International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 2, No. 4, October 2013

Research Paper

IMAGE ENCRYPTION USING COLOR KEY IMAGES

S Reddy Jyoteeswara Prasad[1]* and R V S Sathyanarayana[1]

*Corresponding Author: S Reddy Jyoteeswara Prasad, ✉ jyoteeswar.prasad@gmail.com

This paper introduces a new concept for image encryption using binary key images. Here the original 2D (Grayscale image) or 3D (Color) image is encrypted with 2D or 3D key images (key-image is either color or a grayscale image). A binary image is introduced as a "keyimage" 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 key image is either a Bitplane or Edgemap generated from the image, 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. 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. 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. 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.

Keywords: Loss less image encryption, Key-image, Edge map, Plaintext attack, Brute force attack, Ciphertext attack

INTRODUCTION

Image security is a major challenge in storage and transmission applications. 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

[1] S V University College of Engineering, Tirupati, Andhra Pradesh, India.

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) (1999) and Advanced Encryption Standard (AES) (2001) are two examples of this approach. However, such encryption methods incur large computational costs and show poor error resilience. Several data encryption algorithms like chaos based (Cheng and Gui, 2000) and combinatorial permutations (Li *et al.*, 2008) are proposed for encrypting images. Image encryption can be accomplished by scrambling image pixel positions using different techniques in the spatial domain (Yang and Bourbakis, 2004; Ashtiyani *et al.*, 2008; and Zhou *et al.*, 2008). is the recursive sequence based image scrambling approach. It scrambles images using different recursive sequences such as the Fibonacci sequence (Zou *et al.*, 2004), Cellular automata (Chen and Lai, 2007) and chaotic maps (Yen and Geo, 2000; and Guan *et al.*, 2005). Image encryption can also be accomplished by scrambling coefficient matrices/blocks in the transform domain (Li *et al.*, 2005; and Gu and Han, 2006). 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 (Han *et al.*, 1999; and Lukac and Plataniotis, 2005). 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.

Figure 1: Bitplane Crypt Algorithm

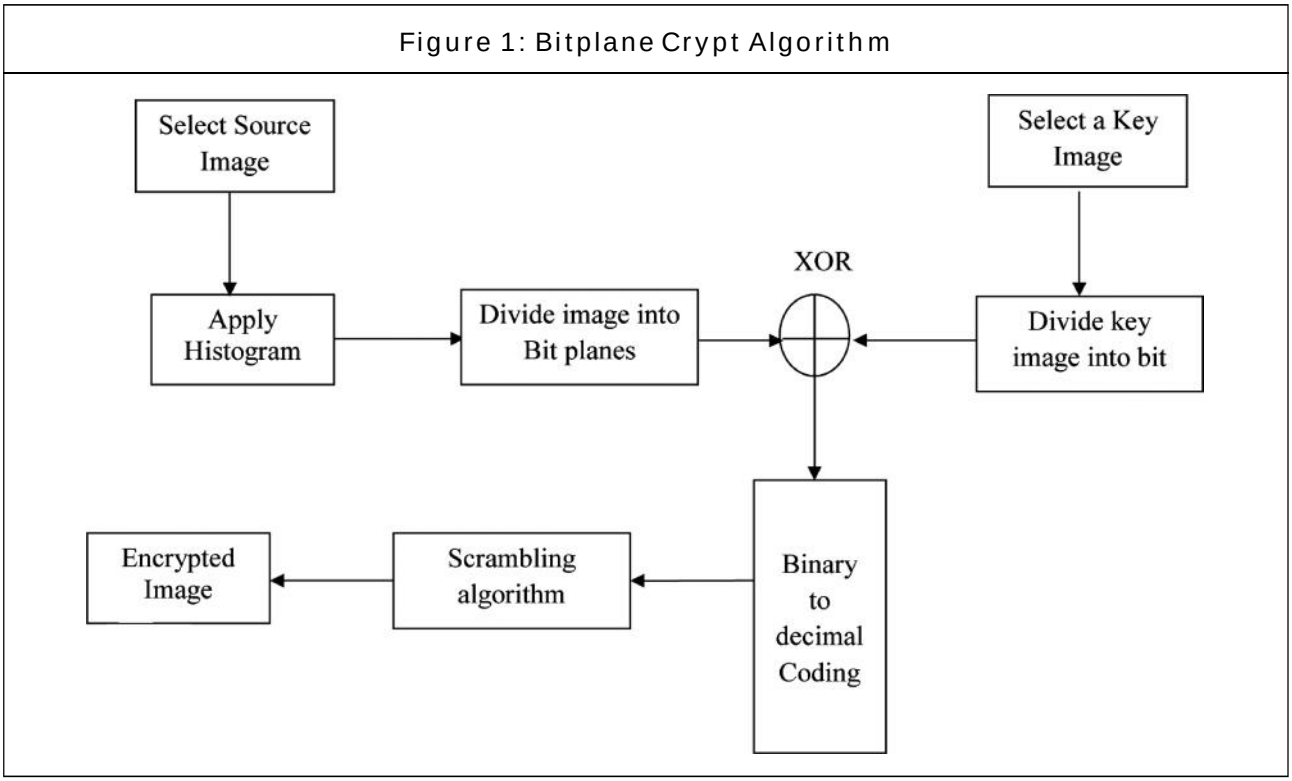


IMAGE ENCRYPTION ALGORITHMS

In this section, a binary image is introduced as a "keyimage" 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.

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 Figure 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. Then convert those bit planes from binary to decimal. The algorithm combines the bit planes together. Finally, a select scrambling algorithm is applied to the image to obtain the final resulting encrypted image.

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 keyimage 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.

BitplaneCrypt Algorithm

Input The original 2D or 3D image to been crypted.

Step 1 Choose anewor existing image with the same size of the original image.

Step 2 Obtain the key-image by extract the r^{th} bit plane of the image in Step1.

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 Convert the bit planes from binary to decimal.

Step 7 Combine all bit planes together to obtain the 2D image or components.

Step8 Scramble the resulting image or components in Step 6 using a selected scrambling method to generate the resulting encrypted image.

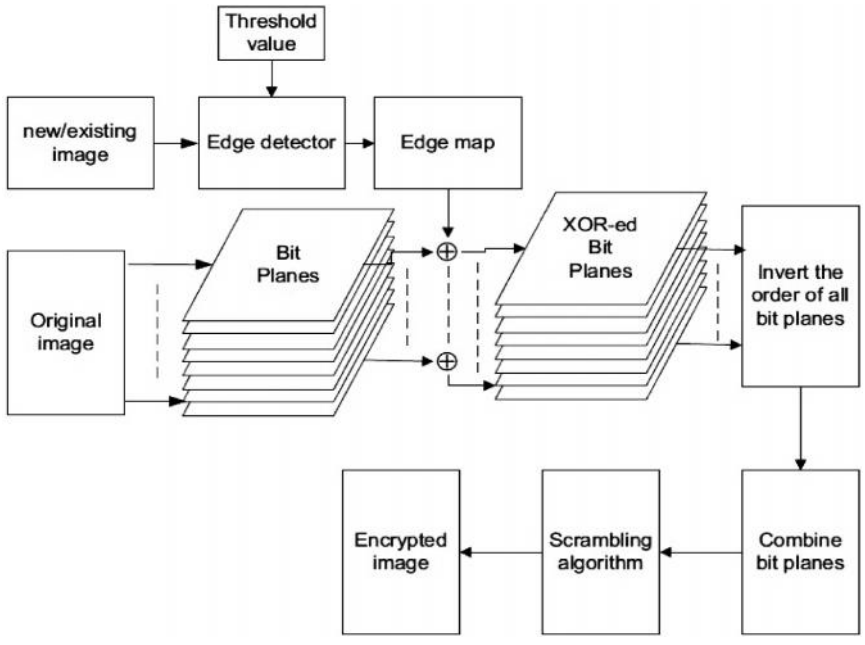
Output The encrypted 2D or 3D image.

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 differentimage with the same size as the original image using a specific edge detector with a selected threshold value.

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 Figure 2. Similar to

Figure 2: The EdgemapCrypt Algorithm



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 theoriginal image can be used to generate the edge map, the keyimage.

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.

Edgemap Crypt Algorithm

Input The original 2D or 3D image to been crypted.

Step 1 Choose anewor existing image with the same size of the original image.

Step 2 Obtain the key-image by extract the r^{th} bit plane of the image in Step 1.

Step 3 Calculate the edge map from the existing image with the same size of the original image.

Step 4 Decompose the original image or each component of the 3D image into its binary bit planes.

Step 5 Perform the XOR operation between the edge map of 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 image.

Step 8 Scramble the resulting image or components in Step 7 using a selected scrambling method to generate the resulting encrypted image.

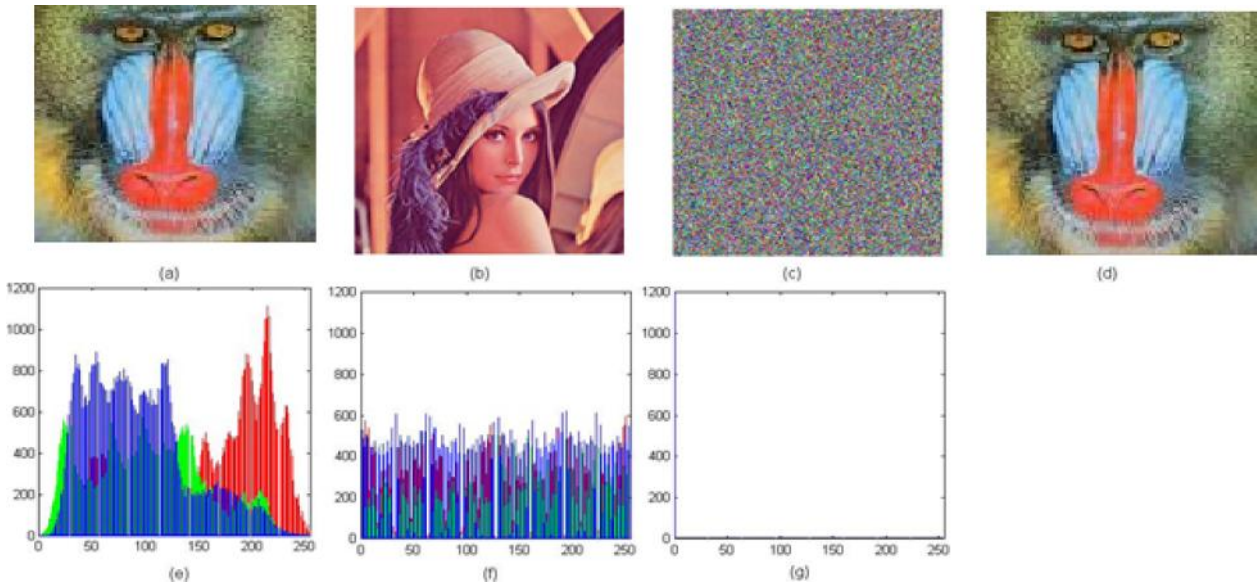
Output The encrypted 2D or 3D image.

EXPERIMENTAL RESULTS

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. Figures 3 and 4 show the examples of color image encryption using the BitplaneCrypt and EdgemapCrypt algorithms, separately. The key-image in Figure 3 uses the 4^{th} bit plane of the image. The key image in Figure 4 is an edge map generated from salt and pepper image using

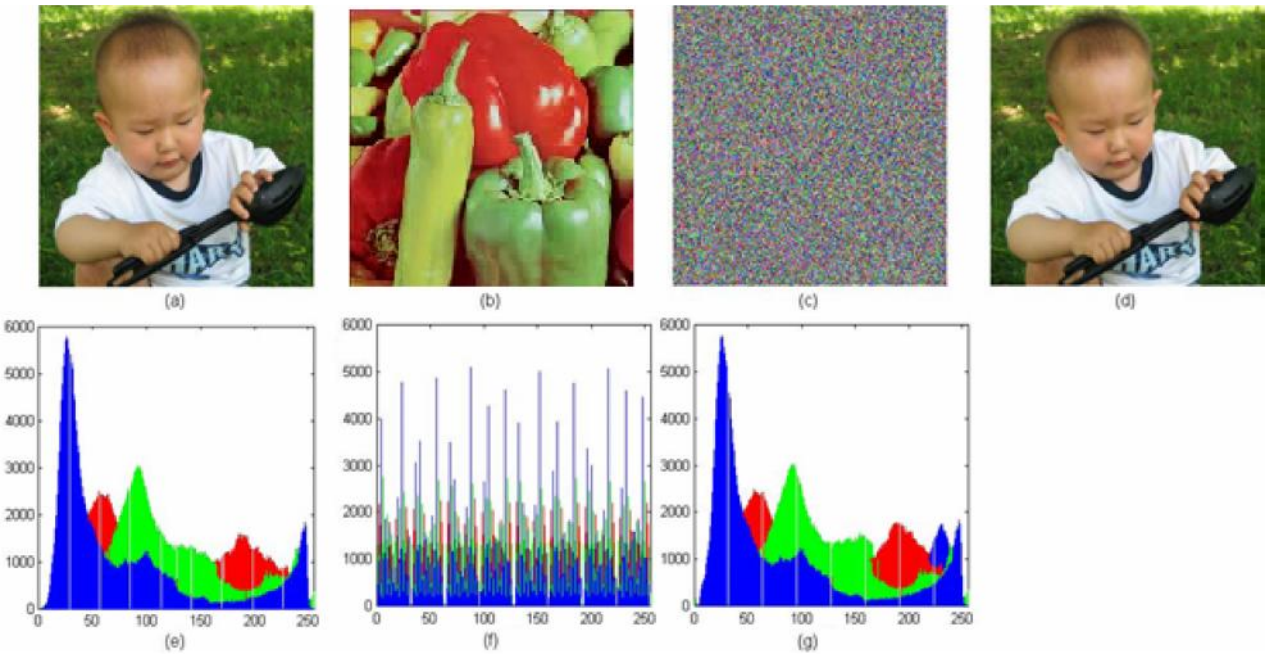
Figure 3: Color Image Encryption Using the BitplaneCrypt Algorithm



Note: (a) The original 256×256 color image; (b) A 256×256 color key image; (c) The encrypted color image; (d) The reconstructed color image; (e) Histogram of the original image in (a); (f) Histogram of the encrypted image in (c); (g) Histogram of the difference between (d) and (a).

Canny edge detector with threshold 0.1. The results show that the color images are fully encrypted and then completely reconstructed. The histograms in Figure 3f and Figure 4f also

Figure 4: Color Image Encryption Using the EdgemapCrypt Algorithm



Note: (a) The original 512×512 color image; (b) A 512×512 color key image; (c) The encrypted color image; (d) The reconstructed color image; (e) Histogram of the original image in (a); (f) Histogram of the encrypted image in (c); (g) Histogram of the recontructed image in (d).

verified the distributions of the encryption images are equal in the data level range. The reconstructed images in Figure 3d and Figure 4d and their histograms in Figure 3g and Figure 4g demonstrate the complete reconstruction of the original images. These further prove that the BitplaneCrypt and EdgemapCrypt algorithms are lossless encryption methods.

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.

Security Key Space

As the discussion in Section I, 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.

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.

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 . 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 II, 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.

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. 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.

Chosen-Ciphertext Attack

The chosen-ciphertext attack is an attack model in which the attacker can choose some ciphertexts and their corresponding plaintexts. 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.

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. As a result, the attackercan 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 theimage data does not change during the encryption process. Both the BitplaneCrypt and EdgemapCrypt algorithms change the image data and pixel locations. They can withstandthe chosen-plaintext attack.

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. Advanced Encryption Standards (2001), National Institute of Standards and Technology, available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
2. Ashtiyani M, Birgani P M and Hosseini H M (2008), "Chaos-Based Medical Image Encryption Using Symmetric Cryptography", *ICTTA*, pp. 1-5.
3. Cheng J and Gui J I (2000), "A New Chaotic Key-Based Design for Image Encryption and Decryption", The 2000 IEEE International Symposium on Circuits and Systems, Proceedings, ISCAS Geneva, Vol. 4, pp. 49-52.
4. Chen R J and Lai J L (2007), "Image Security System Using Recursive Cellular Automata Substitution", *Pattern Recognition*, Vol. 40, No. 5, pp. 1621-1631.
5. Data Encryption Standard (1999), National Institute of Standards and Technology, available at http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
6. Guan Z H, Huang F J and Guan W J (2005), "Chaos-Based Image Encryption Algorithm", *Physical Letters A*, Vol. 346, Nos. 1-3, pp. 153-157.
7. Gu G S and Han G Q (2006), "The Applications of Chaos and DWT in Image Scrambling", in *Machine Learning and Cybernetics*, International Conference on, pp. 3729-3733.
8. Han J W, Park C-S, Ryu D-H and Kim E-S (1999), "Optical Image Encryption Based on XOR Operations", *Optical Engineering*, Vol. 38, No. 1, pp. 47-54.
9. lyer K C and Subramanya A (2009), "Image Encryption by Pixel Property Separation", *Cryptology*.
10. Li S, Li C, Chen G, BourBakis N G and Lo K-T (2008), "A General Quantitative Cryptanalysis of Permutation-Only Multimedia Ciphers Against Plaintext Attacks", *Signal Processing: Image Communication*, Vol. 23, No. 3, pp. 212-223.
11. Li T, Zhou S, Zeng Z and Ou Q (2005), "A New Scrambling Method Based on Semi-Frequency Domain and Chaotic System", in *Neural Networks and Brain*, ICNN&'05, International Conference, pp. 607-610.
12. Lukac R and Plataniotis K N (2005), "Bit Level Based Secret Sharing for Image Encryption", *Pattern Recognition*, Vol. 38, No. 5, pp. 767-772.
13. Yang M and Bourbakis N (2004), "Data-Image-Video Encryption", *Potential, IEEE*, Vol. 23, No. 3, pp. 28-34.
14. Yen J C and Geo J I (2000), "Efficient Hierarchical Chaotic Image Encryption Algorithm and its VLSI Realization", Vision, Image and Signal Processing, IEEE Proceeding, Vol. 147, No. 2, pp. 167-175.

15. Zhou Y, Again S, Joyner V M and Panetta K (2008), "Two Fibonacci p-Code Based Image Scrambling Algorithms", in Image Processing: Algorithms and Systems VI, San Jose, CA, USA.

16. Zou J, Ward R K and Qi D (2004), "A New Digital Image Scrambling Method Based on Fibonacci Numbers", in *Circuits and Systems*, ISCAS'04, Vol. 3, pp. 965-968.