DATA HIDING IN A CLASSICAL IMAGE USING SCALABLE IMAGE ENCRYPTION

T. Vijaya Kumar[1], P. Sri Harsha[2], M. Sai Sri Ram[3], V. Sri Sandilya[4], P. Mahesh Babu[5], T. Vamsi Krishna[6]

[1-6]*Dept of ECE, Vasireddy Venkatadri Institute of Technology, Guntur, India*

ABSTRACT:

Data hiding is the process of hiding data into data sources such as audio, video or image signals with no change in the actual quality of the signal. The most commonly used technique for hiding data is the Least Significant Bit and use of redundancy in the cover image by performing loss less compression. The encryption of data or signal is a general method for providing privacy protection such as confidentiality of data. As on encryption phase, initially the images are encrypted with the encryption key and at decryption phase the image is restored back with the help of that encrypted key.

The image cannot be decrypted back if there is no encrypted key or if there is a wrong encrypted key. For the encryption process to happen we have implemented a light weight encryption algorithm known as SIT: Secure IoT. Since the traditional algorithms are complex and time taking algorithms, they take more computational time and also exhaust the system power for encryption. However, less complex algorithm may compromise for the desired integrity. For the purpose of hiding data in an image, LSB method of hiding is used.

I.INTRODUCTION

Cryptography is the science of writing a hidden code (or) the study of mathematical techniques related to aspects of data security such as the integrity of data, confidentiality, entity authentication and origin authentication of data. The basic cryptographic techniques used are Symmetric (or) Hidden key cryptography and Asymmetric (or) Public key cryptography. It employs complex computational algorithms for encryption and decryption. In order to reduce the complexity, this cryptography ideas can be used. The fast increase of data exchange in electronic way, data security is becoming important in the storage and transmission of data. The wide usage of images in industrial process, made it important to protect the confidential image data from unauthorized access.

Security is an important issue in storage and communication of images, and encryption is one of the ways to ensure security of images. Image encryption is different from text encryption.

Although, there are many traditional cryptosystems to encrypt images directly, it is not seen to be a good idea for some reasons. One is that the image size is almost always greater Sthan the size of text. Therefore, the traditional cryptosystems need much time to directly encrypt the image data. The other one is that the decrypted image must be equal in size and content to the original image. In order to transmit hidden images from one place to another, a huge variety of image encryption techniques have been proposed. Since, the security of data transmission is very much important in the modern computerized and interconnected world, security problems such as modification, forgery, duplication, and others, on the Internet have been focused on very carefully. In this paper the proposed algorithm is based on a light weight image encryption technique known as SIT: Secure IoT, used for the encryption of images.

This is a light weight technique because the number of rounds of confusion and diffusion are very less compared to those of other traditional encryption techniques. In this paper section II deals with the key expansion algorithms, section III deals with the encryption algorithm, section IV deals with LSB method, section V deals with results and section VI deals with future work.

II. KEY EXPANSION

The most fundamental component in the processes of encryption and decryption of text or the image is the key. It is this key on which entire security of the data depends upon, if this key is known to an attacker, the privacy that we intend to maintain for data will be lost. So, necessary steps must be taken in order to make the revelation of the key as difficult as possible. Among many of the algorithms available for the process of encryption, feistel and substitution algorithms are most prominent one's. Feistel based algorithms for the process of encryption are based on many rounds of confusion and diffusion, in which each of the rounds require a separate key. The encryption or decryption of the proposed algorithm is composed of fifive rounds of confusion and diffusion, so that we require fifive unique keys for the encryption and decryption process. We have introduced a key expansion block in this section for clear understanding of key generation process. To maintain the security against the attacks from the hackers, the length of the is suggested to be long, let the length of the true key 'k' be large so2 that it becomes beyond the capability of the enemy or hacker to perform '((2k)−1)' encryptions and decryptions for key searching attacks. The proposed algorithm for the process of encryption of image is a 64-bit block cipher, which means it requires a 64-bit key for the encryption of 64-bits of data. So that, a cipher key '(Kc)', 64- bits in length is taken from the user, which shall be used as input bits to the expansion block in the image encryption process. The key expansion block, upon performing many operations on key will generate fifive unique keys at the end, of which four are generated by some substantial operations and the fifth one is through the XOR operation of the above generated keys. These would be used in the process of encryption and decryption and are strong enough to remain indistinct during attacks from the hackers and enemies. The key expansion block is as shown in Fig. 1.

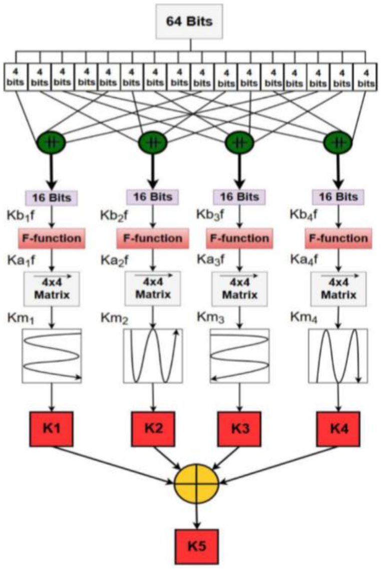


Fig 1. Key Expansion

Detailed explanation about the process of key expansion is discussed below:

- In the fifirst step of key expansion process, the 64-bit cipher key (Kc) is divided into small segments or blocks of 4-bits each.
- In the second step of key expansion, the 4-bit divided segments are clustered into 16-bit blocks as first four 4-bit segments into one cluster and next four into second cluster and so on to achieve 4 clusters each of 16-bits.
- In the third step of expansion process, the first 4-bits of each 16-bit clusters and confused into one block and second 4-bits into another and so on to

form four 16-bit blocks which are fed to an f-function.

- Mathematically, the input for f-function block are obtained by doing a substitution of small 4-bit segments of key '(Kc)' as shown in below equation.

$$Kb_i f = \|_{j=1}^{4} Kc_{4(j-1)+i}$$

Where, i = 1 to 4 for fifirst 4 round keys.

- The next step of operation is to get the term 'Kaif' by passing the 16-bits of 'Kbif' obtained by performing some mathematical operations above, to the f-function as shown in equation below.

$$Ka_i f = f(Kb_i f)$$

•The f-function consists of two tables namely P table and Q table. These tables of f-function are used to perform some of the linear and non-linear transformations which result in some confusions and diffusions of data given to them as shown in Fig. 2.

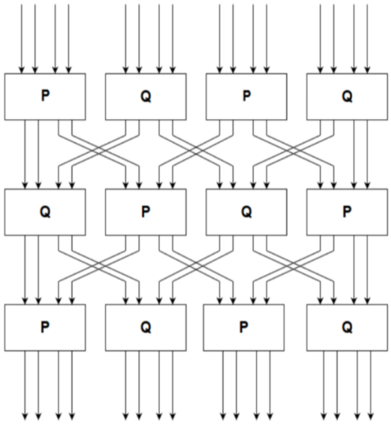
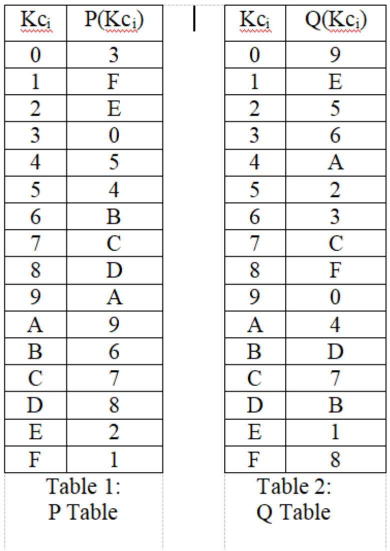


Fig. 2: F-Function

- The P table can handle 4-bits at a time and Q table can also handle 4-bits.

- The transformations made by P and Q tables are shown in the



Kc_i	$P(Kc_i)$
0	3
1	F
2	E
3	0
4	5
5	4
6	B
7	C
8	D
9	A
A	9
B	6
C	7
D	8
E	2
F	1

Table 1:
P Table

Kc_i	$Q(Kc_i)$
0	9
1	E
2	5
3	6
4	A
5	2
6	3
7	C
8	F
9	0
A	4
B	D
C	7
D	B
E	1
F	8

Table 2:
Q Table

Tables 1and 2.

- In the next step, the 16-bit output of each f-function block is

arranged in 4×4 matrix.

- The final step of key expansion process is the formation of round

keys K1, K2, K3 and K4 and obtaining the fifth key K5 from these four keys by performing the XOR operation among them. For this process, the matrices are first transformed into the arrays of 16 bits to obtain round keys (Kr). The arrangement of these bits for the generation of round keys is shown in equations below.

Q Table3

K1 = a4 ++ a3 ++ a2 ++ a1 ++ a5 ++ a6 ++ a7 ++ a8 ++ a12 ++ a11

++ a10 ++ a9 ++ a13 ++ a14 ++ a15 ++ a16.

K2 = b1 ++ b5 ++ b9 ++ b13 ++ b14 ++ b10 ++ b6 ++ b2 ++ b3 ++

b7 ++ b11 ++ b15 ++ b16 ++ b12 ++ b8 ++ b4.

K3 = c1 ++ c2 ++ c3 ++ c4 ++ c8 ++ c7 ++ c6 ++ c5 ++ c9 ++ c10

++ c11 ++ c12 ++ c16 ++ c15 ++ c14 ++ c13.

K4 = d13 ++ d9 ++ d5 ++ d1 ++ d2 ++ d6 ++ d10 ++ d14 ++ d15

++ d11 ++ d7 ++ d3 ++ d4 ++ d8 ++ d12 ++ d16.

- An XOR operation is performed among the four round keys to

obtain the fififth key for encryption process and the expression used

for that process is shown in equation below.

$$K5 = \bigoplus_{i=1}^{4} Ki$$

III. ENCRYPTION

After completing the generation of round keys from k1 to k4, the process of encryption can be started. For creating confusion and diffusion in encryption process, the data should undergo some logical operations such as right shifting, left shifting, swapping and substitution. The process of encryption is shown in Fig. 3. In the fifirst step of the encryption process, an array of 64-bit plain text '(Pt)' that

is to be encrypted is seperated into four small segments each of 16- bits size, such as Px0−15, Px16−31, Px32−47 and Px48−63. As the data progresses, in each round of the encryption process, swapping operation is performed so as to decrease the originality of data that should be encrypted by changing the order of bits, resulting in the increase of confusion in text. In the second round of encryption process, XNOR operation is performed among the respective round key 'Ki' obtained from the earlier process and Px0−15 and the same is done among 'Ki' and Px48−63 resulting in 16-bit data outputs as Ro11 and Ro14 respectively. The XNOR output is then given to the f- function block for generating the result as 'Efl1' and 'Efr1' respectively for Ro11 and Ro14 as shown in Fig. 3.

The f-function block which is used in the process of encryption is same as that of the f-function which is being used in the key expansion, comprised of confusion and diffusion operations, having the capability of handling 16-bits of data. Now, bitwise XOR function is applied between 'Efl1' & Px32−47 to obtain Ro12 and 'Efr1' & Px16−31 to obtain Ro13 each of 16-bits in length. Finally, some swapping operation is performed on data in such a way that for the next round of encryption process, Ro11 will become Px16−31, Ro12 will become Px0−15, Ro13 will become Px48−63 and Ro13 will become Px32−47 as shown in Fig. 3.

The process will be repeated for the remaining rounds of encryption process using equation below.

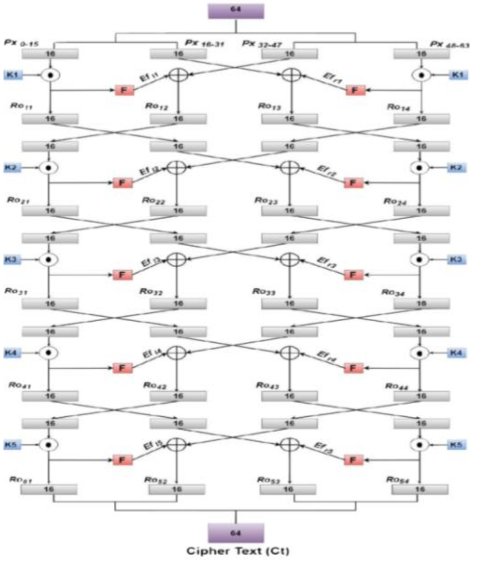


Fig 3. Process of obtaining of cipher text

The results of final round of encryption process are concatenated to obtain Cipher Text (Ct) as shown in equation below.

Ct = R51 ++ R52 ++ R53 ++ R54.

IV. LSB METHOD

This method is used for hiding the data in images. In this method of data hiding, gray scale images are used as cover images. Cover-images in which messages to be hidden are called stego- images. Usually for hiding, the quality of stego-image is treated as image quality. One of the most prominent and common technique used for hiding the data into the image is executed by replacing LSB bits of cover image by message bits. Here, the LSB planes refer to Least Significant Bits in each of the 8-bit segments generated by partitioning the binary cover. This method can achieve high capacity of operation. This allows hiding data in image. Generally, the LSB method of data hiding makes no change in the size of file or image after the completion of hidden process also, but basing on the size of the data that is to be hidden inside the image, the size of the file or image can be distorted noticeably.

V. RESULTS

The simulation of the algorithm for image encryption and decryption process is done to perform the standard tests includin4 avalanche, image entropy and image histogram on Intel Core i5 processor using MATLAB R2014. The process includes key expansions along with encryption and decryption of data. The Avalanche test proves that, for a single bit change in key generated through key expansion process brings a change of around 49% in the cipher bits, and that is near to an idealistic change of 50%. The results in Fig. 4 and Fig. 5 shows that the precise decryption is very much possible and the actual data is obtained only if the key which has to be used is used in the process of decryption, else the image remains non recognizable or some unknown symbols are seen on the output. For a visual conformation of avalanche test, the wrong key which is just a bit different from the original key is given at the decryption stage which resulted in some unknown data as the output, hence the strength of the algorithm can be noticed from this result. To perform entropy and histogram tests we have chosen some popular 8-bits grey scale images. The results of histogram in Fig. 6 for the original image and the encrypted image, and also the uniform distribution of intensities after the encryption can be the further indication of desired security provided by the proposed algorithm. Generally, an 8-bit grey scale image can achieve a maximum entropy of 8 bits. From the results in table 3, it can be seen that the entropy of all encrypted images is close to that of maximum value of 8.

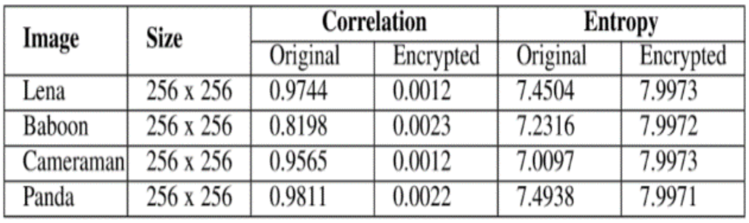


Image	Size	Correlation		Entropy	
		Original	Encrypted	Original	Encrypted
Lena	256 x 256	0.9744	0.0012	7.4504	7.9973
Baboon	256 x 256	0.8198	0.0023	7.2316	7.9972
Cameraman	256 x 256	0.9565	0.0012	7.0097	7.9973
Panda	256 x 256	0.9811	0.0022	7.4938	7.9971

Table 3: Correlation and Entropy

Finally, the correlation comparison in Fig. 7 shows the difference between the original data and the encrypted data. Original data, which is an image in our case can be seen to be highly correlated and detaining a high value of around '1' for correlation coeffificient. Whereas the encrypted image does not seem to have any correlation. The values of correlation are also shown in table 3. The results for the LSB method of hiding the data into an image are shown in Fig. 8, which is the un known or encrypted data with the text hidden embedded into it, the image or the text can be known only after the decryption is done completely.

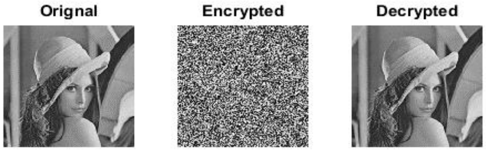


Fig. 4: The decrypted data using the original key.

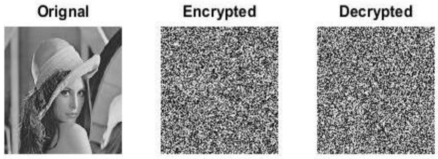


Fig. 5: The decrypted data using wrong key.

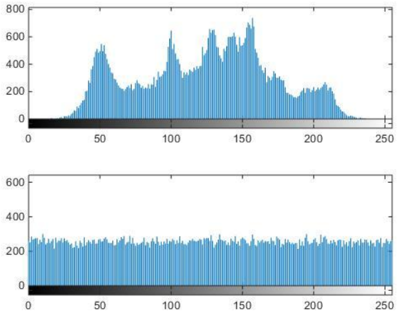


Fig 6. Histogram of the image

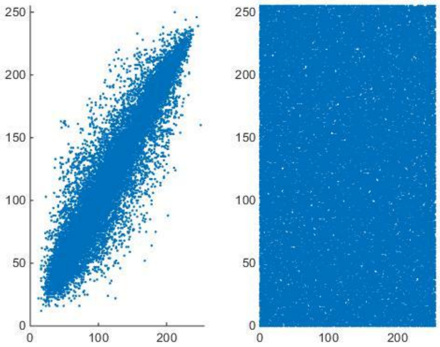


Fig 7: Correlation of image.

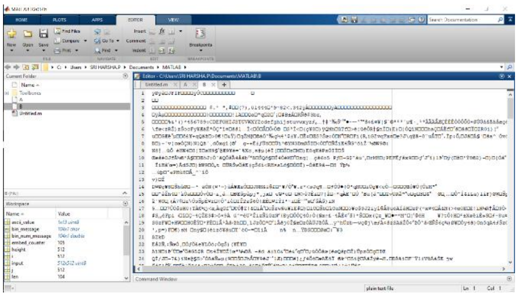


Fig 8: Data hidden using LSB method.

VI. FUTURE WORK

The future work of the project is supposed to be the implementation of the image encryption algorithm on different kits for the purpose of faster computation of data. FPGA kits are one among such kits or processing platforms which are capable of parallel computation of functions which leads to the faster computational speed and high speed result aquisition. Also, the usage of encrypted image using SIT to be used for the data hiding technique such as LSB is also a future work.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] R. Want and S. Dustdar, “Activating the internet of things [guest editors’ introduction],” Computer, vol. 48, no. 9, pp. 16–20, 2015.

[3] J. Romero-Mariona, R. Hallman, M. Kline, J. San Miguel, M. Major, and L. Kerr, “Security in the industrial internet of things,”

2016.[4] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the internet of things: a review,” in Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, vol. 3. IEEE, 2012, pp. 648–651.

[5] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, and D. Wagner, “Smart locks: Lessons for securing commodity internet of things devices,” in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM, 2016, pp. 461–472.

[6] D. Airehrour, J. Gutierrez, and S. K. Ray, “Secure routing for internet of things: A survey,” Journal of Network and Computer Applications, vol. 66, pp. 198–213, 2016.

[7] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks, vol. 10, no. 7, pp. 1497–1516, 2012.

[8] L. Da Xu, “Enterprise systems: state-of-the-art and future trends,” IEEE Transactions on Industrial Informatics, vol. 7, no. 4, pp. 630–640, 2011.

[9] P. Zhao, T. Peffer, R. Narayanamurthy, G. Fierro, P. Raftery, S. Kaam, and J. Kim, “Getting into the zone: how the internet of things can improve energy effificiency and demand response in a commercial building,” 2016.

[10] S. Misra, M. Maheswaran, and S. Hashmi, “Security challenges