Lossless Encryption for Color Images using a Binary Key-image

SIMHADRI KOLLU
Electronics & Communication Engineering.
Godavari Institute of Engineering & Technology.
Rajahmundry, India.
E-mail: simhadri_engg@yahoo.co.in.

P. SOUNDARYA M.Tech.
Electronics & Communication Engineering.
Godavari Institute of Engineering & Technology.
Rajahmundry, India.
E-mail: soundarya_palivela@yahoo.co.in

IJAEST

Abstract-**The Cryptographic security of data depends on the security provided for the key used to encryption. Lossless Encryption for Color images using a Binary Key-image. The condition, the key image size is same as the original image. The key image is either a bit plane or an edge map generated from another image. The lossless image encryption algorithms using this key image technique. The key is selected to the grayscale image for new/existing grayscale image and the key is selected to the color image for new/existing color image. The code is done in both grayscale and color images using lossless encryption algorithms. The execution of these algorithms is discussed against common attacks such as the plaintext attacks, brute force attack and cipher text attacks. The security analysis and experimental results show that the proposed algorithms can fully encrypt all types of images. This makes them suitable for securing video surveillance systems, multimedia applications and real-time applications such as mobile phone services.**

Keywords-lossless image encryption, key-image, plaintext attack, brute force attack, ciphertext attack, bit plane, edge map.

1. INTRODUCTION

Cryptography is, traditionally, the study of means of converting information from its normal, comprehensible from into an incomprehensible format, rendering it unreadable without secret knowledge-the art of encryption. The first visual cryptographic technique was pioneered by Moni Naor and Ad Shamir in 1994. It involved breaking up the image into n shares so that only someone with all n shares could decrypt the image by overlaying each of the shares over each other. Practically this can be done by printing each share on a separate transparency and then placing all of the transparencies on top of each other. In their technique n-1 shares revealed no information about original image.

Visual surveillance systems and networks make remote video monitoring available for homeland security purpose and also easy to transmit and share videos and image data. With the ubiquitous deployment of visual surveillance systems in many important areas such as airports, commercial centers and also military strategic places, large amounts of videos and images with security information are generated, transmitted and stored. Image security is a major challenge in storage and transmission applications. For example, medical images with a patient's records may be shared among the doctors in different branches of a health service organization over networks for different clinical purposes. These images and videos may contain private information. Providing high security for these images and videos becomes an important issue for individuals, business and governments as well.

Image encryption is an effective approach to protect images or videos by transforming them into completely different formats. Several data encryption algorithms like Data Encryption Standard (DES) [1] and Advanced Encryption standard (AES) [2] are being employed for protecting digital information, chaos based [3] and combinatorial permutations [4] are proposed for encrypting images. Applications in the automobile, medical, Construction and the Fashion industry require designs, scanned data, building plans and blue-prints to be safe guarded against espionage. Considering the long life time of images in the mentioned domains, it is imperative to develop and employ techniques which protect the content throughout their lifetime [5].

Image encryption can be accomplished by scrambling image pixel positions using different techniques in the spatial domain [6-8]. One example is the recursive sequence based image scrambling approach. It scrambles images using different recursive sequences such as the Fibonacci sequence [9], Cellular automata [10] and chaotic maps [11-12]. Image encryption can also be accomplished by scrambling coefficient matrices/blocks in the transform domain [13, 14]. These approaches have extremely low security levels due to the lack of security keys. Another approach for image encryption is to change image pixel values based on the combination of image bit plane decomposition and logic operations [15, 16]. This security method is also much lower because the results of its decomposition process and logic operations are predictable. It's not immune to plaintext attacks. To achieve higher levels of security, solution is to change image pixel values or blocks while scrambling the positions using different techniques.

We introduce two new lossless image encryption algorithms,

i. BitplaneCrypt algorithm.
ii. EdgemapCrypt algorithm.

Using a new concept using a binary "key-image", with the same size of the original image to be encrypted. The bit plane crypt algorithm generates the key-image by extracting a binary bit plane from another new or existing image. The other algorithm is an edge map obtained from a new or existing image using a specific edge detector with a specified threshold. The algorithms decompose the original image into its binary into its binary bit planes. The bit planes are encrypted by performing an XOR operation with the key image one by one. And then the order of all bit planes is inverted. And combine all bit planes. The resulting encrypted image can be obtained by applying a scrambling algorithm to the image.

2. IMAGE ENCRYPTION ALGORITHMS

The underlying foundation of both algorithms is to change image pixel values by performing the XOR operation between the key- image and each bit plane of the original image. This is followed by an image scrambling process whish changes the locations of image pixels or blocks.

A. *The BitplaneCrypt algorithm for 2D image*

The BitplaneCrypt algorithm uses a binary bit plane as the key-image. The algorithm is described in Fig. 1. The original image decomposes the binary bit planes. And the new or existing image decomposes bit planes and the key image by exacting the r^{th} bit plane of the selected image, where r is the location of bit plane. The key-image size is same as the original image. Perform the XOR operation between the key image and each bit plane of the original image, the XOR-ed bit planes are invert the order of all bit planes and combined. Finally scramble the resulting image using a selected scrambling method to generate the resulting encrypted image.

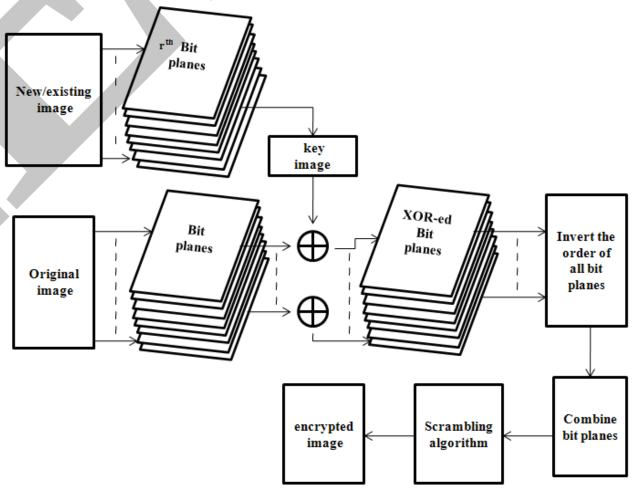

Figure 1. 2D image BitplaneCrypt algorithm

The users have flexibility to choose any new/existing image to generate the key image. This image can be a public image or an image created by the users themselves. The key-image can be selected from one of the bit planes of this image. Therefore, the security keys of the algorithm consist of the image or the location of the image used to generate the key-image. The authorized user provides the correct security key in the generated key-image. In the decryption process, the user unscrambles the encrypted image using the corresponding scrambling algorithm; it then decomposes the image into bit planes. Each bit planes applied an XOR

IJAEST

operation with the key-image. And reverted to the all bit planes in the original order. The original image can be reconstructed by combining all bit planes.

B. The EdgemapCrypt algorithm for 2D image

This algorithm described in Fig. 2, generate the key-image from another new/existing image with the same size as the original image using the specific edge detector with a selected threshold value. The key image is considered as the edge map. The edge map is frequently used in image enhancement, compression, segmentation and recognition. The applications of edge map can also be extended to image encryption.

The EdgemapCrypt algorithm, first the original image decomposes the binary bit planes. Each of them is encrypted by performing an XOR operation with the edge map. Next the algorithm inverts the order of all bit planes and combines them together. And apply the scrambling algorithm to generate the final resulting encrypted image.

The new/existing image is an either a public online data base image or a new image generate by the users. The edge map can be obtained by using any existing edge detector such as canny, prewitt, sobel, and any other edge detectors. Therefore, the security keys for this algorithm consist of the image or its location which is used to generate the edge map, the type of the edge detector, the edge detector threshold's value, and the security keys of the scrambling algorithm.

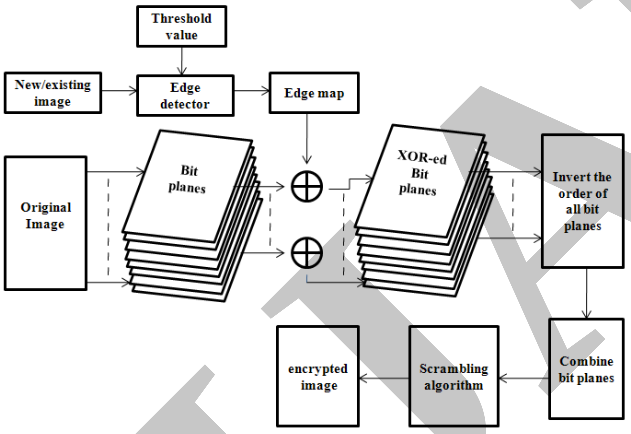

Figure 2. 2D image EdgemapCrypt algorithm

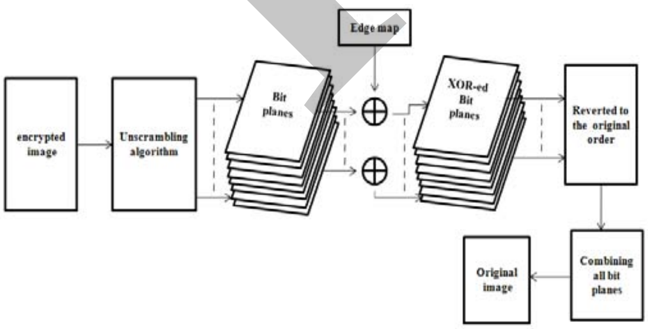

Figure 3. EdgemapCrypt Decryption algorithm

The decryption process, first apply the unscrambling algorithm to the encrypted image using the selected scrambling algorithm and it decomposes the unscrambled image into binary bit planes and perform the XOR operation between the edge map and each bit planes. The users should be provided the security keys which help them to obtain the correct edge map. The order of all bit planes is restored to the original order and combines all bit planes. The resulting image is the original image. The EdgemapCrypt decryption algorithm illustrated in Fig. 3.

C. BitplaneCrypt algorithm for 3D image

The BitplaneCrypt algorithm uses a binary bit plane as the key image. This bit plane is extracted from another new or existing image which is different from the original image to be encrypted.

The BitplaneCrypt algorithm is described in Fig. 4. It first generate the key-image by extracting the r^{th} bit plane of the selected image, where is the location of the bit plane. The 3D image or RGB (Red Green Blue) image, assuming i, consider

$$i = \begin{cases} 1 \text{ for R,} \\ 2 \text{ for G,} \\ 3 \text{ for B.} \end{cases}$$

The algorithm then decomposes the original image i into binary bit planes and performs an XOR operation between each of these bit planes and key-image and the order of bit planes are inverted. The resulting encrypted image can be obtained by applying a scrambling algorithm to the image from a combination of all bit planes.

The BitplaneCrypt Algorithm

Input The original 3D image (assuming i) to be Encrypted.
Step 1 Choose a new or existing image with the same size of the original image.
Step 2 Obtain the key image by extract r^{th} bit plane Of the image in Step 1.
Step 3 Consider i={1 (R),
2 (G),
3 (B).
Step 4 Decompose the original image i into binary Bit planes.
Step 5 Perform the XOR operation between the key-Image and each bit plane in Step 4.
Step 6 Invert the order of all bit planes.
Step 7 Combine all bit planes together to obtain the 3D image.
Step 8 Scramble the resulting image using a selected scrambling method to generate the resulting Encrypted image.
Output The encrypted 3D image.

Figure 4. BitplaneCrypt algorithm 3D image

IJAEST

The users have flexibility to choose any new or existing image to generate the key-image. This image can be a public image or an image created by the users themselves. The key-image can be selected from one of the bit plains of the image. The security keys of the algorithm consist of the image or the location of the image used to generate the key-image, the location of the bit plane as the key-image.

In the decryption process, the correct security keys should be provided to the authorized user to generate the key-image. The user unscrambling the encrypted image using the corresponding scrambling algorithm, the resulting image into bit planes. Applied an XOR operation between the key-image and each bit plane the order of bit planes is reverted to the original order. The original image can be reconstructed by combining all bit planes.

D. The EdgemapCrypt algorithm for 3D image

The edge map is frequently used in image enhancement, compression, segmentation and recognition. The application of edge maps can also be extended to image encryption. An edge map is considered as the key-image in this algorithm. The edge map is generated from new/existing image for color or grayscale image with same size as the original image using the specific edge detector with a selected threshold value.

The EdgemapCrypt algorithm for 3D image is described in Fig. 5.

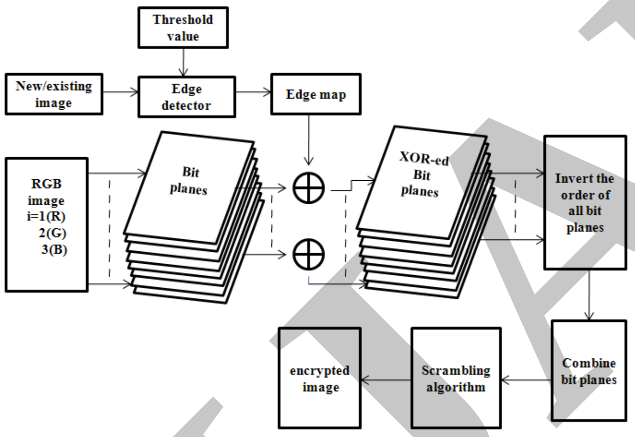

Figure 5. EdgemapCrypt algorithm 3D image

The edgemapCrypt algorithm, first decomposes the RGB image i=1(R), 2(G) and 3(B) into binary bit planes. Each bit plane is encrypted by performing an XOR operation with the key-image (an edge map created from new/existing image) and invert the all XOR-ed bit planes. The resulting encrypted image can be obtained by applying a scrambling algorithm to the image from a combination of all bit planes.

An edge map is a public image or a new image created by the users. The edge map can be obtained by using any existing edge detector such as sobel, canny, prewitt or any other edge detector. The canny edge detector is better compare to other detectors. The users have flexibility to choose any new image or any existing edge detector or any threshold value to generate the edge map used as a key-image. They also have flexibility to use any existing image scrambling method for the EdgemapCrypt algorithm. The security of this algorithm consist of the image or its location which is used to generate the edge map, type of edge detector, threshold value, and the security keys of the scrambling algorithm.

To reconstruct the RGB image, the users should be provided the correct security key-image (i.e. edge map). The decryption process first generates the edge map from the selected image using the security keys. It then unscrambles the encrypted image using the selected scrambling algorithm. Next, it decomposes the unscrambled image into its binary bit planes and performs XOR operation between the edge map and each bit plane. The order of all bit planes is restored to the original order. And combine all bit planes. The resulting image is the RGB image.

3. EXPERIMENTAL RESULTS

Different 2D and 3D images such as grayscale images, color images and animal images have been successfully implemented for Lossless encryption algorithms. The simulation results are provided to show the performance of the algorithms for 2D and 3D image encryption.

A. 2D image Encryption

There are several types of 2D images such as animal images, grayscale images and biometrics. Decomposes the number of bit planes and encrypted one by one for 2D.

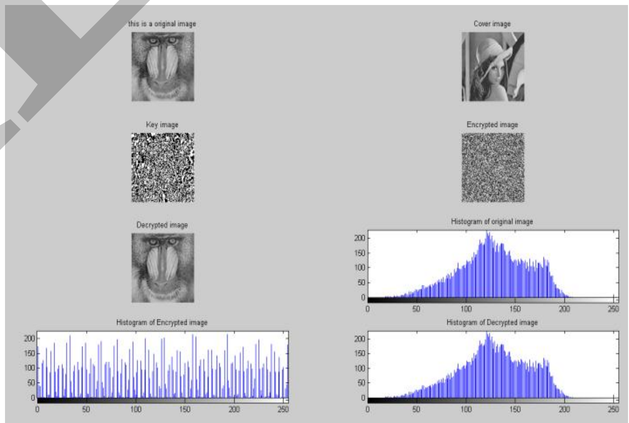

Figure 6. Grayscale image encryption using the BitplaneCrypt algorithm. (a) The original 512×512 grayscale baboon image; (b) A cover image 512×512 Lena image; (c) The 6th bit plane of the image in (b); (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

IJAEST

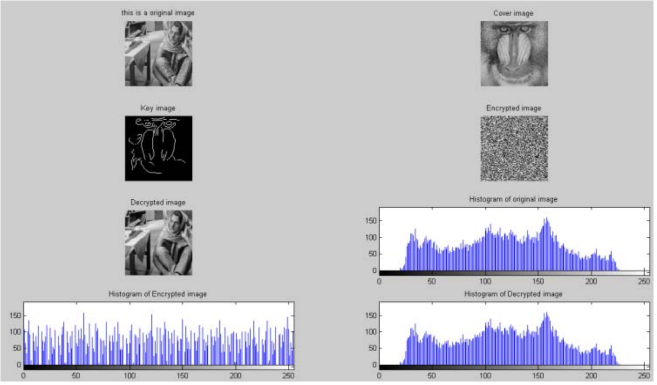

Figure 7. Grayscale image encryption using the EdgemapCrypt Algorithm.
(a) The original 256×256 grayscale image; (b) A cover image 256×256 baboon image; (c) The edge map of the baboon image in(b),canny,0.4; (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

Figure 6. Shows that an example of grayscale image encryption using the BitplaneCrypt algorithm. The key image is 6^{th} bit plane of the 512×512 grayscale cover image. Figure 7 shows results of grayscale image encryption using the EdgemapCrypt algorithm. The key image is obtained from a 256×256 grayscale baboon image using the canny edge detector with a threshold 0.4.

Both the results, the original images are fully encrypted as show in Fig. 6(d) and Fig. 7(d). The encrypted images are almost equal in grayscale value range as show in Fig. 6(g) and Fig. 7(g). The reconstructed image is same as the original image. This is one advantage of the presented algorithms. The reconstructed images are verified in Fig. 6(e) and Fig 7(e). The histograms are same in the original images in Fig. 6(f) and 7(f), and decryption images in Fig. 6(h) and 7(h).

The Lena image encryption examples using the BitplaneCrypt and EdgemapCrypt algorithms are shown in Fig. 8 and Fig. 9, respectively. The BitplaneCrypt algorithm, the key image is 4^{th} bit plane of a 128×128 grayscale Barbara image. The key image of the EdgemapCrypt algorithm in Fig. 9 is generated from a 512×512 grayscale baboon image using a prewitt edge detector with a threshold 0.3. The original images are also fully encrypted and completely reconstructed. The encryption images in Fig. 8(d) and Fig. 9(d), histograms in Fig. 8(g) and Fig. 9(g). The decryption images can be verified in Fig. 8(e) and Fig. 9(e).The decryption images histograms in Fig. 8(h) and Fig. 9(h), respectively. All this results prove that the presented algorithms are Lossless Encryption methods.

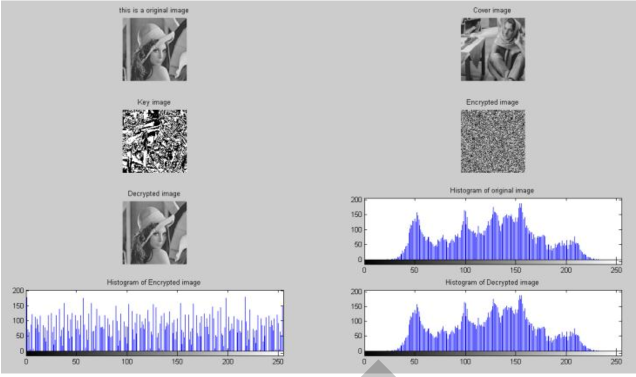

Figure 8. Grayscale Lena image encryption using the BitplaneCrypt Algorithm.
(a) The original 128×128 grayscale Lena image; (b) A cover image 128×128 barbara image; (c) The 4^{th} bit plane of the image in (b); (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

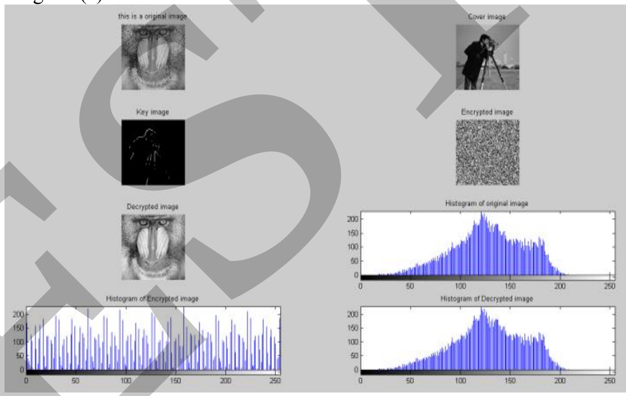

Figure 9. Grayscale baboon image encryption using the EdgemapCrypt Algorithm.
(a) The original 512×512 grayscale baboon image; (b) A cover image 512×512 cameraman image; (c) The edge map of the cover image in(b),prewitt, 0.3; (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

B. 3D image Encryption

The color image encryption, such as color images and 3D animal and medical images, the color images are contain the several 2D components, each component can be considered as a 2D image. The presented algorithms can be accomplished by encrypting all the 2D components only one by one for the 3D image encryption. In this encryption the original and existing images are color images, the convert existing color image to grayscale image and the key image generated to the grayscale image.

IJAEST

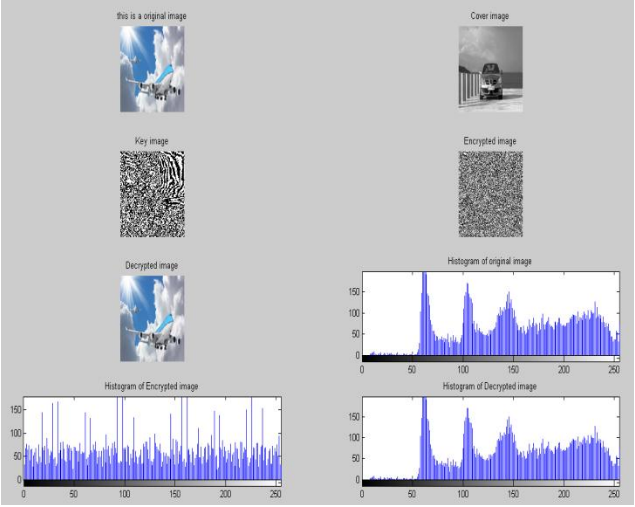

Figure 10. Color image encryption using the BitplaneCrypt algorithm.
(a) The original 512×512 color image; (b) A cover image 512×512 car image; (c) The 7th bit plane of the image in (b); (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

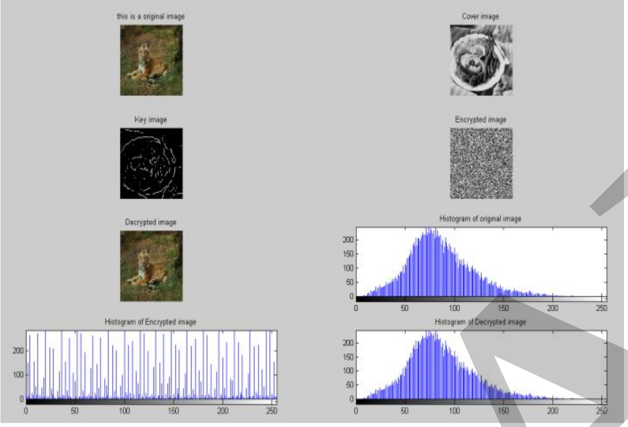

Figure 11. Color image encryption using the EdgemapCrypt Algorithm.
(a) The original 256×256 color image; (b) A cover image 256×256 image; (c) The edge map of the cover image in(b), sobel, 0.2; (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

Figure 10, Shows that an example of color image encryption using the BitplaneCrypt algorithm. The key image is 7th bit plane of the 512×512 color cover image. Figure 11, Shows results of color image encryption using the EdgemapCrypt algorithm. The key image is obtained from a 256×256 color image, the convert color image to grayscale image and using the sobel edge detector with a threshold 0.2.

Both the results, the original images are fully encrypted as show in Fig. 10(d) and Fig. 11(d). The encrypted images are equal in histograms as show in Fig. 10(g) and Fig. 11(g). The reconstructed image is same as the original color image. The reconstructed images are verified in Fig. 10(e) and Fig 11(e). The histograms are same in the original images in Fig. 10(f) and 11(f), and decryption images in Fig. 10(h) and 11(h).

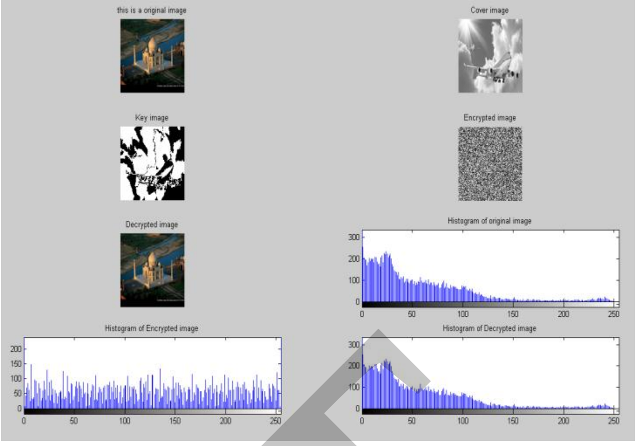

Figure 12. Taj mahal Color image encryption using the BitplaneCrypt Algorithm.
(a) The original 512×512 color image; (b) A cover image 512×512 aero plane image; (c) The 2th bit plane of the image in (b); (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

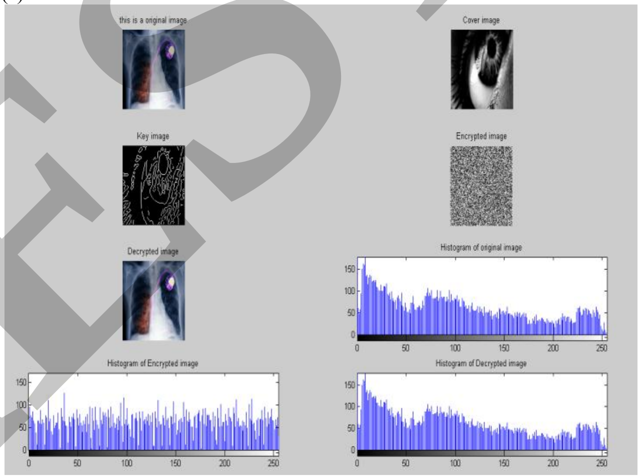

Figure 13. Chest X-ray Color image encryption using the EdgemapCrypt Algorithm.
(a) The original 512×512 color image; (b) A cover image 512×512 image; (c) The edge map of the cover image in(b), canny, 0.2; (d) Encrypted image; (e) Decrypted image; (f) Histogram of original image in (a); (g) Histogram of encrypted image in (d); (h) Histogram of decrypted image in (e).

Figure 12 and Figure 13, other examples of the BitplaneCrypt algorithm and the EdgemapCrypt algorithm. The key image is 2th bit plane of the 512×512 color cover image in Fig. 12. The edge map image in Fig.13, using the canny edge detector with threshold 0.1.

Both the results, the original images are fully encrypted as show in Fig. 12(d) and Fig. 13(d). The encrypted images are equal in histograms as show in Fig. 12(g) and Fig. 13(g). The reconstructed image is same as the original color image. The reconstructed images are verified in Fig. 12(e) and Fig 13(e). The histograms are same in the original images in Fig. 12(f) and 13(f), and decryption images in Fig. 12(h) and 13(h).

IJAEST

4. SECURITY ANALYSIS

The security is informant for both Lossless encryption algorithms and encrypted objects. We discuss the cryptography point of view some security issues of the BitplaneCrypt and EdgemapCrypt algorithms.

i. Security Key Space

The selection of the security key is important for BitplaneCrypt and EdgemapCrypt algorithms. The security keys of BitplaneCrypt algorithm are the location of the image or the combination of the image used to generate the key-image. The location of the bit plane used as the key image, the security keys of the scrambling algorithm. Another is an EdgemapCrypt algorithm, the security key is edge map depends on the location of the image the type of the edge detector, the edge detector's threshold, and the security keys of the scrambling algorithm. The original image can be completely reconstructed without any distortion only when the correct security keys are being utilized.

The key image generate both algorithms are new or existing image with the same size of the original image. The BitplaneCrypt algorithm, it has a huge numbers of possible choices assuming p_1. Each of its bit planes can be used as a key-image. Its gray levels within 0-255, therefore the number of possible choices of the key-image for this algorithm is $8P_1$. In addition, the scrambling algorithm can be used to the scramble the bit plane in both algorithms for new or existing image. The presented algorithms, the selection of the security keys in the selected image scrambling algorithm are also part of combination of the security keys, assuming their possible choices are P_S which not more than M!N! If the grayscale original image is an M×N. thus security keys space of the BitplaneCrypt algorithm is $8P_1P_S$.

The EdgemapCrypt algorithm, any new/existing edge detector can be used. Assuming its possible choices is P_E. The threshold value is rational number less than 1. However, not all the threshold values can achieve a desirable encryption result. Assuming P_{TH} is the possible number of choices may not be ∞, the security keys space for the EdgemapCrypt algorithm is $P_1\ P_E\ P_{TH}\ P_S$.

ii. Brute Force Attack

The Brute Force Attack is an attack model. The attacker tries to guess the number of possible security keys by conducting an exhaustive search of all combinations of security keys of the encryption algorithm. Theoretically, the key space is feasible for the encryption algorithm. It is limited and the attacker knows encryption algorithm.

The Lossless encryption algorithm are not infinite for security key spaces, they are still sufficiently large since to generate the key image for using the large number of possible new/existing images. As a result, the two algorithms can withstand the brute force attack.

iii. Ciphertext-only Attack

In Cryptography, the original information is called the plaintext and encrypted plaintext is called the ciphertext.

The ciphertext-only attack is an attack model, in this algorithm only studying the ciphertext and tries to deduce the security keys. This attack can be used to studying the encrypted images for the recover the original images data. The attacker recover the more portions of the original images, fewer portions of the images are encrypted for without knowing the encryption algorithm and its security keys the security level is an extremely low in this encryption scheme if it cannot with stand this attack.

From the experimental results, the original images are fully encrypted. It is visually unrecognizable and totally different from the original images. They contain almost no visual information of the original images. The histograms are equal in their distributions of the encrypted images. As a result the BitplaneCrypt and EdgemapCrypt algorithms can withstand the ciphertext-only attack.

iv. Chosen-Ciphertext Attack

It is an attack model in which the attacker can chose some ciphertexts and their corresponding plaintexts. Therefore, the attackers using the unknown ciphertext from recover the original plaintext or deduce the security keys in encryption. The attack could also be accomplished without knowing the encryption algorithm and its security keys if the image data does not change during the encryption process.

From the above analysis, the encryption process changes both the image data and pixel location. Therefore, the presented algorithms can also withstand the chosen-ciphertext attack.

v. Known Plaintext Attack

Known plain attack is an attack model in which an attacker studying a number of plaintexts and the corresponding ciphertext for tries to obtain the security keys of encryption algorithm. The condition of this attack is that the attacker should have same plaintexts and corresponding ciphertext. The attackers, without knowing the encryption algorithm and its security keys nevertheless partially or completely break the encrypted image if the encryption process doesn't change the images data.

The Lossless encryption algorithms are design to change the image data for the XOR operation and inverting the order bit planes. The images pixel positions are changed using the images scrambling algorithm. This make the encrypted image data are not use full for the attacker using this type of attack. Thus, both algorithms can withstand the known-plaintext attack.

vi. Chosen-Plaintext Attack

The chosen-plaintext attack is an attack model in which the attacker can choose a number of plaintexts and then deduce there corresponding ciphertexts. As result, reconstruct the original plaintext from unknown ciphertexts or the attacker can choose any useful information as plaintext in order to deduce the security keys of encryption algorithms. The attack can break the encrypted image

IJAEST

without knowing the encryption algorithm and its security keys, if the images data doesn't change during the encryption process.

Both the BitplaneCrypt and EdgemapCrypt algorithms change the images data and pixel locations they can withstand the chosen-plaintext attack.

5. CONCLUSION

We have introduced Lossless Encryption for Color Images using a Binary Key-image. We also introduced two image encryption algorithms based on this key-image. To generate key image an either a bit plane in the BitplaneCrypt or an edge map in the EdgemapCrypt algorithm.

The grayscale images and color images are fully encrypted from both algorithms the completely reconstruct the 2D images and 3D images without any distortion from the original image. The BitplaneCrypt and EdgemapCrypt algorithms have extremely large security key space for the cryptography point of view and can withstand most common attacks search as the plaintext attacks and brute force attacks and ciphertext attacks.

The key image size is generated from existing image, which as the same size as original image. An Edge detector such as canny, sobel, prewitt or any other detector with any specified threshold can be used to create the edge map as key image for the EdgemapCrypt method can be applied to these two presented algorithms for any new image. All these ensure the images can be protected with a higher security level.

Both algorithms are easy to implement in hardware because they operate at the binary levels. They are also suitable for multimedia applications and real time application such as mobile phone services and wireless networks.

REFERENCES

[1] National Institute of Standards and Technology, "Data Encryption Standard (DES)," http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf, 1999.

[2] National Institute of Standards and Technology, "Advanced Encryption Standards (AES)," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[3] J. Cheng; J.I. GUI, "A new chaotic key-based design for image Encryption and Decryption," The 2000 IEEE International Symposium on Circuits and Systems, 2000. Proceedings. ISCAS Geneva, vol.4, pp. 49-52, May. 2000.

[4] S. Li, C. Li, G. Chen, N. G. BourBakis, and K.-T. Lo, "A General quantitative cryptanalysis of Permutation-only Multimedia ciphers against plaintext attacks," Signal Processing: Image Communication, vol. 23, no. 3, pp.212-223, 2008.

[5] K. C. lyer and A. Subramanya, "Image Encryption by Pixel Property Separation," http://eprint.iacr.org/2009/043.pdf, Cryptology ePrint Archive, 2009.

[6] M. Ashtiyani, P. M. Birgani, and H. M. Hosseini, "Chaos-Based Medical Image Encryption Using Symmetric Cryptography," in Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on, 2008, pp. 1-5.

[7] M. Yang, N. Bourbakis, and S. Li, "Data-image-video Encryption," potential, IEEE, vol. 23, no. 3, pp. 28-34, 2004

[8] Y. Zhou, S. Again, V. M. Joyner, and K. Panetta, "Two Fibonacci p-code based image scrambling algorithms," in Image Processing: Algorithms and Systems VI, San Jose, CA, USA, 2008, pp. 681215-12.

[9] J. Zou, R. K. Ward, and D. Qi, "A new digital image Scrambling method based on Fibonacci numbers," in Circuits And Systems, 2004. Iscas'04. Proceeding of the 2004 International Symposium on, 2004, pp.III-965-8 vol. 3.

[10] R.-J. Chen and J.-L. Lai, "Image security system using Recursive cellular automata substitution," pattern Recognition, vol. 40, no. 5, pp. 1621-1631, 2007.

[11] J. C. Yen and J. I. Geo, "Efficient hierarchical Chaotic Image Encryption algorithm and its VLSI realization," Vision, Image and Signal Processing, IEEE Proceeding-, vol.147, no. 2, pp. 167-175, 2000.

[12] Z. H. Guan, F. J. Huang, and W. J. Guan, "Chaos-based Image Encryption algorithm," Physical Letters A, vol. 346, No. 1-3, pp. 153-157, Oct 2005

[13] G.-S. GU and G.-Q. Han, "The Applications of Chaos and DWT in Image Scrambling," in Machine Learning and Cybernetics, 2006 International Conference on, 2006, pp. 3729-3733.

[14] T. Li, S. Zhou, Z. Zeng, and Q. Ou, "A new Scrambling Method based on semi-frequency domain and chaotic System," in Neural Networks and Brain, 2005. ICNN&B '05. International Conference on, 2005, pp.607-610.

[15] J.-W. Han, C.-S. Park, D.-H. Ryu and E.-S. Kim, "Optical Image Encryption based on XOR operations," Optical Engineering, Vol. 38, no. 1, pp. 47-54, 1999.

[16] R. Lukac and K. N. plataniotis, "Bit level based secret Sharing for image encryption," Pattern Recognition, vol. 38, no. 5, pp. 767-772, 2005.

IJAEST