Selective Object Encryption for Privacy Protection

Yicong Zhou*[a], Karen Panetta[a], Ravindranath Cherukuri[b], Sos Agaian[b]
[a]Department of Electrical and Computer Engineering, Tufts University, Medford, MA USA 02155;
[b]Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX USA 78249

ABSTRACT

This paper introduces a new recursive sequence called the truncated P-Fibonacci sequence, its corresponding binary code called the truncated Fibonacci p-code and a new bit-plane decomposition method using the truncated Fibonacci p-code. In addition, a new lossless image encryption algorithm is presented that can encrypt a selected object using this new decomposition method for privacy protection. The user has the flexibility (1) to define the object to be protected as an object in an image or in a specific part of the image, a selected region of an image, or an entire image, (2) to utilize any new or existing method for edge detection or segmentation to extract the selected object from an image or a specific part/region of the image, (3) to select any new or existing method for the shuffling process. The algorithm can be used in many different areas such as wireless networking, mobile phone services and applications in homeland security and medical imaging. Simulation results and analysis verify that the algorithm shows good performance in object/image encryption and can withstand plaintext attacks.

Keywords: Image encryption, object encryption, truncated P-Fibonacci sequence, truncated Fibonacci p-code, truncated Fibonacci p-code decomposition

1. INTRODUCTION

Video surveillance systems for homeland security purposes are used to monitor many strategic places such as public transportation hubs, airport security, commercial and financial centers. Large amounts of videos and images with private information are generated. The potential exists for the loss of privacy and information abuse. Privacy protection becomes an important issue for these monitoring systems. One solution is to encrypt the selected object in a video/image to ensure privacy protection while allowing decryption for legitimate security needs at anytime. Privacy is preserved since the selective object encryption allows monitoring the activities without knowing the identities monitored in the videos. When a suspicious activity needs to be investigated, the identities can be uncovered with proper authorization [1].

Privacy protection is also important in many other areas. Biometrics authentication systems use personal biological or behavioral characteristics, such as face, fingerprint and signatures, to verify the user identity. The security of biometric data is particularly important because these biometric data not only serve as security keys for the authentication systems but also contain private information. Medical image transmission through wired and wireless networks allows different doctors the ability to digitally access the medical records of a patient. Hence, there is an urgent need to provide the confidentiality of medical image data related to the patient when medical image data is stored in databases and transmitted over networks. [2-4].

To protect privacy, several interesting approaches have been developed such as concealing regions of interest (ROIs) by scrambling the sign of selected transform coefficients in the transform-domain [5], Encrypting biometric images using Fractional Fourier transform [6], using a track-based system for human movement analysis and privacy protection adaptive to environmental contexts [7], de-identifying face images[8] and many others. These methods are good contributions to privacy protection for specific applications. However, a universal security method is required.

In this paper, we introduce a new lossless image encryption algorithm based on the truncated Fibonacci p-code decomposition. It is suitable for all types of images or objects in any kind of images for privacy protection in real-time applications. It protects privacy by encrypting a selected object in either a part of an image or the entire image, or encrypting the whole image, achieving different security needs in practical applications.

* Yicong.Zhou@tufts.edu; phone 1-617-627-5183; fax 1-617-627-3220

Mobile Multimedia/Image Processing, Security, and Applications 2009, edited by Sos S. Agaian, Sabah A. Jassim,
Proc. of SPIE Vol. 7351, 73510F · © 2009 SPIE · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.817699

The algorithm first generates a boundary mask of the selected object by using a segmentation algorithm or an edge detection method such as Canny, Sobel, or the shape-adaptive DCT [9, 10]. It then separates the image into the selected object and the image without the selected object based on this boundary mask. The selected object is encrypted using the truncated Fibonacci p-code decomposition method and bit plane shuffling. Finally, the encrypted object is combined with the image without the selected object to obtain the resulting encrypted image.

2. TRUNCATED FIBONACCI P-CODE AND ITS DECOMPOSITION

In this section, we introduce a new recursive sequence called truncated P-Fibonacci sequence (TPFS). We also introduce its corresponding binary code, called truncated Fibonacci p-code (TFPC), and bit-plane decomposition, namely truncated Fibonacci p-code decomposition. This decomposition method is well suitable for image encryption because the truncated Fibonacci p-code and its decomposition results are parameter dependent.

2.1 Truncated P-Fibonacci sequence

Definition 2.1: The P-Fibonacci sequence is a recursive sequence defined by [11],

$$F_p(n) = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ F_p(n-1) + F_p(n-p-1) & n > 0 \end{cases} \quad (1)$$

where p is a non-negative integer.

Based on the definition above, P-Fibonacci sequence will differ based on the p value. Specially,

1) $p = 0$, the P-Fibonacci sequence is geometric progression increasing by two, 1, 2, 4, 8, 16…;

2) $p = 1$, the P-Fibonacci sequence is the classical Fibonacci sequence is 1, 1, 2, 3, 5, 8, 13, 21…;

3) For the large values of p, the P-Fibonacci sequence starts with successive 1's and immediately after that 1, 2, 3, 4 … p …

Some examples are given in Table 1.

Table 1. P-Fibonacci sequences with different p values.

P \ n	0	1	2	3	4	5	6	7	8	…
0	1	2	4	8	16	32	64	128	256	…
1	1	1	2	3	5	8	13	21	34	…
2	1	1	1	2	3	4	6	9	13	…
3	1	1	1	1	2	3	4	5	7	…
4	1	1	1	1	1	2	3	4	5	…
…	…									
∞	1	1	1	1	1	1	1	1	1	…

Table 2. Truncated P-Fibonacci sequences with different p values.

P \ n	0	1	2	3	4	5	6	7	8	…
0	1	2	4	8	16	32	64	128	256	…
1	1	2	3	5	8	13	21	34	55	…
2	1	2	3	4	6	9	13	19	28	…
3	1	2	3	4	5	7	10	14	19	…
4	1	2	3	4	5	6	8	11	15	…
…	…									

Definition 2.2: A recursive sequence called truncated P-Fibonacci sequence (TPFS) is defined as,

$$T_p(n) = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ F_p(n+p) & n > 0 \end{cases} \quad (2)$$

where $F_p(n+p)$ is P-Fibonacci sequence defined in equation (1).

The truncated P-Fibonacci sequence also changes with different p values. For example,

1) $p = 0$, the truncated P-Fibonacci sequence is geometric progression increasing by two, 1, 2, 4, 8, 16…;
2) $p = 1$, the truncated P-Fibonacci sequence is the truncated classical Fibonacci sequence is 1, 2, 3, 5, 8, 13, 21…;
3) $p = \infty$, the truncated P-Fibonacci sequence is a positive integer sequence, 1, 2, 3, 4, 5, 6, 7…

Some TPFS examples are also given in Table 2.

2.2 Truncated Fibonacci P-code

Definition 2.3: A non-negative decimal number can be represented by the following format,

$$A = c_0 T_p(0) + c_1 T_p(1) + \ldots + c_{n-1} T_p(n-1) \quad (3)$$

where n and p are nonnegative integers, $i = 0,1,\ldots,n-1$, $c_i \in (0,1)$, $T_p(i)$ is the TPFS defined in equation (2). The binary coefficient sequence $(c_{n-1},\ldots,c_1,c_0)$ is called the truncated Fibonacci P-code (TFPC) of A, namely,

$$A = (c_{n-1},\ldots,c_1,c_0)_p \quad (4)$$

For a certain p value, the TFPC of a specific decimal number is shorter than the Fibonacci p-code in [11]. This makes TFPC more efficient to be generated. Similar to the Fibonacci p-code, TFPC is also not unique. For example, if $A = 35$, $p = 4$, the TFPC of A will be,

$$35 = (1,0,0,0,0,0,0,0,0,0,1)_4 = (0,1,0,0,0,1,0,0,0,0,0,1)_4 = (0,1,0,0,0,0,1,0,0,1,0,0)_4 = \ldots$$

To obtain a unique TFPC of a non-negative decimal number for specific parameter p, the following condition should be satisfied,

$$A = T_p(n) + s \quad (5)$$

where $0 \le s < T_p(n-p)$.

The above condition is same as the constraint of the Fibonacci p-code in [11]. There are at least p 0's between two consecutive 1's in the unique TFPC of any non-negative decimal number. For the above example, its unique TFPC will be $35 = (1,0,0,0,0,0,0,0,0,0,1)_4$.

2.3 Truncated Fibonacci P-code Decomposition

For a certain p value, every non-negative decimal number has a unique TFPC representation if the condition in equation (5) is satisfied. Its TFPC will differ only based on different p values because the TFPS is specified by the parameter p values.

Based on the definition 2.3, a grayscale image can also be decomposed into the TFPC bit-planes called the Truncated Fibonacci P-code decomposition. The traditional image decomposition is a special case of the TFPC decomposition when $p = 0$.

A grayscale image with gray levels within 0-255 is decomposed into 12 TFPC bit planes when $p = 1$. A TFPC decomposition example is shown in Fig.1. For a specific grayscale image, the number of its TFPC bit-planes changes with parameter p values. For instance, the number of its TFPC bit-plane is 17 for $p = 3$, and 21 for $p = 5$ respectively.

Moreover, the contents of its TFPC decomposition results are also different based on different p values. This makes the TFPC decomposition a suitable method for image encryption. To show the difference between the TFPC decomposition and the Fibonacci p-code decomposition, Fig. 2 provides the decomposition result of the same grayscale image using the Fibonacci p-code with $p = 1$.

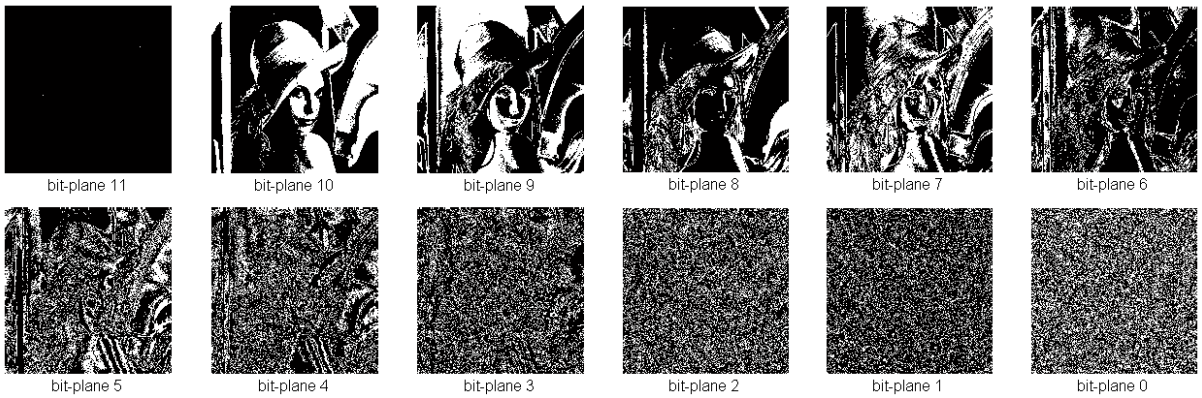

Fig. 1. Truncated Fibonacci P-code decomposition of the grayscale Lena image, p=1.

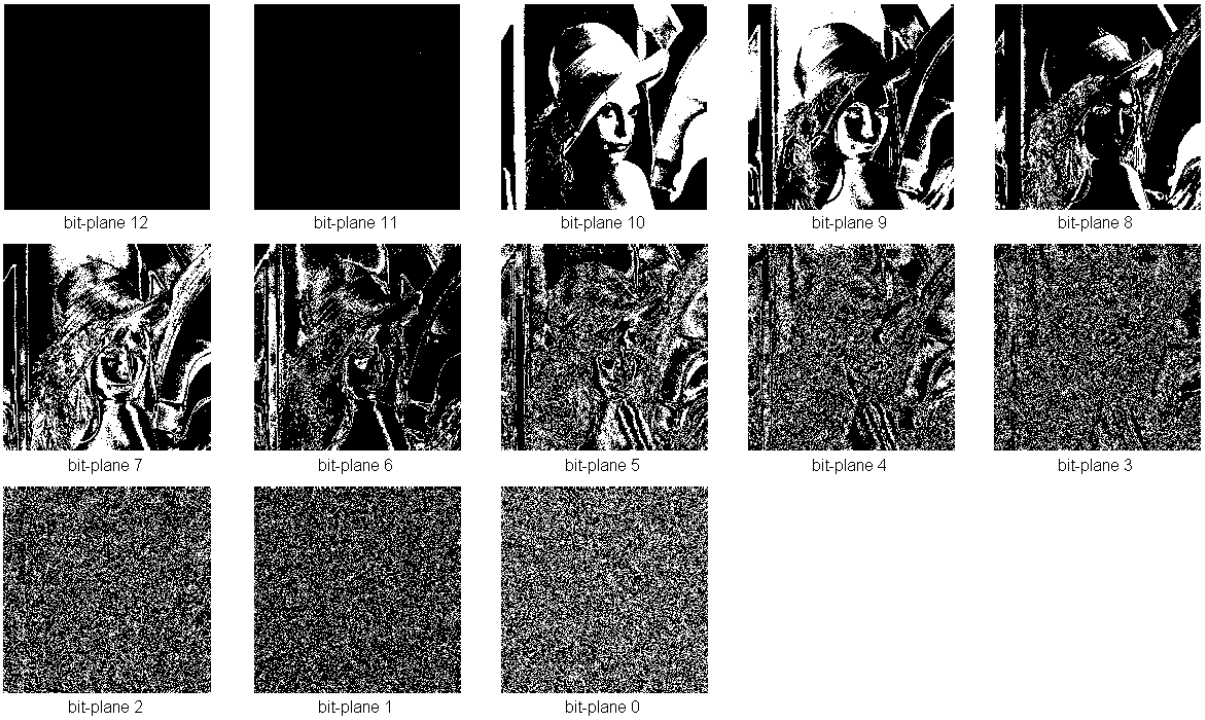

Fig. 2. Fibonacci P-code decomposition of the grayscale Lena image, p=1.

3. IMAGE ENCRYPTION ALGORITHM

The 2D image such as grayscale image, biometric image and medical image contain a 2D data matrix. To improve the speed of encryption process while protecting private information, one solution is to selectively encrypt the important part or region of an image. In this section, we introduce a new lossless encryption algorithm called "ObjectEcrypt" to encrypt a selected object. This selected object can be defined as either an object in an image or in a specific region of the image,

a selected part/region of an image, or an entire image. The ObjectEcrypt algorithm is shown in Fig. 3. It can be used in real-time applications such as wireless network and mobile phone services.

The ObjectEcrypt algorithm first creates a boundary mask of the selected object using an edge detector or segmentation algorithm. In this paper, we use Canny edge detector to generate the object boundary mask. The algorithm then uses this mask to separate the original image into the selected object and image without the object. The selected object is decomposed into several TFPC bit planes based on the specified parameter p. The parameter p which has infinite number of possible choices can act as one of the security keys for the ObjectEcrypt algorithm. The order of TFPC bit planes is shuffled by an existing or new shuffling algorithm. The encrypted object can be obtained by combining all the shuffled bit planes and scaling down all pixel values back to the range of gray levels. Finally, the encrypted object is combined with the image without the object to acquire the resulting encrypted image.

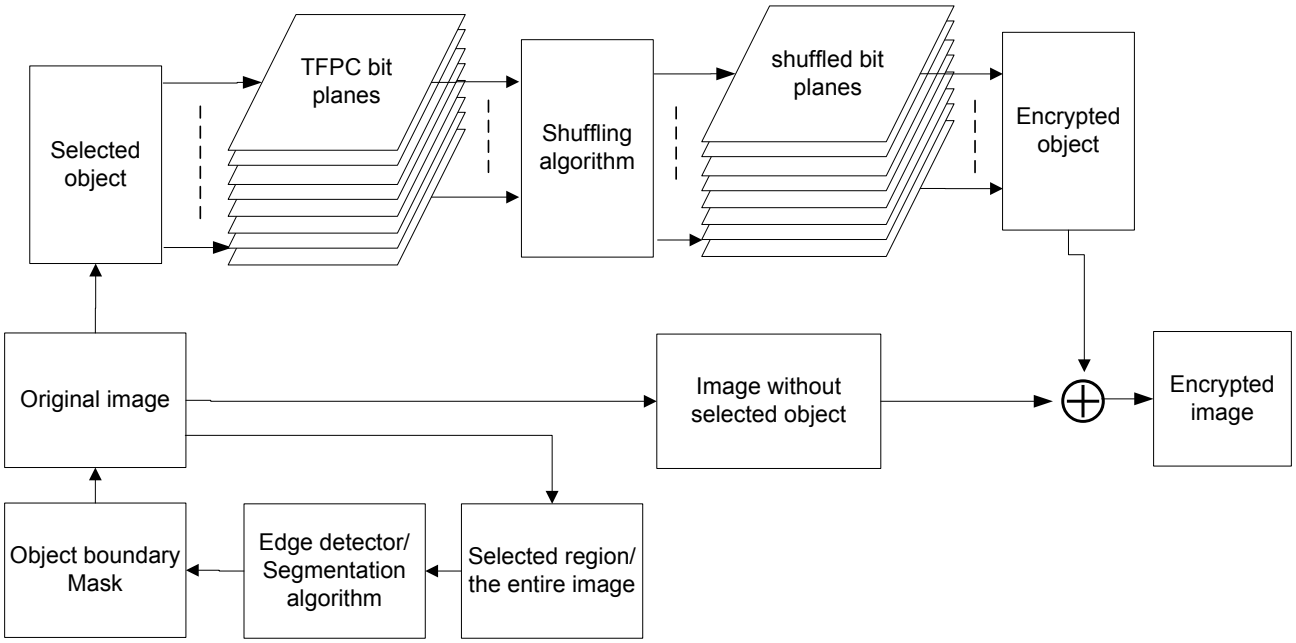

Fig. 3. Block diagram of the ObjectEcrypt algorithm.

The users also have flexibility to choose any new or existing method for the bit plane shuffling process such as inverting the order of the bit planes. In this paper, we choose right-round shift to shuffle the order of the TFPC bit planes. To make the shifted TFPC of each image pixel satisfy the constraint in equation (5), p zero bit-planes are added in front of the most significant bit plane before shifting all the TFPC bit planes. The shift algorithm is shown in Fig. 4. The shifting process will move all bit planes one bit position to their right side and the p^{th} zero bit plane will be shifted to the position of the least significant bit plane in the TFPC bit planes. If the shifting times $r > 1$, the shifting process will more all bit plane r bit positions.

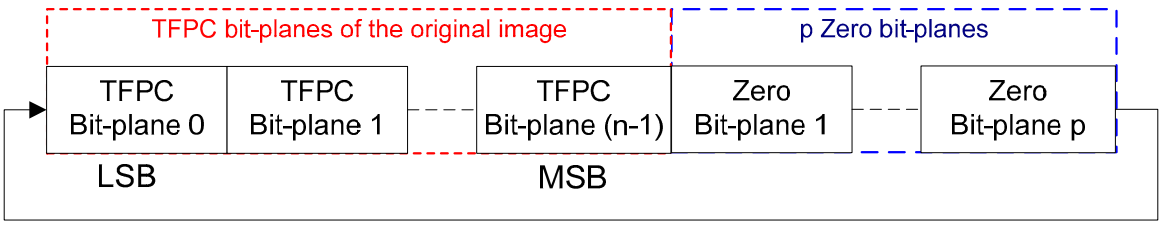

Fig. 4. Block diagram of the shifting algorithm.

The security keys of the ObjectEcrypt algorithm consist of the parameter p in the TFPC, the security keys in the shuffling algorithm, and the information of the encrypted object including the boundary mask and original data range before scaling down. In this paper, the number of times to shift, parameter r, is the security key of the shifting algorithm. The shifting times r should be less than the sum of the parameter p and the number of TFPC bit planes, i.e., $r < p + n$. These are required to be provided to the authorized users for their decryption process.

To reconstruct the original object/image in the decryption process, the encrypted object is extracted from the encrypted images using the object boundary mask. It is first scaled up to the original data range and decomposed to TFPC bit planes. The order of these TFPC bit planes is reverted back to its original order by using left-round shift. The reconstructed image can be obtained by converting the TFPC bit planes back to gray levels.

The 3D images such as color images and 3D medical images contain three 2D data matrices called 2D components. The object in 3D images can be encrypted by applying the ObjectEncrypt algorithm to its 2D components one by one.

4. SIMULATION RESULTS

The selected object can be defined as either an entire image or an object in a specific region of the image. In this section, we provide some experimental examples for these two cases to show the performance of the ObjectEcrypt algorithm for encrypting the selected objects in 2D and 3D images.

4.1 Object encryption in 2D images

Fig. 5 shows an example of grayscale image encryption using the ObjectEcrypt algorithm with security key, $p = 2, r = 5$. Fig. 6 provides a medical image encryption example, $p = 2, r = 4$. The objects in these two examples are defined as the entire images. In both examples, the original images are fully encrypted by the ObjectEcrypt algorithm. The encrypted images shown in Fig. 5(b) and Fig. 6(b) are unrecognizable. The original images are also completely reconstructed without any distortion. This can be verified by the reconstructed images in Fig. 5(c) and Fig. 6(c) which visually look the same as their original images. Their corresponding histograms in Fig. 5(f) and Fig. 6(f) also witness this perfect reconstruction.

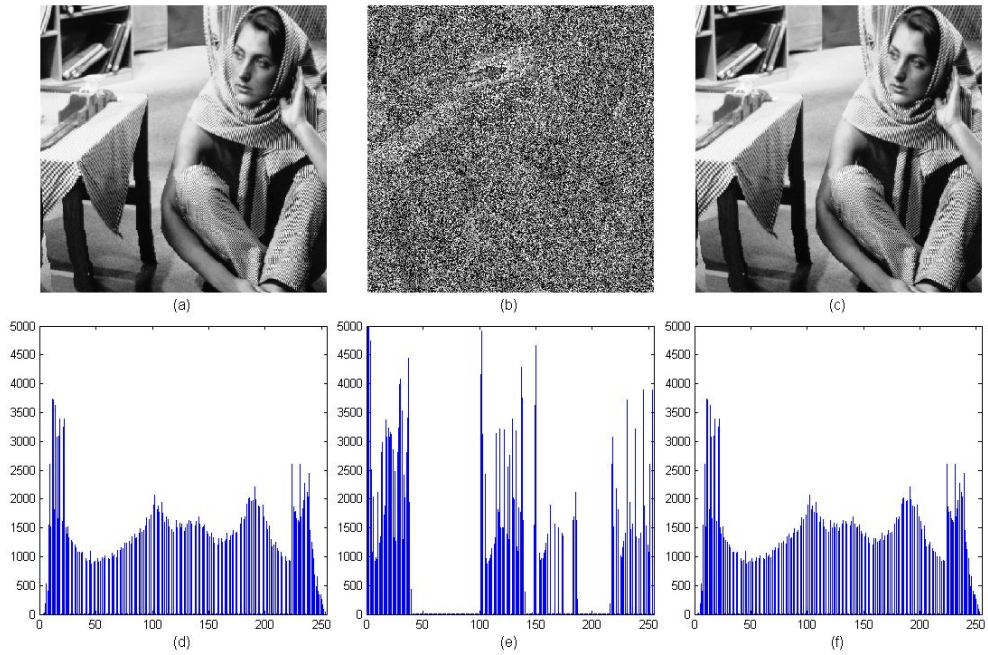

Fig. 5. Grayscale image encryption, $p = 2, r = 5$. (a) Original grayscale image; (b) Encrypted image; (c) Reconstructed image; (d) Histogram of the original image; (e) Histogram of the encrypted image; (f) Histogram of the encrypted image.

To achieve the goal of privacy protection in real-time application, it is not necessary to encrypt the entire image/video. Instead, selectively encrypting some important objects/regions in the image/video is an effective scheme. These important objects/regions usually contain private information such as human faces, fingerprints or patient's medical records along with text based personal information. Fig. 7 gives an example of the ObjectEcrypt algorithm for selected object encryption. Only the selected object has been fully protected. As a result, the computation cost of the encryption process will be significant decreased. This makes that the ObjectEcrypt algorithm is well suitable for privacy protection in real-time applications. The example in Fig. 7 shows that the selected object and the original image are also fully recovered. All these above examples demonstrate that the ObjectEcrypt algorithm is a lossless encryption method.

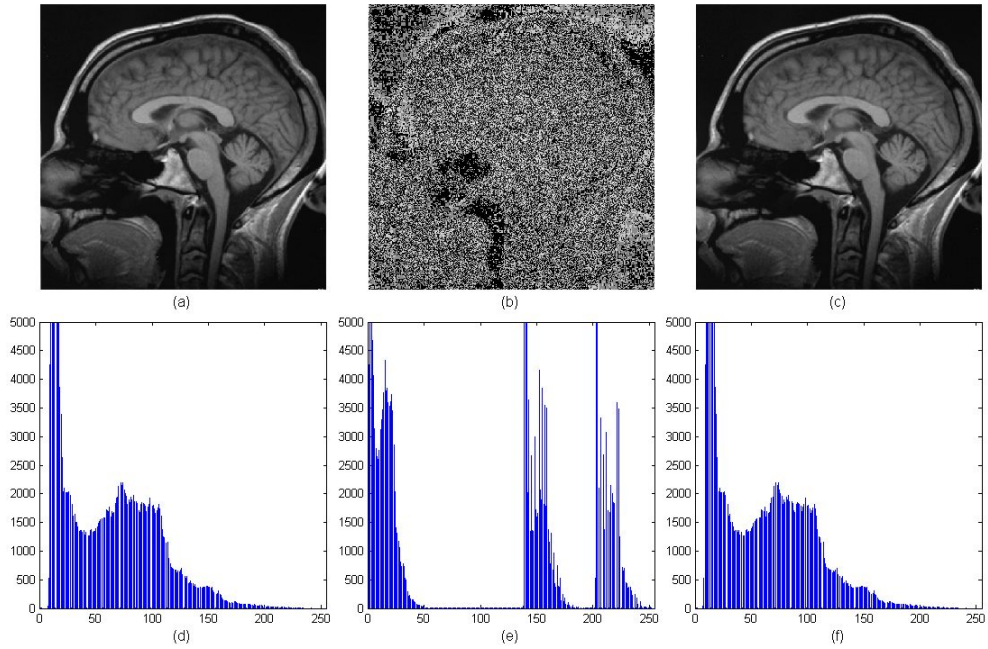

Fig. 6. Medical image encryption, $p = 2, r = 4$. (a) Original medical image; (b) Encrypted image; (c) Reconstructed image; (d) Histogram of the original image; (e) Histogram of the encrypted image; (f) Histogram of the encrypted image.

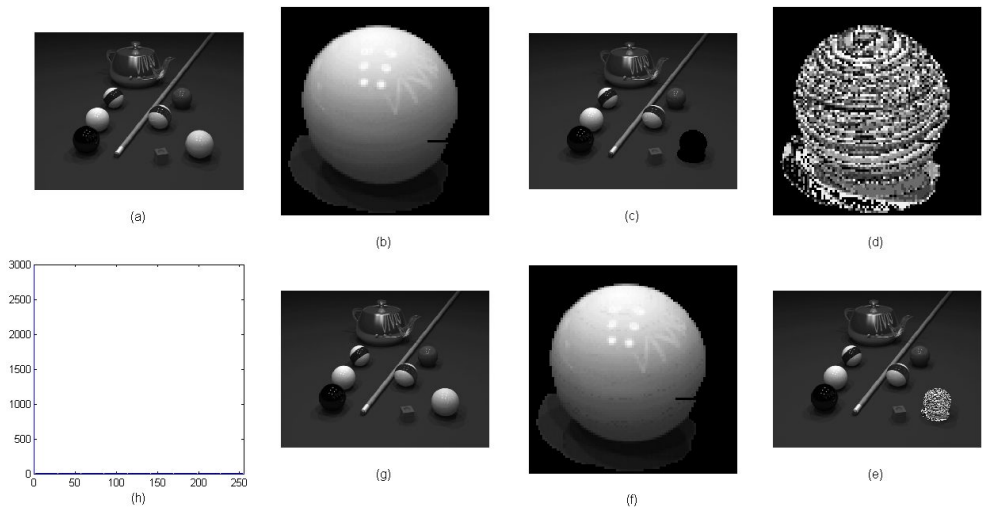

Fig. 7. Selected object encryption in a grayscale image, $p = 1, r = 5$. (a) Original grayscale image; (b) Selected object; (c) Image without the selected object; (d) Encrypted image of the selected object; (e) Encrypted image; (f) Reconstructed object; (g) Reconstructed image; (h) Histogram of the difference between (a) and (g).

4.2 Object encryption in 3D images

The selected 3D objects can be encrypted by applying the ObjectEcrypt algorithm to their corresponding 2D components one by one. Fig. 8 provides an example to encrypt a color image defined as an object. Fig. 9 shows the performance of the ObjectEcrypt algorithm for the selected 3D object encryption. The results show that the ObjectEcrypt algorithm can fully protect the private information and the original image/object can be also completely recovered.

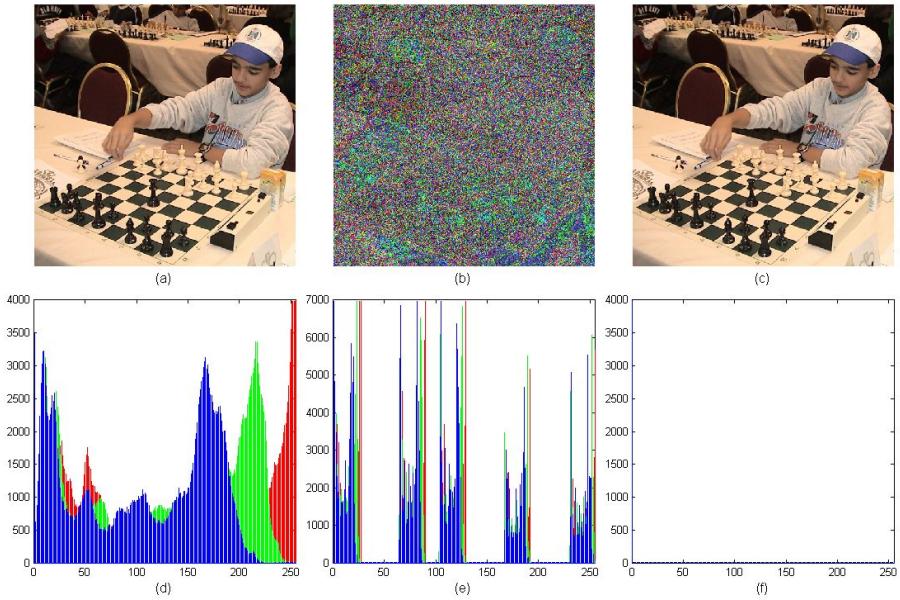

Fig. 8. Color image encryption, $p=1, r=4$. (a) Original color image; (b) Encrypted image; (c) Reconstructed image; (d) Histogram of the original image; (e) Histogram of the encrypted image; (f) Histogram of the difference between (a) and (c).

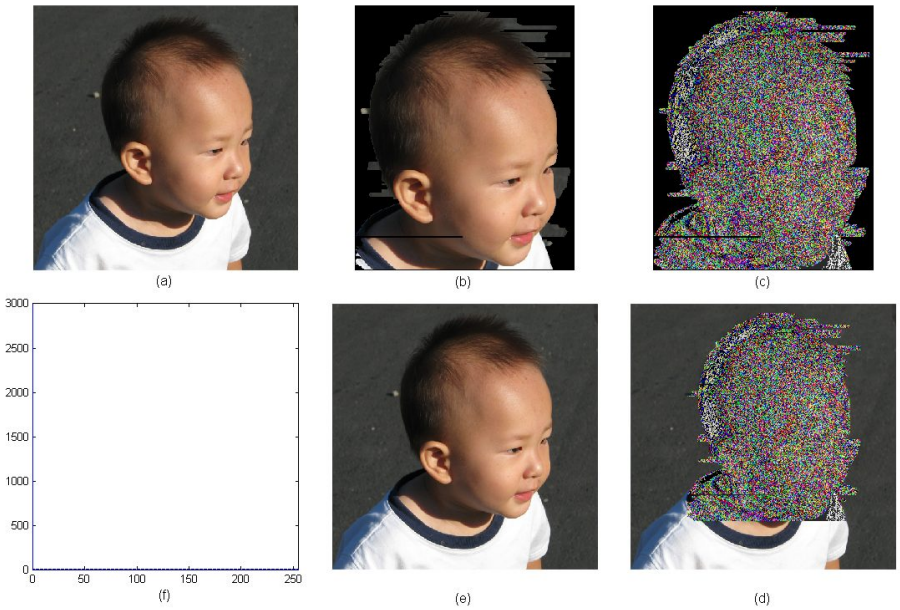

Fig. 9. Selected object encryption in a color image, $p=3, r=5$. (a) Original grayscale image; (b) Selected object; (c) Encrypted image of the selected object; (d) Encrypted image; (e) Reconstructed image; (f) Histogram of the difference between (a) and (f).

5. SECURITY ANALYSIS

In this section, we discuss the security issue of the ObjectEcrypt algorithm such as security key space and plaintext attacks.

5.1 Security key space

The security key of the ObjectEcrypt algorithm is the combination of the parameter p of TFPC, the shifting times r of the shifting algorithm in this paper, and information of the encrypted object. The parameter p of TFPC has infinite number of possible choices. The number of the TFPC bit planes changes with different p values. Based on discussion in section 3, the shifting times r is less than the sum of the parameter p and the number of TFPC bit planes ($r < p + n$). Thus, the parameter r also has limitless number of possible choice. Furthermore, the selected object changes with its regions and different methods to generate its boundary mask. All these ensure the possible number of combination of the security keys for the ObjectEcrypt algorithm to be infinite. As a result, the algorithm has unlimited security key space. It can resist the brute force attack.

5.2 Plaintext attacks

The users have flexibility to define the object to be encrypted as an entire image, any specific part/region of an image, or an object in an image or in a selected region in an image. They are also allowed to select any desired edge detection method or segmentation algorithm to obtain the object boundary mask. Therefore, the selected object is unpredictable and user-dependent. To encrypt the selected object, the ObjectEcrypt algorithm changes all pixel data in the select object by shuffling the order of its TFPC bit planes. The data of the encrypted object is not useful for the purpose of plaintext attacks. As a result, the selected object is protected with a high level of security. The ObjectEcrypt algorithm can withstand the plaintext attacks.

6. CONCLUSION

We have introduced a new recursive sequence called the truncated P-Fibonacci sequence. We also presented its corresponding binary code, the truncated Fibonacci p-code (TFPC), and the TFPC based decomposition method. They are parameter-dependent.

Based on these, we have introduced a new lossless encryption algorithm to encrypt the selected object. The object to be encrypted is either a full image, a part of an image, or a selected object in an image or in a specific region in an image. Any new or existing method of edge detection and image segmentation can be used to generate the object boundary mask in the ObjectEncrypt algorithm. The users have flexibility to apply any new or existing method for shuffling process.

Experimental results and security analysis have demonstrated that the ObjectEcrypt algorithm can protect the selected object with a high security level and withstand the brute force attack and the plaintext attacks. It can be used for privacy protection in the real-time applications such as homeland security, medical imaging, wireless network, and mobile phone services

REFERENCES

[1] P. Carrillo, H. Kalva, and S. Magliveras, "Compression independent object encryption for ensuring privacy in video surveillance," in *Multimedia and Expo, 2008 IEEE International Conference on*, 2008, pp. 273-276.

[2] K. Usman, H. Juzoji, I. Nakajima, S. Soegidjoko, M. Ramdhani, T. Hori, and S. Igi, "Medical Image Encryption Based on Pixel Arrangement and Random Permutation for Transmission Security," in *e-Health Networking, Application and Services, 2007 9th International Conference on*, 2007, pp. 244-247.

[3] R. Norcen, M. Podesser, A. Pommer, H. P. Schmidt, and A. Uhl, "Confidential storage and transmission of medical image data," *Computers in Biology and Medicine,* vol. 33, no. 3, pp. 277-292, 2003.

[4] Yang Ou, Chul Sur, and Kyung Rhee, "Region-Based Selective Encryption for Medical Imaging," in *Frontiers in Algorithmics*, 2007, pp. 62-73.

[5] F. Dufaux and T. Ebrahimi, "Scrambling for Privacy Protection in Video Surveillance Systems," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, no. 8, pp. 1168-1174, 2008.

[6] Muhammad Khurram Khan and Jiashu Zhang, "An Intelligent Fingerprint-Biometric Image Scrambling Scheme," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. vol. 4682/2007: Springer Berlin / Heidelberg, 2007, pp. 1141-1151.

[7] Park Sangho and M. M. Trivedi, "A track-based human movement analysis and privacy protection system adaptive to environmental contexts," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, 2005, pp. 171-176.

[8] Elaine M. Newton, Latanya Sweeney, and Bradley Malin, "Preserving privacy by de-identifying face images," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 17, no. 2, pp. 232-243, 2005.

[9] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images," *Image Processing, IEEE Transactions on,* vol. 16, no. 5, pp. 1395-1411, 2007.

[10] Hui-Cheng Hsu, Kun-Bin Lee, N. Y. C. Chang, and Tian-Sheuan Chang, "Architecture Design of Shape-Adaptive Discrete Cosine Transform and Its Inverse for MPEG-4 Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, no. 3, pp. 375-386, 2008.

[11] David Z. Gevorkian, Karen O. Egiazarian, Sos S. Agaian, Jaakko T. Astola, and Olli Vainio, "Parallel algorithms and VLSI architectures for stack filtering using Fibonacci p-codes," *Signal Processing, IEEE Transactions on,* vol. 43, no. 1, pp. 286-295, 1995.