 0.8419 0.9998 0.9993 1.0000 0.9991 1.0000
$\mu P_{\mu}(sk,i)$ = 0.0000 0.8719 0.6111 0.2416 0.6158 0.2323 0.9001 0.8000 0.9001 0.8001
S_{dk} = 0.9729, S_{sk} = 0.5973; Class identified: Images encrypted with different keys

Above results show that the given images can be classified without ambiguity because the score values are well separable. Hence, the cryptographic data of image stream enciphering can be segregated successfully.

5.2 Performance

Performance of classification and segregation of unknown patterns will decrease if the values of features for reference patterns and thresholds for different fuzzy functions are not taken appropriately. Fuzzy membership functions are fixed with appropriate values based on the knowledge of variations studied once only for different images and not for every run to analyse traffic of cryptographic communications for identification of images encrypted with same keys. The values of scores are very high for correct classes and low for incorrect classes as can be seen from the above examples.

A plot showing score values for different classes as plain image, encrypted image, images encrypted with same keys, and images encrypted with different keys is given in Fig. 7 where higher score values (red colour) indicate for correct classes and lower score values (blue colour) indicate for incorrect class. It is seen from the plot that the class scores values are far apart from each other, higher score values for right class and lower class values for wrong class.

The performance of multi-entropy feature based class identification is compared with fuzzy bit-plane feature based method[52]. The results are shown in Table 3.

We see from Table 3 that the proposed multi-entropy based method performs better with high precision. According to Wilkinson test[53], the variance of class scores should be minimum near to zero. Moreover, both the mean and median of class scores should be approximately equal to each other. The mean score values are about 0.97 to 0.99 and variance is less to 0.00019 for correct classes for proposed method whereas these values are about 0.56 to 0.67 and less to 0037 respectively for other method[52]. The variation in class score values is lesser for encrypted images with different keys and higher for plain images and encrypted images with same keys. The higher variation in class scores for plain images and encrypted images with same keys is due to values of features vary within wider range. The proposed classification method performs better compared to SVM method[64] as the same has also been shown by other fuzzy classification methods[52, 60].

Table 3. Classification performance

Image class	Value of class scores for multi-entropy based method (proposed)			Value of class scores for bit-plane feature based method[52]		
	Mean value	Median value	Variance value	Mean value	Median value	Variance value
Plain image	0.9900	0.9900	0.000041	0.5625	0.5473	0.0038
Encrypted image	0.9791	0.9802	0.000169	0.6763	0.6748	0.00035
Image encrypted with same keys	0.9755	0.9728	0.000190	0.5616	0.5441	0.0037
Image encrypted with different keys	0.9732	0.9738	0.000147	0.6731	0.6712	0.00037

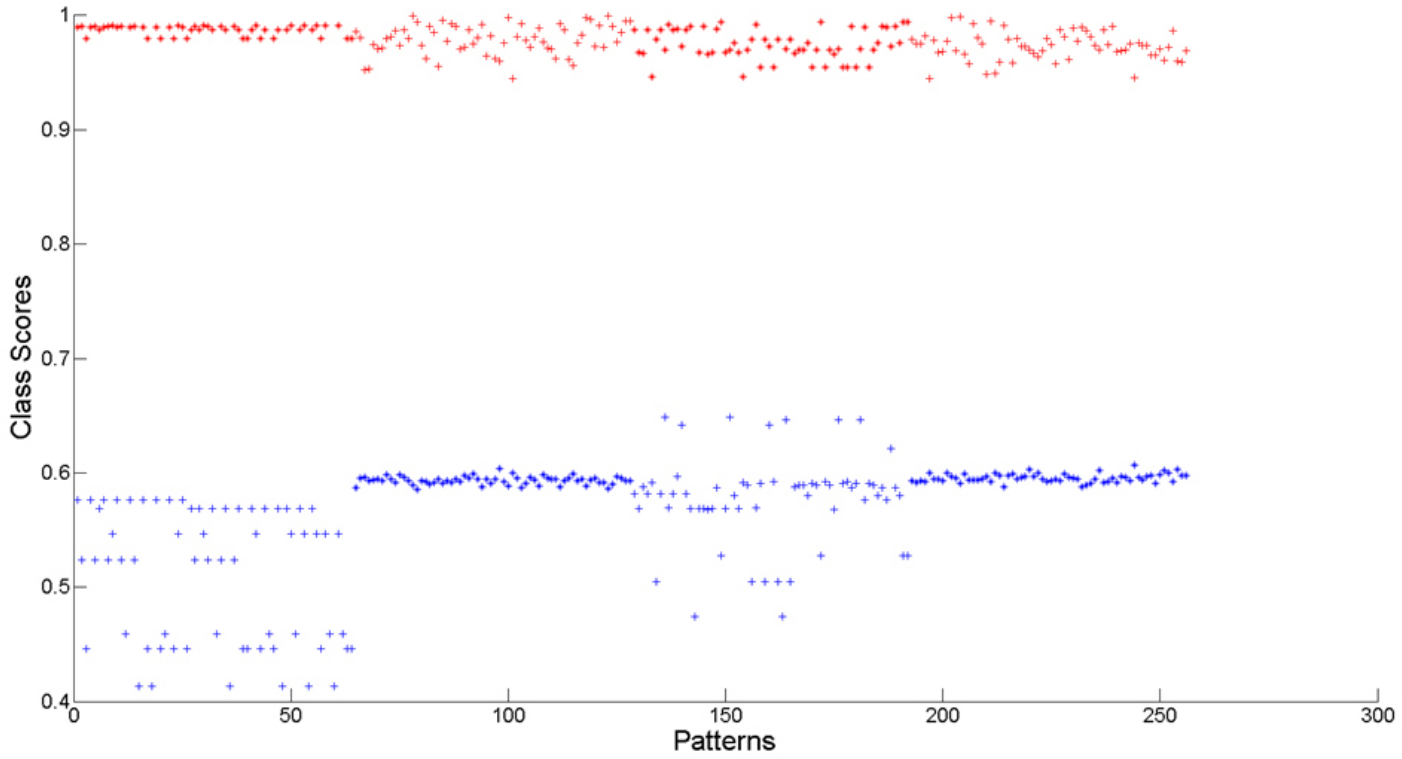

Figure 7. Plot for identification of images with proposed method as encrypted with plain/encrypted and same/different keys

6. COUNTERMEASURES AGAINST TMSK ATTACK

A cipher system should use a unique key generated through a random number generator (RNG) module for every message to be encrypted. RNG module is a random source can be based on software and hardware. It may be a deterministic or nondeterministic. A deterministic RNG module can regenerate the same output bits by giving same seed and it is called as pseudo random number generator (PRNG). A nondeterministic RNG module cannot regenerate the same output sequence, i.e., it gives different output bits for every instance and it is called as Real Random Number Generator (RRNG) or true random number generator (TRNG). A cipher system uses a key of fixed size and it may vary for system-to-system depending on its design. After exhausting all possible keys, these will start repeating irrespective of the design mechanism of TRNG/ PRNG module. So, a cipher system should have the mechanism and procedure to initialize the encryption algorithm uniquely even after repetition of such keys. The PRNG/RRNG module of a cipher system should be checked statistically to meet cryptographic properties[65-68]. Also, if key sequences generated for different keys are equal or approximately equal then the keys are considered as the equivalent keys. The use of equivalent keys is also not advisable as these can be attacked. Some of the measures which prevent the applicability of TMSK attack are mentioned in the following paragraphs:

- Avoid the repetition of keys generated by RNG. The non-repetition of keys can be ensured by measuring the maximal order complexity (MOC) for a number of output bit streams of random source (RNG) to be used for key generation. For a binary sequence of length *N*, the value of MOC[68] is given by $2log_2 N$. The MOC is the maximum length of a binary pattern which repeats in a sequence. The value of MOC should be larger to the size of key for any RNG output sequence to ensure the non-repetition of same key. If the length of the key is L bits, the MOC of output sequence of RNG should be larger than L . For key length of L bits, there are 2^L number of possible keys.
- Initial vector (IV) used to initialize cryptographic algorithms should be derived uniquely to avoid the repetition of same key sequence output[69-71]. The IV generation module also known as key scheduler module should be non-linear, one-way secure mechanism to obtain IV bits. These IV bits should not be taken directly the RNG output. The IV generation module should be obtained in a non-linear manner and it should provide unique IV for every message encryption.
- Both the key and IV bits should meet the key avalanche criteria, i.e., for any keys and IVs, the sequences generated should be drastically different[72] and the possibility of related keys or equivalent keys should not occur. According to avalanche criteria, a small change in input bits to the cipher system should make a drastic change (50% dissimilarity) randomly in the output key sequence bits. The avalanche criteria should meet for all the cases, i.e., key to IV, key to key sequence and IV to key sequence.
- Situation of an equivalent key may appear due to an inappropriate generation and use of key bits and IV bits. There should not be any such weakness in the utilisation of key bits in key scheduler and IV bits in initialising cryptographic algorithm. All the IV bits should be uniquely utilised none of the bits left unused or redundant in crypto algorithm initialisation.

- To meet the security requirements for an intended purpose, key length should be large enough to prevent non-repetition of keys. The PRNG or RRNG to be considered for generating non-repetitive keys of length L and it should pass randomness tests to assure getting key random and unrelated at any instance. For key length of L, the key diversity is 2^L and the keys will start repeating after 2^L number of keys used.
- Cryptographic keys should never be kept alongside with any other data accessible by customers. These keys should be kept separately and securely with adequate protection and not accessible by any unintended customer. It is preferable to keep these keys in encrypted form and should be made available in unencrypted from in secure and tamper protected environment. Unsafe and unprotected storage of keys makes vulnerable to expose and get compromised.
- Cryptographic keys must be destroyed after its expiry to avoid its accidental reuse and repetition. Keys which are not destroyed after expiry may be reused unintentionally. The reuse of keys should never be allowed to avoid applicability of TMSK attack.
- Use of default or pre-set keys if any, should not be allowed for use in encrypting messages. These keys will get repeating due to its frequent usage and such situations need to be avoided in cryptographic communications. To avoid such situations, the provision of such keys should not be provided in the systems.
- Cipher system should have the provision to maintain audit log and alarming indicator to show the compromise of keys.
- Customer should take care to manage keys of ciphers to avoid repetition of same keys[23-26]. The cryptographic keys should be changed in the system well in time to avoid the situation of applicability of TMSK attack.
- If keys are not changed in time as per requirements, the cipher system should have the provision to stop the communication. The cipher system should have the alarming indicator enabling change of keys timely.
- The aim of a cipher system is to provide security to the message to be communicated. If both the provisions plain and crypt modes are required in a cipher system for communication, then the plain mode should not by default because it may communicate a vital information in plain form unintentionally which is not desirable. Crypt mode should work by default. To avoid unintentional communication of vital messages in plain mode, the cipher system should not have the provision of plain mode and it should be removed permanently.
- Always follow the best practices of key management[23-26] to avoid the situations of key reuse, key leakage, and key compromised. If there is any weak point in key management, it should be resolved adequately to mitigate the risks of misuse of cryptographic keys.

Any issue weakening key generation module, initialisation of encryption algorithm, design of cryptographic algorithm, and key management should be resolved appropriately by following appropriate countermeasures.

7. CONCLUSIONS

A problem of traffic analysis of image cryptographic communications of stream cipher has been attempted using multi-entropy measures and fuzzy soft decision approach for classifying unknown images as plain image, encrypted image, images encrypted with same/equivalent key and images encrypted with different keys. The global entropy, sub-block entropy and bit-plane entropy of images have been applied as features in identification methodology. An asymmetric triangular kind of fuzzy membership functions and fuzzy similarity based decision criteria have been used to identify the class. Experimental results show that one can analyse the traffic of image cryptographic communications successfully with high precision performance. Some countermeasures have also been presented to take care in the design of cipher systems, its operations and key management to avoid the repetition of keys, the formation of multiple crypts with same encryption keys, and hence the safeguarding of image cipher from cryptanalytic TMSK attack. The countermeasures suggested can be considered in deciding cipher systems suitably for secure message communications. Methodology presented seems very useful and can be applied to analyse adversary's communications for extracting meaningful information. Moreover, it can also be used in image pattern recognition.

REFERENCES

1. Menezes, A.;Vanstone, S. & Van Oorschot, P. Handbook of Applied Cryptography. CRC Press, Boca Raton, 1996.
2. Katzenbeisser, S. & Petitcolas, F.A.P. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Boston, 2000.
   doi : 10.1201/1079/43263.28.6.20001201/30373.5
3. Ratan, R. & Veni Madhavan, C.E. Steganography based information security. *IETE Tech Rev*, 2002, **19**(4), 213-19.
   doi : 10.1080/02564602.2002.11417034
4. Stinson, D.R. Decomposition constructions for secret sharing schemes. *IEEE Trans. Inform Theory*, 1994, **40**(1), 118-25.
   doi : 10.1109/18.272461
5. Shamir, A. How to share a secret. *Commun ACM*, 1979, **22**(11), 612-613.
   doi : 10.1145/359168.359176
6. Simon, M.; Omura, J.; Scholtz, R. & Levitt, B. Spread spectrum communications handbook. McGraw-Hill, New York, 2002.
7. Klein, A. Stream ciphers. Springer, London, 2013.
   doi : 10.1007/978-1-4471-5079-4
8. Rueppel, R.A. Analysis and Design of Stream Ciphers. Springer, Berlin, 1986.
   doi : 10.1007/978-3-642-82865-2
9. Barker, E. & Kelsey, J. Recommendation of random number generation using deterministic random bit generators. NIST SP800-90A, 2012.
   doi : 10.6028/NIST.SP.800-90a
10. Maciej, P.; Paweł, D. & Ryszard, S. Pseudo-random bit generators based on linear-feedback shift registers in a programmable device. *Measurement Automation*

Monitoring, **06**(62), 2016, 184-86.
11. NIST: NIST Random number generation and testing [OL] (2011). http://csrc.nist.gov/rng.
12. Dodis, Y.; Ostrovsky, R.; Reyzin, L. & Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 2008, **38**(1) 97-139.
doi : 10.1137/060651380
13. Ratan, R. Applications of genetic algorithms in cryptology. *In* Proceedings of the SocProS 2013, AISC, Springer, 2014, **258**, pp. 821-31.
doi : 10.1007/978-81-322-1771-8_71
14. Kenny, C. & Mosurski, K. Random number generators: an evaluation and comparison of random.org and some commonly used generators. 2005. https://www.random.org/analysis/Analysis 2005.pdf.
15. Cusick, T. & Stanica, P. Cryptographic boolean functions and applications. academic press, Elsevier, USA, 2017.
doi : 10.1016/B978-0-12-811129-1.00005-5
16. Asthana, R.; Verma, N. & Ratan, R. Generation of Boolean functions using genetic algorithm for cryptographic applications. *In* Proceedings of the IEEE International Advance Computing, 2014, pp. 1361-66.
doi : 10.1109/IAdCC.2014.6779525
17. Patro, K.A.K. & Acharya, B. Secure multi–level permutation operation based multiple color image encryption. *J. Inf. Secur. Appl.*, 2018, **40**, 111-33.
doi : 10.1016/j.jisa.2018.03.006
18. Wang, X.; Wang, S.; Wei, N. & Zhang, Y. A novel chaotic image encryption scheme based on hash function and cyclic shift. *IETE T Rev*, 2018, **36**(1), 39-48.
doi : 10.1080/02564602.2017.1393352
19. Patro, K.A.K.; Acharya; B. & Nath, V. A secure multi-stage one-round bit-plane permutation operation based chaotic image encryption. *Microsyst Technol*, 2019, **25**(6), 2331-38.
doi : 10.1007/s00542-018-4121-x
20. Tong, X. & Cui, M. Image encryption with compound chaotic sequence cipher shifting dynamically. *Image and Vision Computing*, 2008, **26**(6), 843-50.
doi : 10.1016/j.imavis.2007.09.005
21. Ratan, R. Securing images using inversion and shifting, *In* Proceedings of the SocProS 2011, AISC, Springer, 2012, **131**, pp. 401-12.
doi : 10.1007/978-81-322-0491-6_38
22. Askar, S.S.; Karawia, A.A. & Alammar, F.S. Cryptographic algorithm based on pixel shuffling and dynamical chaotic economic map. *IET Image Process*, 2018, **12**(1), 158-67.
doi : 10.1049/iet-ipr.2016.0906
23. Yashaswini, J. Key management for symmetric key cryptography. *Int. J. Innovative Res. Comput. Commun. Eng.*, 2015, **3**(5), 4328-31.
24. Lloyd, S. & Adams, C. Key management. *In* Encyclopedia of Cryptography and Security. Springer, Boston, 2011.
doi : 10.1007/978-1-4419-5906-5_85
25. Barker, E. & Barker, W. Recommendation for key management: Part 2: Best practices for key management organizations, Computer Security. NIST Special Publication (SP) 800-57 Part 2, 2019.
doi : 10.6028/NIST.SP.800-57pt2r1
26. Stubbs, R. Cryptographic key management - the risks and mitigation, 2018. https://www.cryptomathic.com/news-events/blog/cryptographic-key-management-the-risks-and-mitigations.
27. Li, C.; Lin, D. & Lü, J. Cryptanalyzing an image-scrambling encryption algorithm of pixel bits, *IEEE Multimedia*, 2017, **3**(3), 64-71.
doi : 10.1109/MMUL.2017.3051512
28. Ratan, R. Key independent decryption of graphically encrypted images. *In* Proceedings of the PAISI 2010, LNCS, Springer, 2010, **6122**, pp. 88-97.
doi : 10.1007/978-3-642-13601-6_11
29. Ratan, R. Key independent retrieval of chaotic encrypted images. *In* Proceedings of the PReMI 2009, LNCS, Springer, 2009, **5909**, pp. 483-88.
doi : 10.1007/978-3-642-11164-8_78
30. Ge, X.; Lu, B.; Liu, F. & Luo, X. Cryptanalyzing an image encryption algorithm with compound chaotic stream cipher based on perturbation. *Nonlinear Dynam*, 2017, **90**(2), 1141-50.
doi : 10.1007/s11071-017-3715-7
31. Li, C.; Lin, D.; Lü J. & Hao, F. Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography, *IEEE MultiMedia*, 2018, **25**(4), 46-56.
doi : 10.1109/MMUL.2018.2873472
32. Li, C.; Zhang, Y. & Xie E.Y. When an attacker meets a cipher-image in 2018: A year in review. *J. Inf. Secur. Appl.*, 2019, **48**.
doi: 10.1016/j.jisa.2019.102361
33. David, K. The Codebreakers. Scribner, New York 1996.
34. Nagy, G.; Seth, S. & Einspahr, K. Decoding substitution ciphers by means of word matching with application to OCR. *IEEE T Pattern Anal*, 1987, **9**(5), 710-15.
doi : 10.1109/TPAMI.1987.4767969
35. Bahramali, A.; Houmansadr, A.; Soltani, R.; Goeckel, D. & Towsley, D. Practical traffic analysis attacks on secure messaging. *In* the Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020, doi: 10.14722/ndss.2020.24347.
36. Dainotti, A.; Pescape, A. & Claffy, K.C. Issues and future directions in traffic classification. *IEEE Network*, 2012, **26**(1), 35-40.
doi : 10.1109/MNET.2012.6135854
37. Best practices in network traffic analysis: three perspectives. 2018 https://insights.sei.cmu.edu/sei_blog/2018/10/best-practices-in-network-traffic-analysis-three-perspectives.html.
38. Jiang, D.; Huo, L.; Lv, Z.; Song, H. & Qin, W. A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. *IEEE Trans. Intell. Transp.*, 2018, **19**(10), 3305-19.
doi : 10.1109/TITS.2017.2778939
39. Qi, S.; Jiang, D. & Huo, L. A prediction approach to end-to-end traffic in space information networks. Mobile Netw Appl, 2019.

doi : 10.1007/s11036-019-01424-2
40. Sadkhan, S.B. & Abbas, N.A. Watermarked and noisy images identification based on statistical evaluation parameters. *J. Zankoy Sulaimani- Part A (JZS-A)*, 2013, **15**(3), 159-68.
doi : 10.17656/jzs.10265
41. Din, M.; Ratan, R.; Bhateja, A.K. & Bhateja, A. Multimedia classification using ANN approach. *In* Proceedings of the SocProS 2012, AISC, Springer, 2014, **236**, pp. 905-10.
doi : 10.1007/978-81-322-1602-5_96
42. Renu; Ravi & Ratan, R. Live traffic english text monitoring using fuzzy approach. *In* Proceedings of the SocProS 2012, AISC, Springer, 2014, **236**, pp. 911-18.
doi : 10.1007/978-81-322-1602-5_97
43. Asthana, R.; Sharma, A.; Ratan, R. & Verma, N. Classification of error-correcting coded data using multidimensional feature vectors. *In* Proceedings of the SocProS 2014, AISC, Springer, 2015, **336**, pp. 303-12.
doi : 10.1007/978-81-322-2220-0_24
44. Teimouri, M. & Motlagh, H.K. Reverse engineering of communication networks. 2017, arXiv:1704.05432 [cs. IT].
45. Ray, P.K.; Kant, S.; Roy, B.K. & Basu, A. Classification of encryption algorithms using Fisher's discriminant analysis. *Def. Sci J*, 2017, **67**(1), 59-65.
doi : 10.14429/dsj.1.9153
46. Kant, S. Models for symmetric key cryptosystem identification. *Defence Sci J*, 2012, **62**(1), pp. 38-45.
doi : 10.14429/dsj.62.1440
47. Bansal, J.C. & Pal, N.R. Swarm and evolutionary computation. *In* Evolutionary and Swarm Intelligence Algorithms. Studies in Computational Intelligence, Springer, 2019, **779**, pp. 1-9.
doi : 10.1007/978-3-319-91341-4_1
48. Bansal, J.C.; Singh, P.K. & Pal, N.R. Evolutionary and swarm intelligence algorithms. Studies in Computational Intelligence, Springer, **779**, Cham, Switzerland, 2019.
doi : 10.1007/978-3-319-91341-4
49. Zadeh, L.A. Fuzzy sets. *Inform. Control*, 1965, **8**(3), 338-52.
doi : 10.1016/S0019-9958(65)90241-X
50. Bezdek, J.C. & Pal, S.K. Fuzzy Models for Pattern Recognition. IEEE Press, New York, 1992.
51. Bezdek, J.C.; Keller, J.; Krisnapuram, R. & Pal, N.R. Fuzzy models and algorithms for pattern recognition and image processing. Springer, New York, 2005.
52. Arvind & Ratan, R. Identifying traffic of same keys in cryptographic communications using fuzzy decision criteria and bit-plane measures. *Int. J. Syst. Assur. Eng. Manag.*, 2020, **11**(2), 466-80.
doi : 10.1007/s13198-019-00878-7
53. McCullough, Wilkinson's test and econometric software, *J. Econ. Soc. Meas.*, 2004, **29**, 261-70.
doi : 10.3233/JEM-2004-0199
54. Huber, P. & Ronchetti, E. Robust statistics. Wiley, NY, 2009.
doi : 10.1002/9780470434697
55. Haddon, J.F. & Boyce, J.F. Co-occurrence matrices for image analysis. IEE *Electron Commun. Eng.*, 1993, **5**(2), 71-83.
doi : 10.1049/ecej:19930013
56. Ratan, R. & Arvind. Bit-plane specific measures and its applications in analysis of image ciphers. *In* Proceedings of the SIRS 2018, CCIS, Springer, 2019, **968**, pp. 282-97.
doi : 10.1007/978-981-13-5758-9_24
57. Shannon, C.E. Communication theory of secrecy systems, *Bell Syst Tech J*, 1949, **28**(4), pp. 656-715.
doi : 10.1002/j.1538-7305.1949.tb00928.x
58. Arora, P. On the Shannon measure of entropy, *Inform Sciences*, 1981, **23**, 1-9.
doi : 10.1016/0020-0255(81)90036-0
59. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P. & Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inform Sciences*, 2013, **222**, 323-42.
doi : 10.1016/j.ins.2012.07.049
60. Prasad, M.; Li, D.L.; Lin, C.T.; Prakash, S.; Singh, J. & Joshi, S. Designing Mamdani-type fuzzy reasoning for visualizing prediction problems based on collaborative fuzzy clustering. *IAENG Int. J. Comput. Sci.*, 2015, **42**(4), 404-11.
61. Puri, S. & Kaushik, S. A technical study and analysis on fuzzy similarity based models for text classification. *Int J Data Min Knowl*, 2012, **2**(2), 1-15.
doi : 10.5121/ijdkp.2012.2201
62. Beker, H. & Piper, F. Cipher Systems: The Protection of Communications. Northwood Books, London, 1983.
63. Kerckhoffs, A. La cryptographhie militaire, *Journal des Sciences Militaires*. 1883, IX, 161-91.
64. Keerthi, S.S.; Chapelle, O. & DeCoste, D. Building support vector machines with reduced classifier complexity. *J. Mach. Learn Res.*, 2006, **7**, 1493-1515.
65. Rukhin, A.; Soto, J.; Nechvantal, J.; Smid, M.; Barker, E.; Leigh, E.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; Dray, J. & Vo, S. A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications. Ver. STS2.1, NIST SP 800-22rev1a, 2010. http://csrc.nist.gov//publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.
66. Brown, R.G. Dieharder: A random number test suite. Ver. 3.31.1, 2004.
67. Ratan, R.; Jangid, B.L. & Arvind. Bit-plane specific randomness testing for statistical analysis of ciphers. *In* Proceedings of the SocProS 2019, AISC, Springer, 2020, **1138**, pp. 199-213.
doi: 10.1007/978-981-15-3290-0_16
68. Jansen, Cees J.A. The maximum order complexity of sequence ensembles, *In* Proceedings of the EUROCRYPT'91, Advances in Cryptology, LNCS, Springer, 1991, **547**, pp. 153-59.
doi : 10.1007/3-540-46416-6_13
69. Fluhrer, S.; Mantin, I. & Shamir, A. Weaknesses in the key scheduling algorithm of RC4. *In* Proceedings of the Selected Areas in Cryptography (SAC 2001), LNCS,

Springer, 2001, **2259**, pp. 1-24.
doi : 10.1007/3-540-45537-X_1

70. Berbain, C. & Gilbert, H. On the security of IV dependent stream ciphers. *In* Proceedings of the International Workshop on Fast Software Encryption (FSE 2007), LNCS, Springer, 2007, **4593**, pp. 254-273.
doi : 10.1007/978-3-540-74619-5_17
71. Pirzada, S. J. H.; Murtaza, A.; Xu, T. & Jianwei, L. Initialization vector generation for AES-CTR algorithm to increase cipher-text randomness. *In* Proceedings of the 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2019, pp. 138-42.
doi : 10.1109/ICISCAE48440.2019.221605
72. Mishra, P.R.; Gupta, I. & Pillai, N.R. Generalized avalanche test for stream cipher analysis. *In* Proceedings of the InfoSecHiComNet 2011, LNCS, Springer, 2011, **7011**, pp. 168-80.
doi : 10.1007/978-3-642-24586-2_15

CONTRIBUTORS

Mr Ram Ratan is a Scientist in the DRDO-Scientific Analysis Group, Delhi. Currently, he is working in the area of information security. His research area includes cryptography, image processing and pattern recognition.
In the current study, he has proposed classification methodology using multi-entropy measures to identify images encrypted with same keys.

Dr Arvind Yadav is an Assistant Professor in the Hansraj College, University of Delhi. He has also worked as Scientist in DRDO-Scientific Analysis Group, Delhi. His research area includes mathematics, cryptography, information security and image processing.
In the current study, he has compared the classification performance of proposed method using Wilkinson's test.