2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016)
Dehradun, India 14-16 October 2016

A Novel Image Encryption Algorithm using AES and Visual Cryptography

Venkata Krishna Pavan Kalubandi, Hemanth Vaddi, Vishnu Ramineni, Agilandeeswari Loganathan

School of Information Technology, VIT Vellore.

Abstract— **With the current emergence of the Internet, there is a need to securely transfer images between systems. In this context, we propose a secure image encryption algorithm that uses both AES and Visual Cryptographic techniques to protect the image. The image is encrypted using AES and an encoding schema has been proposed to convert the key into shares based on Visual Secret Sharing. The cryptanalysis of the algorithm is then performed and is proved to be secure. The proposed algorithm is then implemented using python and the results are discussed along with the possible future modifications.**

Keywords— AES, Visual Cryptography, Image Encryption

I. INTRODUCTION

The primary objective of image encryption is to transmit an image securely over a connected network so that no unauthorized user should be able to decrypt the image. Image encryption has applications in many fields including Banking, Telecommunication and Medical Image Processing etc. Encryption has become an important part due to the emergence of Internet, where sending and receiving data across computers need security of some standard. Various algorithms have been proposed in this field to encrypt and decrypt images [1].

Research shows that using AES and Visual Cryptography to encrypt images is not entirely new. In [2], an encryption scheme has been proposed that splits the Image into R, G, and B components and encrypt them using AES. In [3], an encryption scheme has been proposed that uses AES and Visual Cryptography to store bio metric data in the cloud. However, these implementations do not disclose any information related to security in the private key generated using AES. Another interesting approach to encrypt image is using the chaotic theory based algorithms. Chaos theory, one of the studies in Mathematics is mainly involved in analyzing and predicting the behavior of dynamic systems that are highly changeable with respect to initial conditions [15] [16]. Many image encryption algorithms have been proposed using chaotic maps [18] [19] [20]. However, these are relatively newer algorithms and in certain systems where hardware encryption circuits for default algorithms like AES are built in, these kind of algorithms need an entirely new update of hardware. In this paper, we propose an algorithm that secures the key used in AES by converting key into an image and splitting it into n shares using Visual Secret Sharing techniques that is hardware friendly and offers backward compatibility. The proposed algorithm is tested against various attacks proposed till date and is found to be secure.

The rest of the paper is organized as described. Section II gives basic information about AES and Visual Cryptography. Section III gives details about the proposed algorithm along. Section IV describes about the experimental methodology used to evaluate the proposed algorithm. Section V discusses about the results obtained and about the cryptanalysis of algorithm. Section VI concludes the work with future scope.

II. BACKGROUND

A. Advanced Encryption Standard:

AES, Advanced Encryption Standard is a symmetric-key algorithm based on the Rijndael cipher developed by Joan Daemen and Vincent Rijmen. The AES algorithm has a block size of 128 bits. It supports three different key lengths of 128, 192 and 256 bits. AES [3] replaced Data Encryption Standard and it is now used worldwide. In AES there is no Feistel Network as opposed to the previous standard DES. The cipher consists of rounds, where the number of rounds depends on the key length: 10 rounds for a 128 bit key, 12 rounds for 192 bit key, and 14 rounds for a 256 bit key. The whole algorithm operates on a 4 x 4 matrix of bytes. The first rounds consist of four distinct transformation functions: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key. The final round contains three transformations. The Mix Columns function is not used in the final round. Each transformation takes one or more 4x4 matrices as input and produces a 4x4 matrix as output. Provided that all the four rounds are reversible, it is easy to prove that decryption does recover the plaintext. [3]

B. Visual Cryptography:

Visual cryptography, introduced by Naor and Shamir in 1995 [5], is a cryptographic scheme based on human visual system. In this scheme, the encrypted data is decrypted by the human eye. Due to this, Visual Cryptography doesn't require complex mathematical algorithms to encrypt and decrypt data. The main idea of this approach is to encrypt any message (text, image etc..) into a number of different images called shares. Only, when the shares are aligned together in order match the transparency among the subpixels, the secret message can be seen and recovered. One of the simplest

implementation of this approach is the 2 out of 2 sharing scheme, where the message to be encrypted is divided into 2 shares using exclusive xor operation where both shares needed for a successful decryption [5]. This scheme can be further extended to implement the k out of n scheme where the message to be encrypted can be split into n shares but only k shares are sufficient enough for decryption where k≤ n. However, even if k-1 shares are present, no information about the message can be revealed. Naor and Shamir developed this approach only on black and white images. After few years, Verheul and Tilborg [7] developed a systematic approach that can be applied on colored images. The problems involved with these newly proposed approaches is that they use random meaningless shares to hide the secret message and the quality of the recovered plaintext is not up to the mark. More advanced approaches based on visual cryptographic techniques were introduced in [4] [6] [8] where a colored image is divided into numerous meaningful cover images.

When cryptanalysis of the above schemes is considered, the divisions or shares exist in unencrypted form during the transmission and retrieval over a connected network. However, a man in the middle attack cannot predict the secret message with single share, but if the attackers are able to grab all the available shares, there is a possibility of recovering the plain text by using certain normal brute force attacks. Thus to get rid of this problem, we need to tighten the security of shares. For the same purpose we have used Private Key Cryptography in addition to Visual Cryptography so that even if attackers are able to get all the shares but they cannot retrieve the original secret without the access of private key [9].

III. PROPOSED ALGORITHM

As seen from the above information, both AES and Visual Cryptography are vulnerable to certain attacks and therefore are not suitable for Image Encryption. The algorithm proposed in this paper can not only withstand the existing attacks but is also secure from future attacks. In order to achieve such security standard, the best parts of both the algorithms are combined to work together. The proposed algorithm is divided into two phases.

A. Encryption Phase
B. Decryption Phase

A. Encryption Phase

In the encryption phase, two inputs are required. First one is the image I to be encrypted, second one is the Key K, the secret key in the form of string. The key K is then transformed to SK using SHA 256 hashing algorithm. The image I is encoded into Base64 string BI. The message BI, key SK are fed into the AES 256 encryption algorithm to generate cipher text CI. The original key K is then converted into an image KI using ASCII encoding and then it is split into shares KI1, KI2...KIN. The processes is visualized in Fig. 1 below

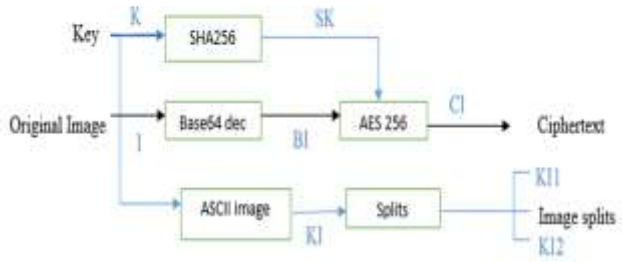

Fig 1. Encryption

Fig. 1. Encryption phase of the algorithm visualized.

Input: Encrypt - (I), (K), (N)
Output: (CI), 1, 2… N (KI)
Here
I – Image to be encrypted. Supports various standard formats.
K – Key File given as a sequence of characters.
N – Number of shares of Key to be generated (min 2)
CI – The base64 encoding of the encrypted image I.
KI – N images or shares of the Key that are generated.

Algorithm (with N = 2):

1. Read the input image and encode it using base64 standard.
2. Read the key file and initiate the AES 256 bit key using the hash (SHA-256) of key file.
3. Encrypt the image using the base 64 encoded text and hash generated in steps 1 and 2 respectively.
4. Create a new Image C of size (w, h) with pixel data p where
a. w - character support for key file (Default: 255)
b. h – Number of characters in the key file
c. p – Pixel Data to be filled (Default: 0)
5. For-each row i in height of image repeat:
a. Let j be ASCII code of the ith character in the key file.
b. Fill the first j pixels of the image in the ith row with black color.
i. C [i] [j] = 0 for every i, j in h, w such that j < ASCII(key[i])
6. Create N (= 2) Images (R, P) of the same size (w,h) and pixel data such that
a. For the first image R, pixel data is generated randomly. It can be either 0 (black) or 1 (white).
i. R [i] [j] = random(0, 1)
b. Second image pixel data P [i] [j] is defined such that
i. P [i] [j] = R [i] [j] xor C [i] [j] for every i, j in (h, w)

7. Output the encrypted encoding CI, Images P and R respectively.

B. Decryption Phase

In the decryption phase, two inputs are required. First one is the cipher text CI to be decrypted, second one is the array of shares KI, the secret shares of the key. From the shares, the original key image KI is constructed. The image KI is then

decoded to the key K by using ASCII decoding. The key K is then transformed to SK using SHA 256 hashing algorithm. The cipher CI, key SK are fed into the AES 256 decryption algorithm to generate message base64 encoding of Image BI. The encoding BI is then converted to output Image I by decoding. The processes is visualized in Fig. 2 below

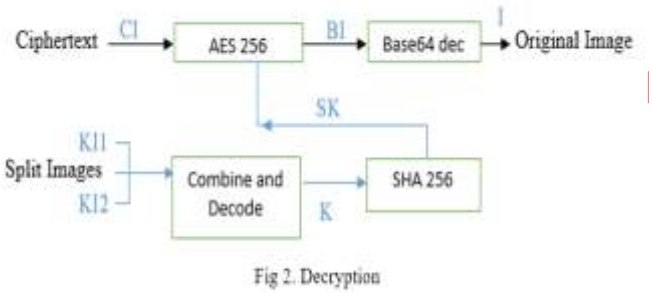

Fig 2. Decryption

Fig. 2. Decryption phase of the algorithm visualized.

Input: *(CI), 1, 2… N (KI)*
Output: *I*
Here
CI – Base 64 encoded cipher text.
KI – N shares of the Key that must be supplied for decryption.
I – Decrypted Image

Algorithm (with N = 2):

1. *Read the input cipher text of the image CI.*
2. *Load the Images K1, K2 from the input KI.*
3. *Create a new Image CK of size (w, h) same as K1, K2 such that*
   a. CK [i] [j] = K1 [i] [j] *xor* K2 [i] [j] *for every i, j in (h, w)*
4. *Initialize key K as an array of characters of size same as height of image CK (h).*
5. *For-each row i in height of image CK repeat:*
   a. *Let count = 0*
   b. *For each pixel j of the image in the i^{th} row with black color.*
      i. *increment count by 1*
   c. *Find the character ki by using ASCII code of the count generated after b. i.e., ki = char(count)*
   d. *Set K[i] = ki*
6. *With the Key K initialize the AES 256 Algorithm with hash(K) (SHA-256)*
7. *Decrypt the cipher text CI and save the decrypted base64 encoding as an Image I*
8. *Output the decrypted image I.*

Analysis of the Algorithm:

The time complexity of the algorithm can be assessed as follows. Three computations are required for the encryption as well as decryption phase. The first phase is applying AES algorithm to the encoded image, the second phase is converting the key to ASCII encoded image and the third phase is converting the key to SHA 256 hashing algorithm. Therefore, the total complexity of the algorithm is given by

*AES (Encoded Image) + SHA (Key) + (Key Length * 26)*

For a considerable image size and small key size, we can ignore the smaller terms and overall the complexity in the worst case is given by

O (AES(Image))

Which gives us a very fast running algorithm to securely encrypt and decrypt images.

IV. EXPERIMENTAL RESULTS

The experiments are performed on a Dell Inspiron 3543 with Intel Core i7-5500U CPU running at 2.4GHz, and supported by 8GB of RAM. The algorithm proposed in this paper is implemented using Python 2.7 scripting language. PIL (Python Imaging Library) module is used to create, load and save images in Python. Crypto module is used for the AES algorithm. The implementation also provides an optional user friendly interface to perform encryption and decryption operations.
The test cases required for encryption and decryption are generated as follows.

- Category 1: Images of medium quality (48 KB) with medium resolution (389 x 352 resol) with varying key size and varying image formats.
- Category 2: Images of medium quality (98.2 KB) with high resolution (1366 x 768 resol) with varying key size and varying image formats.
- Category 3: Images of high quality (4.32MB) with high resolution (4588 x 2365 resol) with varying key size and varying image formats

By selecting images of various sizes, qualities and formats, an honest evaluation of the algorithm in terms of its efficiency, performance is made possible.

The following images has been encrypted with key file sizes 20, 50, 100 of categories 1, 2, 3 respectively. The results are described below:

A. Category 1 Image

Fig. 3. Resolution: 389 * 352**, Size:** 48 kb**, Key Length:** 20

Image Cipher Length: 81600

B. Category 2 Image

Fig. 4 Resolution: 1366 x 768**, Size:** 615 kb**, Key Length:** 50

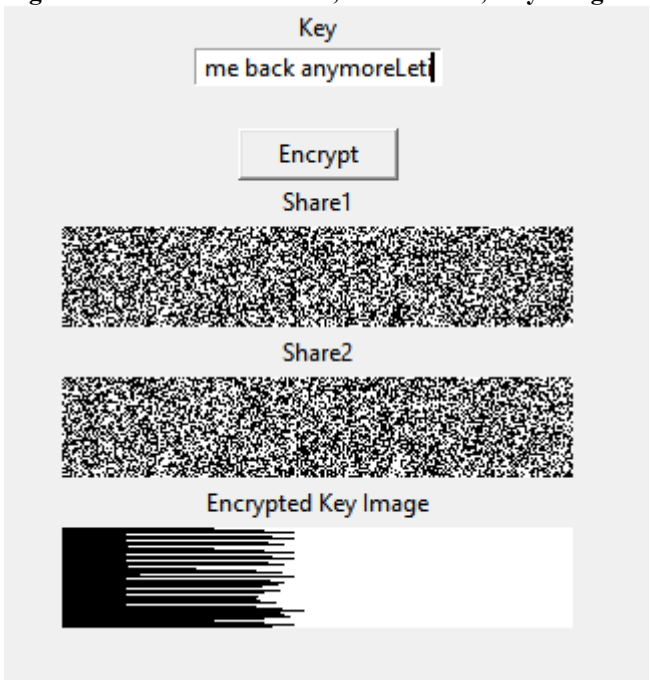

Fig 5: Image Cipher Length: 1119640

C. Category 3 Image

Fig 6: Resolution: 4588 x 2365**, Size:** 4.32MB**, Key Length:** 100

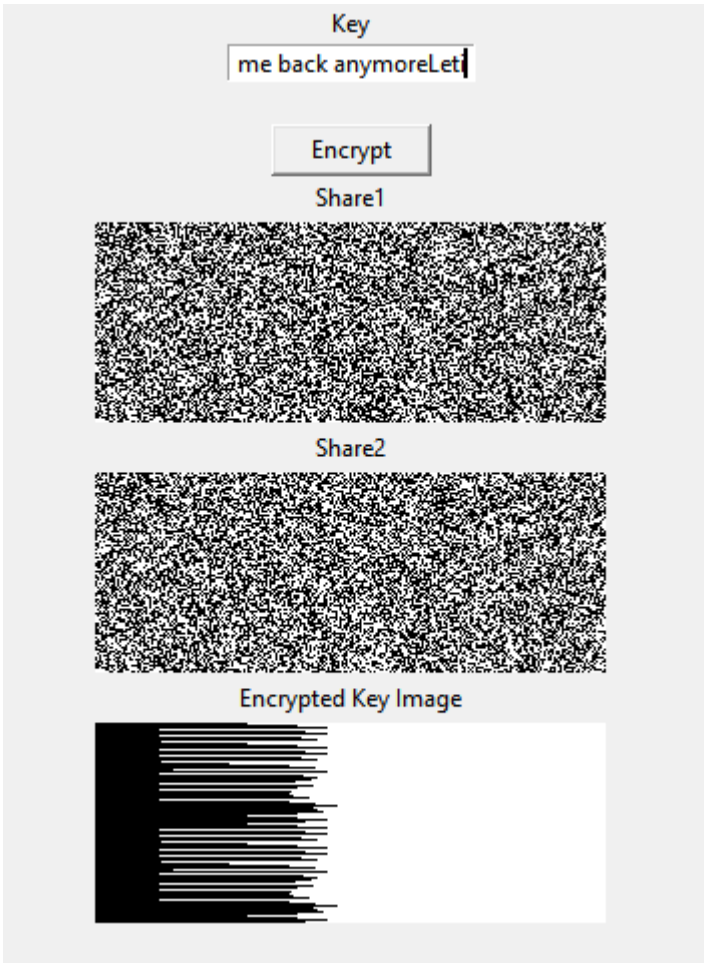

Fig 7: Image Cipher Length: 8062700

As seen from the above data, it is clear that the proposed algorithm can handle images of various categories, file sizes, qualities and resolutions. The size of the shares is based on the key size specified in the encryption phase. Even though large key file size is not recommended, the algorithm can handle large key sizes as well. This is due to the fact that each and every key undergoes a hashing mechanism implemented using SHA-256. The hash size is always 256 regardless of the key size.

With regard to the cipher text length, it has been observed that length of the cipher generated is approximately 1.5 times the size of the image. This may pose a problem for higher quality images because of the cipher text size. In such cases, the length of the cipher can be reduced using hashing mechanisms or the data can be in the form of encrypted chunks where each chunk is a part of the image, compressed using compression algorithms. Another possible solution to compress the image before encryption, so that file size can be reduced comparatively. Hence, the algorithm is suitable for sending compressed images in the network and it should be modified if one has to send raw and high quality images.

V. CRYPTANALYSIS OF ALGORITHM

Cryptanalysis is the study of ciphers, cipher texts and cryptosystems with a goal of finding weaknesses in them that will eventually allow the recovery of the plaintext from the

cipher text, without necessarily revealing much details about the key or the algorithm used for encryption. In this section, some of the common attacks against the existing system are explored and then how the proposed algorithm withstands to such attacks is discussed in detail.

A. Related Key attack:

The related-key attack is one of the cryptanalytic attacks in which the attacker or hacker tries to find the relationship between various keys used to encrypt the data without actually guessing the actual keys. However, the ultimate goal of the attacker is to find the exact relation and recover the actual keys. The relation between the keys can be any arbitrary bijective function F or even a highly sophisticated combination of such functions pre chosen by the attacker. If the simplest case of the attack is considered, then the above relation can be defined just as an XOR operation with a constant: $K2 = K \oplus C$, where the constant C is set by the attacker. This type of a simple relation itself gives the attacker an opportunity to trace back the XOR differences induced by the key difference C through the key scheduling mechanisms of the cipher. However, there are more sophisticated and complex forms of this type of attack that allow many non-linear relations between the keys. [10] [11] [12].

B. Meet-in-the-middle attacks with bi cliques:

Meet-in-the-middle attacks with bi cliques involves using complete bipartite graph structure to further extend the possible number of attack rounds on regular meet in the middle attacks. However, MITM attacks on block ciphers did not receive much attention when compared to differential, linear, impossible differential, and integral approaches. However, these types of attacks are practical in nature when compared in terms of complexity in implementation. A simple meet-in-the-middle attack requires only a single plaintext/ciphertext pair. The limited use of these attacks must be attributed to the requirement for large parts of the cipher to be independent of particular key bits. For the ciphers with non-linear key scheduling algorithms, like AES and most AES candidates, this requirement is very strong. As a result, the number of rounds broken with this technique is rather small which seems to prevent it from producing results on yet unbroken 8-, 9-, and 10-round (full) AES. Collision attacks use some elements of the meet-in-the-middle framework. [13] [14]

C. Other Image Attacks

Other types of attacks on images can be categorized into four types. The first type is the cipher image only attack. In this type the attacker only knows the cipher in the process and has to get back the image. In this case, the attacker has to break the AES algorithm. In the second type, the attacker has access to certain plain images and their corresponding cipher images. Even in this case, the attacker needs to break the AES algorithm in order to crack the key [17]. The third type of attack unique to this algorithm is the availability of both the shares of the key as well as the cipher image. This can be assumed as the worst possible case for the algorithm. Even in this case, the attacker has to figure the permutations involved in the ASCII encoding and also has to guess the underlying implementation of hashing algorithm that is used in conjunction with AES. This additional layer of security is not offered by AES or Visual Cryptography alone.

Even though attacks *A* and *B* these are less likely to occur, the proposed algorithm is designed in such a way that no information can be retrieved even if AES gets broken. This is achieved with the help of double encryption provided by Visual Cryptography and SHA-256 hashing algorithm. The attacker cannot even predict the key as the shares are visually encrypted and a compromise of all the shares alone can provide a hint at the key. Even at this stage, the attacker has to understand the encoding of the image to key to perform decoding. In such cases, triple encryption protection can be used to protect the encoding schema as well. However, it takes more time to encrypt the image and it has very limited applications where highest level of security is needed.

VI. CONCLUSION AND FUTURE WORK

The cryptographic methodology proposed in this paper has been tested on different types of input images with change in size of the image and keys of AES encryption algorithm. The entire time secret image is retrieved with good visual quality. In fact, there is no observable change in the quality of image, since the image processing is mostly done on the key images and its shares. The confidentiality of shares is also tested by changing the key shares before reaching to the destination. In all the cases it has been observed that if any intruder will be successful in getting the encrypted shares from network, he or she cannot retrieve the original secret image without availability of cipher. Also since the complexity of the encryption is double layered the hacker could not possibly get to know the algorithms used in the encryption. So it is still difficult to crack the visual cryptography even if the hacker gets his hands on the key shares over the network.

As future work, this scheme can possibly be modified to use color image in place of binary image for the key and then generate the shares using Visual Cryptography. Improvement in the quality of key share images can be done i.e., a meaningful image instead of normal binary image. More sophisticated public key encryption can be used to reduce key size and cipher text size. Image processing attacks like translation, rotation and scaling of key shares can also be controlled.

REFERENCES

[1]. Pakshwar, Rinki, Vijay Kumar Trivedi and Vineet Richhariya. "A survey on different image encryption and decryption techniques.", IJCSIT) International Journal of Computer Science and Information Technologies 4.1, 2013.

[2]. Kumar, M. Arun, and K. Jhon Singh, "Novel Secure Technique using Visual Cryptography and Advance AES for images.", International Journal of Knowledge Management and e-learning, Vol. 3, No. 1, pp. 29-34, 2011.

[3]. Nikita, Ranjit Kaur, "A Survey on Secret Key Encryption Techniques", Impact: International Journal of Research in Engineering & Technology IMPACT: IJRET, May, 2014.

[4]. Chang, C. C. and Yu. T. X., "Sharing a Secret Gray Image in Multiple Images, in the Proceedings of International Symposium on Cyber Worlds: Theories and Practice", Tokyo, Japan, Nov. 2002.

[5]. M. Naor and A. Shamir, Visual cryptography. "Advances in Cryptology" EUROCRYPT '94, 1995

[6]. C. Chang, C. Tsai, and T. Chen, "A new scheme for sharing secret color images in computer network"., International Conference on Parallel and Distributed Systems, July 2000.

[7]. E. Verheul and H. V. Tilborg., "Constructions and properties of k out of n visual secret sharing schemes. Designs", Codes and Cryptography, 1997.

[8]. C. Yang and C. Laih., "New colored visual secret sharing schemes. Designs", Codes and Cryptography, 2000.

[9]. Kulvinder Kaur and Vineeta Khemchandani, "Securing Visual Cryptographic Shares using Public Key Encryption", Advance Computing Conference (IACC), Feb, 2013

[10]. Orr Dunkelman, Nathan Keller⋆, and Adi Shamir, "Improved Single-Key Attacks on 8-round AES-192 and AES-256"**,** ASIACRYPT, LNCS, 2010.

[11]. Alex Biryukov, Dmitry Khovratovich, Ivica Nikolić, "Distinguisher and Related-Key Attack on the Full AES-256", Advances in Cryptology - CRYPTO 2009

[12]. Joan DAEMEN, Vincent RIJMEN, "On The Related-Key Attacks Against AES", Proceedings of The Romanian Academy, Series A, 2012

[13]. Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir, "Key Recovery Attacks of Practical Complexity on AES Variants with up to 10 Rounds", Cryptology ePrint Archive, 2009

[14]. Andrey Bogdanov⋆, Dmitry Khovratovich, and Christian Rechberger⋆, Biclique Cryptanalysis of the AES-192 and AES-256, "International Conference on the Theory and Application of Cryptology and Information Security", 2009.

[15]. S. Parker., L. O. Chua., "Chaos: a tutorial for engineers. Proceedings of the IEEE", vol. 75, no. 8, pp. 982–1008, 1995

[16]. W.Wu .,N. F. Rulkov., "Studying chaos via 1-Dmaps—a tutorial. IEEE Trans. on Circuits and Systems I Fundamental Theory and Applications", vol. 40, no. 10, pp. 707–721, 1993

[17]. Chin-Chen Changa, Min-Shian Hwangb, Tung-Shou Chenc, "A new encryption algorithm for image cryptosystems", 2000

[18]. Y.-Q. Zhang, X.-Y. Wang "Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation", 2014.

[19]. Y.-Q. Zhang, X.-Y. Wang "A new image encryption algorithm based on non-adjacent coupled map lattices", 2015.

[20]. H. Liu, X. Wang "Color image encryption based on one-time keys and robust chaotic maps" 2010