Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Image Encryption Using Binary Key-images

Yicong Zhou, Karen Panetta, *Fellow, IEEE*
Department of Electrical and Computer Engineering
Tufts University
Medford, MA 02155, USA
yzhou0a@ece.tufts.edu, karen@ece.tufts.edu

Sos Agaian, *Senior Member, IEEE*
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, TX 78249, USA
Sos.Agaian@utsa.edu

***Abstract*—This paper introduces a new concept for image encryption using a binary "key-image". The key-image is either a bit plane or an edge map generated from another image, which has the same size as the original image to be encrypted. In addition, we introduce two new lossless image encryption algorithms using this key-image technique. The performance of these algorithms is discussed against common attacks such as the brute force attack, ciphertext attacks and plaintext attacks. The analysis and experimental results show that the proposed algorithms can fully encrypt all types of images. This makes them suitable for securing multimedia applications and shows they have the potential to be used to secure communications in a variety of wired/wireless scenarios and real-time application such as mobile phone services.**

***Keywords*—image encryption, key-image, bit plane, edge map, brute force attack, chipertext attack, plaintext attack**

I. INTRODUCTION

Network technologies and media services provide ubiquitous conveniences for individuals and organizations to collect, share, or distribute images/videos in multimedia networks and wireless or mobile public channels. Image security is a major challenge in storage and transmission applications. For example, video surveillance systems for homeland security purposes are used to monitor many strategic places such as public transportation, commercial and financial centers. Large amounts of videos and images with private information are generated, transmitted, or restored every day. In addition, 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 security for these images and videos becomes an important issue for individuals, business and governments as well. Moreover, applications in the automobile, medical, construction and fashion industry require designs, scanned data, and blue-prints to be protected against espionage. Considering the long lifetime of image in the afore-mentioned domains, it is imperative to develop and employ techniques which protect the content throughout their lifetime [1]. Image encryption is an effective approach to protect images or videos by transforming them into completely different formats.

Several interesting approaches for image encryption have been developed. One method based on the cryptography concept considers images as data blocks or streams. It encrypts images block by block or stream by stream using different techniques. Data Encryption Standard (DES) [2] and Advanced Encryption Standard (AES) [3] are two examples of this approach. However, such encryption methods incur large computational costs and show poor error resilience [4].

Image encryption can be accomplished by scrambling image pixel positions using different techniques in the spatial domain [5-7]. One example is the recursive sequence based image scrambling approach. It scrambles images using different recursive sequences such as the Fibonacci sequence [8], Cellular automata [9] and chaotic maps [10, 11]. Image encryption can also be accomplished by scrambling coefficient matrices/blocks in the transform domain [12, 13]. Nevertheless, these approaches have extremely low security levels due to the lack of security keys or the small key space. Furthermore, the permutation-only based encryption schemes are known to be vulnerable for plaintext attacks [14].

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]. The security level of this method is much lower because the results of its decomposition process and logic operations are predictable. It is not immune to plaintext attacks.

To achieve higher levels of security, one solution is to change image pixel values while scrambling the positions of image pixels or blocks using different techniques. In this paper, we introduce two new lossless image encryption algorithms using a new concept "key-image" which is a binary image with the same size as the original image to be encrypted. One algorithm, called the BitplaneCrypt, generates the key-image by extracting a binary bit plane from another new or existing image. The key image of the other algorithm, called EdgemapCrypt, 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 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 the bit planes is inverted. The resulting encrypted image can be obtained by applying a scrambling algorithm to the image from a combination of all bit planes.

The rest of this paper is organized as followed. Section II introduces the two image encryption algorithms. Experimental examples are provided in Section III to show the performance of the two algorithms for 2D and 3D image encryption. Section

IV addresses the cryptanalysis for the two algorithms. A conclusion is reached in Section V.

II. IMAGE ENCRYPTION ALGORITHMS

In this section, A binary image is introduced as a "key-image" with the same size as the original image to be encrypted. We also introduce two image encryption algorithms using this key-image. One is called the BitplaneCrypt algorithm, while the other is named as the EdgemapCrypt algorithm. Both of them can fully encrypt 2D and 3D images such as grayscale images, color images and medical images.

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 which changes the locations of image pixels or blocks.

A. The BitplaneCrypt algorithm

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. 1. It first generates the key-image by exacting the r^{th} bit plane of the selected image, where r is the location of the bit plane. The algorithm then decomposes the original image into its binary bit planes and performs an XOR operation between each of these bit planes and the key-image. Next, the order of bit planes is inverted. The algorithm combines the bit planes together. Finally, a select scrambling algorithm is applied to the image to obtain the final resulting encrypted image.

Algorithm-1	***The BitplaneCrypt Algorithm***
Input	The original 2D or 3D image to be encrypted
Step 1	Choose a new or existing image with the same size of the original image, (convert the image into 2D image if it is a 3D image.)
Step 2	Obtain the key-image by extract the r^{th} bit plane of the image in Step 1.
Step 3	Decompose the original image or each component of the 3D image into its binary bit planes
Step 4	Perform the XOR operation between the key-image and each bit plane in Step 3.
Step 5	Invert the order of all bit planes
Step 6	Combine all bit planes together to obtain the 2D image or components
Step 7	Scramble the resulting image or components in Step 6 using a selected scrambling method to generate the resulting encrypted image. (For the 3D image, scramble its 2D components one by one.)
Output	The encrypted 2D or 3D image

Figure 1. The BitplaneCrypt algorithm

Since the 3D image contains several 2D data matrices called 2D components, the 3D image encryption can be accomplished by encrypting all its 2D components one by one.

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 bit planes of this image. Any new or existing image scrambling method can be used in the BitplaneCrypt algorithm. Therefore, 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 chosen as the key-image and the security keys of the scrambling method if applicable.

The correct security keys should be provided to the authorized user to generate the key-image. In the decryption process, the user unscrambles the encrypted image using the corresponding scrambling algorithm and its security keys. It then decomposes the image into bit planes. Each bit plane is applied an XOR operation with the key-image. The order of bit planes is reverted to the original order. The original image can be reconstructed by combining all bit planes.

Similar to the encryption process, the original 3D image can be reconstructed by decoding its 2D components one by one.

B. The EdgemapCrypt algorithm

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. In this section, we introduce a new image encryption algorithm using an edge map which is called the EdgemapCrypt algorithm.

An edge map is considered as the key-image in this algorithm. Such edge map is generated from another different image with the same size as the original image using a specific edge detector with a selected threshold value.

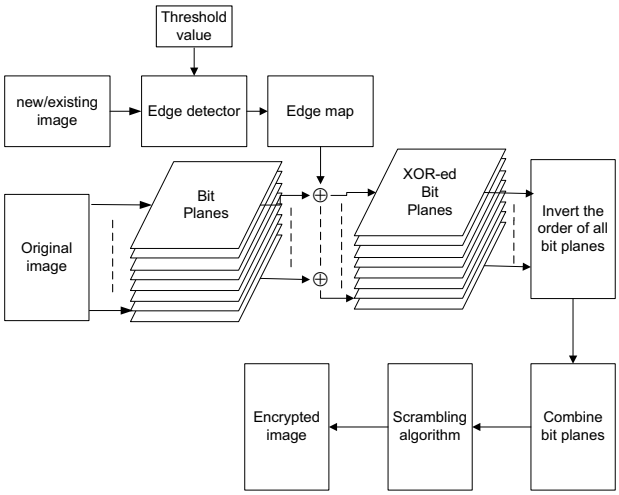

Figure 2. The EdgemapCrypt algorithm

The EdgemapCrypt algorithm first decomposes the original image into its binary bit planes. Each of them is encrypted by performing an XOR operation with the key-image, which is an edge map created from another image. Next, the algorithm inverts the order of all XORed bit planes and combines them together. The resulting image is scrambled by using a selected

scrambling algorithm to generate the final resulting encrypted image. The EdgemapCrypt algorithm is illustrated in Fig. 2.

Similar to the BitplaneCrypt algorithm, a 3D image can be encrypted by applying the EdgemapCrypt algorithm to all its 2D components individually.

Any new or existing image with the same size of the original image can be used to generate the edge map, the key-image. It could be an image in the public online database or a new image generate by the users. The edge map can be obtained by using any existing edge detector such as Canny, Sobel, Prewitt, or any other edge detector. The users have flexibility to choose any existing 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. 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's threshold, and the security keys of the scrambling algorithm.

To reconstruct the original image, the users should be provided the security keys which help them to obtain the correct 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. The reconstructed 2D image/component can be obtained by combining all bit planes.

III. Experimental Results

The BitplaneCrypt and EdgemapCrypt algorithms have been successfully implemented in 18 different 2D and 3D images such as grayscale images, color images and medical images. Several simulation results are provided to show the performance of the algorithms for 2D and 3D image encryption.

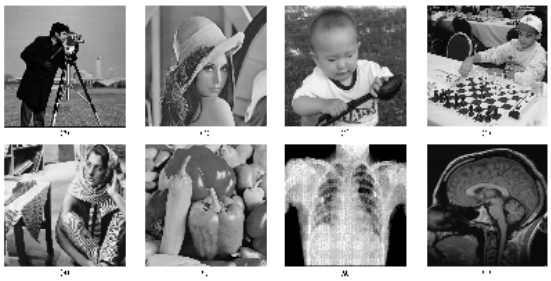

Figure 3. Test images. (a) 256×256 Cameraman; (b) 256×256 Lena; (c) 256×256 Baby; (d) 256×256 Chessplayer; (e) 512×512 Barbara; (f) 512×512 Peppers; (g) 512×512 CT ribs image; (h) 512×512 MRI brain image

In all experimental results of this paper, both algorithms utilize the image scrambling algorithm based on the Generalized P-Gray Code in [17] with the security keys: $n = 2, p = 0$. Fig. 3 shows several 2D images to be used as test images or images to generate the key-image.

A. 2D Image Encryption

There are several types of 2D images such as grayscale images, medical images and biometrics. The 2D image can be decomposed into several binary bit planes and encrypted one by one.

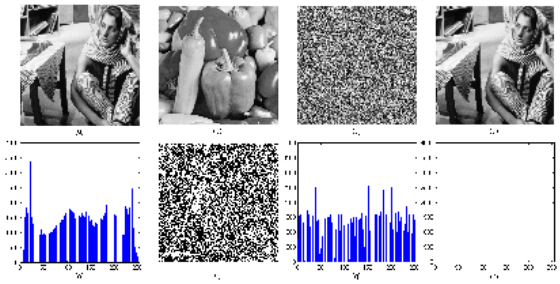

Figure 4. Grayscale image encryption using the BitplaneCrypt algorithm. (a) The original 512×512 grayscale image; (b) A 512×512 Peppers image; (c) The encrypted image; (d) The reconstructed image; (e) Histogram of the original image in (a); (f) The 5th bit plane of the Peppers image in (b); (g) Histogram of the encrypted image in (c); (h) Histogram of the difference between (d) and (a).

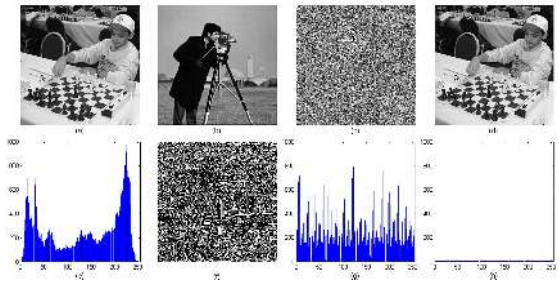

Figure 5. Grayscale image encryption using the EdgemapCrypt algorithm. (a) The original 256×256 grayscale image; (b) A 256×256 Cameraman image; (c) The encrypted image; (d) The reconstructed image; (e) Histogram of the original image in (a); (f) The edge map of the Cameraman image in (b), Sobel, 0.3; (g) Histogram of the encrypted image in (c); (h) Histogram of the difference between (d) and (a).

Fig. 4 shows an example of grayscale image encryption using the BitplaneCrypt algorithm. The key-image in this example is the 5th bit plane of a 512×512 grayscale Peppers image. Fig. 5 shows a result of grayscale image encryption using the EdgemapCrypt algorithm. The key-image is obtained from a 256×256 grayscale Cameraman image using the Sobel edge detector with a threshold 0.3.

From these results, the original images are fully encrypted as shown in Fig. 4(c) and Fig. 5(c). The distributions of the pixel values of the encrypted images are almost equal in grayscale value range as shown in Fig. 4 (g) and Fig. 5(g). This is one advantage of the presented algorithms. The original images are completely reconstructed. These can be verified by the reconstructed images in Fig. 4(d) and Fig. 5(d) and their histogram of the difference between the original image and the reconstructed image in Fig. 4(h) and Fig. 5(h).

The medical image encryption examples using the BitplaneCrypt and EdgemapCrypt algorithms are shown in Fig. 6 and Fig. 7, respectively. The key-image of the BitplaneCrypt algorithm in Fig. 6 is the 7th bit plane of a 512×512 grayscale

Barbara image. The key-image of the EdgemapCrypt algorithm in Fig. 7 is generated from a 512×512 grayscale Peppers image using a Prewitt edge detector with a threshold 0.2. The original medical images are also fully encrypted and completely reconstructed. This full encryption can be demonstrated by the encrypted image in Fig. 6(c) and Fig. 7 (c) and their histograms in Fig. 6(g) and Fig. 7(g), individually. The perfect reconstruction can be verified by the reconstructed images in Fig. 6(d) and Fig. 7(d) and their histograms in Fig. 6(h) and Fig. 7(h), respectively. All these results prove that the presented algorithms are lossless encryption methods.

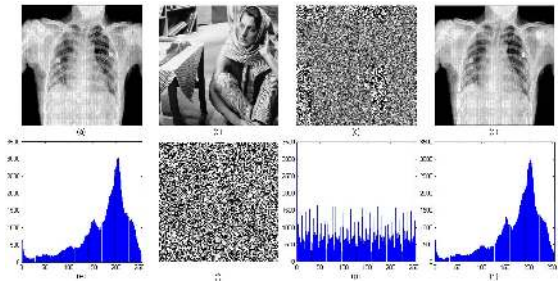

Figure 6. Medical image encryption using the BitplaneCrypt algorithm. (a) The original 512×512 CT ribs image; (b) A 512×512 Barbara image; (c) The encrypted image; (d) The reconstructed image; (e) Histogram of the original image in (a); (f) The 7th bit plane of the Barbara image in (b); (g) Histogram of the encrypted image in (c); (h) Histogram of the reconstructed image in (d).

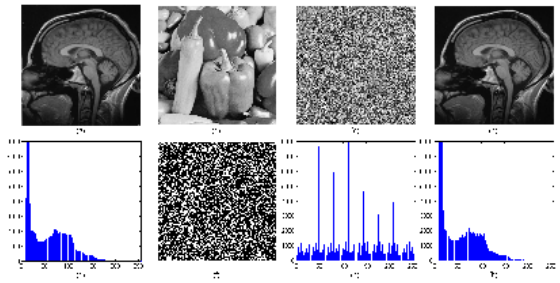

Figure 7. Medical image encryption using the EdgemapCrypt algorithm. (a) The original 512×512 MRI brain image; (b) A 512×512 Peppers image; (c) The encrypted image; (d) The reconstructed image; (e) Histogram of the original image in (a); (f) The edge map of the Peppers image in (b), Prewitt, 0.2; (g) Histogram of the encrypted image in (c); (h) Histogram of the reconstructed image in (d).

B. 3D Image Encryption

The 3D images, such as color images and 3D medical images contain several 2D components. Each component can be considered as a 2D image. The 3D image encryption using the presented algorithms can be accomplished by encrypting all the 2D components one by one.

Fig. 8 and Fig. 9 show the examples of color image encryption using the BitplaneCrypt and EdgemapCrypt algorithms, separately. The key-image in Fig. 8 uses the 4th bit plane of a 512×512 grayscale Chessplayer image. The key image in Fig. 9 is an edge map generated from a 512×512 grayscale Barbara image using Canny edge detector with threshold 0.1.

The results show that the color images are fully encrypted and then completely reconstructed. The histograms in Fig. 8(g) and Fig. 9(g) also verified the distributions of the encryption images are equal in the data level range. The reconstructed images in Fig. 8(d) and Fig. 9(d) and their histograms in Fig. 8(h) and Fig. 9(h) demonstrate the complete reconstruction of the original images. These further prove that the BitplaneCrypt and EdgemapCrypt algorithms are lossless encryption methods.

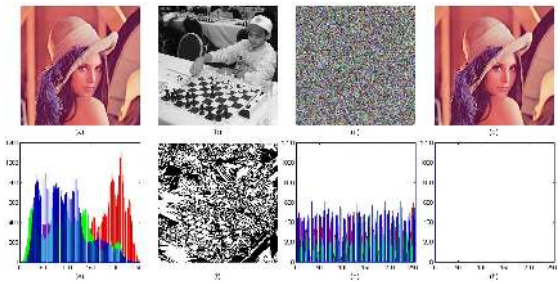

Figure 8. Color image encryption using the BitplaneCrypt algorithm. (a) The original 256×256 color image; (b) A 256×256 grayscale Chessplayer image; (c) The encrypted color image; (d) The reconstructed color image; (e) Histogram of the original image in (a); (f) The 4th bit plane of the Chessplayer image in (b); (g) Histogram of the encrypted image in (c); (h) Histogram of the difference between (d) and (a).

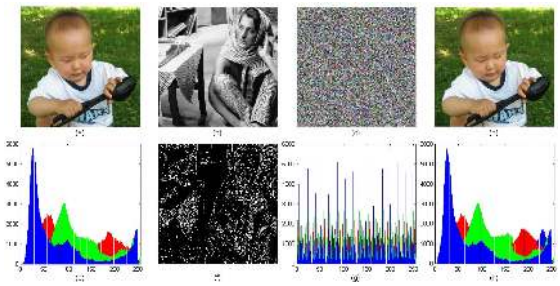

Figure 9. Color image encryption using the EdgemapCrypt algorithm. (a) The original 512×512 color image; (b) A 512×512 grayscale Barbara image; (c) The encrypted color image; (d) The reconstructed color image; (e) Histogram of the original image in (a); (f) The edge map of the Barbara image in (b), Canny, 0.1; (g) Histogram of the encrypted image in (c); (h) Histogram of the recontructed image in (d).

IV. SECURITY ANALYSIS

Security is important for both the encrypted objects and the encryption algorithms. We discuss some security issues of the BitplaneCrypt and EdgemapCrypt algorithms from the cryptography point of view in this section.

A. Security Key Space

As the discussion in Section II, the security keys of the BitplaneCrypt algorithm are the combination of the image or the location 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. On the other hand, the security keys of the EdgemapCrypt 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's threshold, and the security keys of the scrambling algorithm.

The combination of the security keys is extremely important for both presented algorithms. The original image can be completely reconstructed without any distortion only when the correct security keys are being utilized. This can be verified by the reconstructed images in Fig. 10(b) and Fig. 11(b) and their histograms in Fig. 10 (f) and Fig. 11(f). Otherwise, the reconstructed images cannot be recognized as shown in Fig. 10 (c), (d) and Fig. 11 (c), (d).

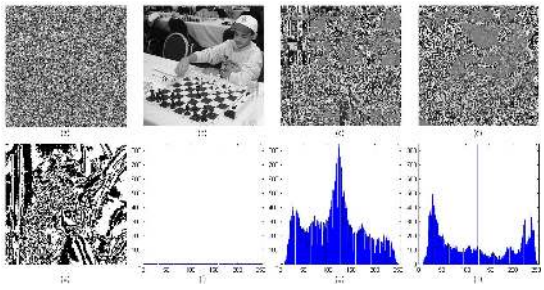

Figure 10. Grayscale image decryption using the BitplaneCrypt algorithm with different security keys. (a) The encrypted 256×256 grayscale Chessplayer image with security keys: the 4th bit plane of the 256×256 grayscale Lena image and $n=2, p=0$ for the scrambling algorithm; (b) The reconstructed grayscale image using the correct secruity keys; (c) The reconstructed grayscale image using the same key-image and $n=2, p=2$ for the scrambling algorithm; (d) The reconstructed grayscale image using the 7th bit plane of the 256×256 grayscale Lena image and the same security keys for the scrambling algorithm; (e) the key-image: the 4th bit plane of the 256×256 grayscale Lena image; (f) Histogram of the difference between the original image and the reconstructed image in (b); (g) Histogram of the difference between the original image and the reconstructed image in (c); (h) Histogram of the difference between the original image and the reconstructed image in (d).

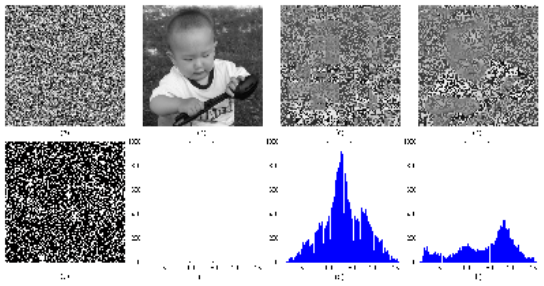

Figure 11. Grayscale image decryption using the EdgemapCrypt algorithm with different security keys. (a) The encrypted 256×256 grayscale Baby image with security keys: the 256×256 grayscale Chessplayer image, Prewitt, 0.5, and $n=2, p=0$ for the scrambling algorithm; (b) The reconstructed grayscale image using the correct secruity keys; (c) The reconstructed grayscale image using the same key-image and $n=2, p=1$ for the scrambling algorithm; (d) The reconstructed grayscale image using the security keys: the 256×256 grayscale Cameramean image, Sobel, 0.3, and the same security keys for the scrambling algorithm; (e) the key-image: the edge map of the 256×256 grayscale Chessplayer image, Prewitt, 0.5; (f) Histogram of the difference between the original image and the reconstructed image in (b); (g) Histogram of the difference between the original image and the reconstructed image in (c); (h) Histogram of the difference between the original image and the reconstructed image in (d).

Any new or existing image with the same size as the original image can be used to generate the key-image for both algorithms. It has a huge numbers of possible choices, assuming P_I. Each of its bit planes can be used as a key-image for the BitplaneCrypt algorithm. The number of possible choices of the key-image for this algorithm is $8P_I$ for its gray levels within 0-255. In addition, any new or existing image scrambling algorithm can be used to scramble the bit planes in both algorithms. The security keys of the selected image scrambling algorithm are also part of combinations of the security keys for the presented algorithms, assuming their possible choices are P_S which is not more than $M!N!$ if the original image is an $M \times N$ grayscale image. Thus, the security key space of the BitplaneCrypt algorithm for an $M \times N$ grayscale image with is $8P_IP_S$.

Moreover, any new or existing edge detector can be used in the EdgemapCrypt algorithm, assuming its possible choice is P_E. The edge detector's threshold is rational number within 0 to 1. However, not all the threshold values can achieve a desirable encryption result. The number of their possible choices may not be infinite, assuming P_{TH}. The security key space for the EdgemapCrypt algorithm is $P_IP_EP_{TH}P_S$.

B. Brute Force Attack

The Brute force attack is an attack model in which the attacker tries to guess the security keys by conducting an exhaustive search of all the possible combinations of security keys of the encryption algorithms. Theoretically, this approach is feasible if the key space of the encryption algorithm is limited and the attacker knows the encryption algorithm.

Even if the security key spaces of both algorithms are not infinite, they are still sufficiently large since the large number of possible new/existing images can be used to generate the key-image. As a result, the two algorithms can withstand the brute force attack.

C. Ciphertext-only Attack

In cryptography, the plaintext is the original information to be encrypted. The ciphertext is the encrypted plaintext.

The ciphertext-only attack is an attack model in which an attacker tries to deduce the security keys by only studying the ciphertext [18]. This attack can be used to recover the original image data by studying the encrypted images. If fewer portions of the images are encrypted, more portions of the original images can be recovered by an attacker without knowing the encryption algorithm and its security keys. An encryption scheme has an extremely low security level if it cannot withstand this attack.

From the experimental results in Section III, the encrypted images are visually unrecognizable and totally different from the original images. They contain almost no visual information of the original images. The distributions of the encrypted images are equal in their histograms. These ensure the BitplaneCrypt and EdgemapCrypt algorithms can withstand the cipher-only attack.

D. Known-Plaintext Attack

The known-plaintext attack is an attack model in which an attacker tries to obtain the security keys of encryption algorithm by studying a number of plaintexts and the corresponding ciphertexts [18]. The condition of this attack is

that the attacker should have some plaintexts and the corresponding ciphertext. It is possible for the attacker to partially or completely break the encrypted image without knowing the encryption algorithm and its security keys if the encryption process does not change the image data.

The XOR operation and inverting the order of the bit planes in the BitplaneCrypt and EdgemapCrypt algorithms are designed to change image data. The image scrambling algorithm is used to change image pixel positions. These make the encrypted image data are not useful for the attacker using this type of attack. Thus, both algorithms can withstand the known-plaintext attack.

E. Chosen-Ciphertext Attack

The chosen-ciphertext attack is an attack model in which the attacker can choose some ciphertexts and their corresponding plaintexts [18]. Therefore, the attacker can deduce the security keys in encryption algorithms, or recover the original plaintext from the unseen ciphertext. 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 analysis above, the presented algorithms can also withstand the chosen-ciphertext attack because both image data and pixel locations are changed during the encryption process.

F. Chosen-Plaintext Attack

The chosen-plaintext attack is an attack model in which the attacker can choose a number of plaintexts and then deduce their corresponding ciphertexts [18]. As a result, the attacker can choose any useful information as plaintext in order to deduce the security keys of encryption algorithms, or reconstruct the original plaintexts from the unknown ciphertexts. The attack can break the encrypted image without knowing the encryption algorithm and its security keys, if the image data does not change during the encryption process.

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

V. CONCLUSION

In this paper, we have introduced a new concept for image encryption using a binary key-image. We also introduced two image encryption algorithms based on this key-image. The key-image is either a bit plane in the BitplaneCrypt algorithm or an edge map in the EdgemapCrypt algorithm.

Experiments have demonstrated that both algorithms can fully encrypt the 2D and 3D images. The original 2D and 3D images can also be completely reconstructed without any distortion. Cryptanalysis has shown that the algorithms have extremely large security key space and can withstand most common attacks such as the brute force attack, cipher attacks and plaintext attacks.

Any new or existing image with the same size as the original image can be used to generate the key-image. All edge detectors with any specified threshold value can be used to create the edge map as a key-image for the EdgemapCrypt algorithm. Any existing image scrambling method can be applied to these two presented algorithms. All these ensure the images can be protected with a higher security level.

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

REFERENCES

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

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

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

[4] B. B. Zhu, M. D. Swanson, and S. Li, "Encryption and authentication for scalable multimedia: current state of the art and challenges," in Internet Multimedia Management Systems V, Philadelphia, PA, USA, 2004, pp. 157-170.

[5] 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.

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

[7] Y. Zhou, S. Agaian, 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.

[8] 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. Proceedings of the 2004 International Symposium on, 2004, pp. III-965-8 Vol.3.

[9] 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.

[10] J. C. Yen and J. I. Guo, "Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation," Vision, Image and Signal Processing, IEE Proceedings -, vol. 147, no. 2, pp. 167-175, 2000.

[11] Z. H. Guan, F. J. Huang, and W. J. Guan, "Chaos-based image encryption algorithm," Physics Letters A, vol. 346, no. 1-3, pp. 153-157, Oct 2005.

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

[13] 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.

[14] 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.

[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.

[17] Y. Zhou, K. Panetta, and S. Agaian, "Partial Multimedia Encryption with Different Security Levels," in Technologies for Homeland Security, 2008 IEEE Conference on, 2008, pp. 513-518.

[18] A. J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone Handbook of Applied Cryptography. New York: CRC Press, Inc., 1997.