Reversible Multiple Image Secret Sharing using Discrete Haar Wavelet Transform

Ashwaq T. Hashim[1], Suhad A. Ali[2]
[1]Department Control and Systems Eng., University of Technology, Iraq
[2]Department of Computer science, Science College for women, Babylon University, Iraq

Article Info

Article history:

Received Feb 2, 2018
Revised Jun 30, 2018
Accepted Jul 11, 2018

Keyword:

Block Cipher
Haar DWT
Linear System
Multiple Secret Sharing
Pseudo Random Generator
Secret Sharing

ABSTRACT

Multiple Secret Image Sharing scheme is a protected approach to transmit more than one secret image over a communication channel. Conventionally, only single secret image is shared over a channel at a time. But as technology grew up, there is a need to share more than one secret image. A fast (r, n) multiple secret image sharing scheme based on discrete haar wavelet transform has been proposed to encrypt m secret images into n noisy images that are stored over different servers. To recover m secret images r noise images are required. Haar Discrete Wavelet Transform (DWT) is employed as reduction process of each secret image to its quarter size (i.e., LL subband). The LL subbands for all secrets have been combined in one secret that will be split later into r subblocks randomly using proposed high pseudo random generator. Finally, a developed (r, n) threshold multiple image secret sharing based one linear system has been used to generate unrelated shares. The experimental results showed that the generated shares are more secure and unrelated. The size reductions of generated shares were 1:4r of the size of each of original image. Also, the randomness test shows a good degree of randomness and security.

Corresponding Author:

Suhad A. Ahmed,
Department of Computer, Science College for women,
Babylon University, Babylon, Iraq.
Email: wsci.suhad.ahmed@uobabylon.edu.iq

1. INTRODUCTION

Due to its importance, image security fields grows faster and widely, leading to the inventing of image steganography, image protection, image watermarking and the area of secret image sharing. Images need to be protected from being revealed, from being copied or from losing its content if it was containing valuable contents. By mentioning security, Cryptography is the main player in the whole field and it plays an important role in the infrastructure of the modern computing. There is no exaggeration to say that no application, even the real world one, requires no keys (such as passwords) in any part of it for serving the purposes of confidentiality, authentication, and nonrepudiation [1].

The strength of any cryptographic applications relies mainly on its key secrecy and key strength, because if the key got lost or falls in the wrong hands, this can lead to catastrophic consequences and nothing is more disastrous than losing a confidential secret information, consider a case where a confidential document that can lead to the defeat in a battle like war information or forces location. Thus, to tackle such problem many cryptosystem designers suggest such a solution:

Consider a secret S (a phrase, a password or a key) is divided into n>1 parts (called secret shares) and it satisfies these conditions:

1. The secret S can be easily restored from r shares where ($r \leq n$) shares.
2. It is impossible to restore the secret S from less than r shares.
3. Share size should not exceed the size of the secret S.

This scheme is called a r out of n, (r, n) threshold cryptography scheme or simply a secret sharing system. Secret sharing system provides a higher level of protection for the key from being lost or falls in the wrong hands by providing secure backup copies to the secret key distributed over several servers or locations. Moreover, secret sharing can be considered as a mechanism to transfer secret information using public communication channels [2]. The secret sharing scheme is an advanced cryptography branch that plays crucial role in defense passively, and it would be used to protect valuable or classified information and documents against dangers like robbery and illegal accesses [2]. Multiple Secret Image Sharing or simply (MSIS) is a secret image sharing scheme that protect more than one secret images at a time. Conventionally, transmission of a single secret image is possible over a channel at a time. But as long as technology grows, there is an enormous need to share more than one secret image at a time [3].

2. RELATED WORKS

In 1979, Shamir [4] introduced the concepts and theory of secret sharing. Shamir’s secret sharing scheme was based on a n-degree polynomial, and shares were the points on that polynomial. In 2002, Thien and Lin [5] proposed an extension of Shamir’s scheme where the secret image is shared by n shares, and any r (r <n) shares can be used to reconstruct the secret. This scheme is started with a permutation technique to shuffle the image’s pixels and de-correlate it, then shares were made by processing the image pixels or patterns in the spatial domain, each participant receiveing his own share as a shadow image looks like a random noise image holding partial information of the secret. Share size is just 1/r of the secret image. In 2007, Chin and Ching [6], introduced a way of share images based on the reversible integer-to-integer (ITI) wavelet transform.

This method processes the transform coefficients in each subband, and each of the resulting combination coefficients were divided into n shares. It was allowing the recovery of the complete secret image using any r or more shares (r ≤ n). This method has larger shadow images without coding than those belong to the methods that applying coding as a preprocessing for the inputting to the sharing phase. Also, in this method, the data is encoded either by Huffman coding or by arithmetic coding before the data is sent to the sharing phase. Also, in this method the Huffman coding or arithmetic coding have been used to encode data before it is sent to the sharing phase. In 2007, Jun et al [7], proposed a scalable secure approach to share and hide secret image. The secret image given first is divided into numerous nonoverlapping blocks, and each block is then transformed into one-level discrete wavelet transform. Then, the wavelet coefficients have been quantized into 256 gray levels. After that the gray value information of the quantized image was rearranged by using a bit-plane scanning method. Finally, the image data is rearranged into n shadows by using multiple thresholds. Finally, each shadow image of R, G, and B channels is hidden in the cover image.

The results of the test indicated that the increasing of the number of shadows can lead to recovery of the secret image with better quality. In 2011, Yang et al [8], suggested a fast secret image sharing method based on Haar wavelet transform and Shamir’s method. Firstly, they used discrete Haar wavelet transform to reduce the secret image to its quarter size (i.e., 1-level LL subband). Then, the modified Shamir’s algorithm has been applied to only this LL subband for generating the shadow images. In 2011, T. Hoang and et al [9], proposed a (2, n) gray image secret sharing scheme. The proposed scheme is basis on three existing approaches: block truncation coding (BTC), vector quantization (VQ) and discrete wavelet transform (DWT). In this scheme, the set of generated shares have been much smaller than the original image. Any gray image can be reconstructed by combining at least two shadows.

The quality of the reconstructed image ranged from 29.5 bears to 36.5 dB. In 2012, Sagar et al. [10] proposed a new method to perform color visual cryptography based on wavelet transform. Wavelet transform has been utilized to obtain a gray image from a color image where the intensity image (Y) formed from the YCbCr color transform. Then, Error-Diffusion Filter has been applied on the obtained grey image. After that the visual cryptography system (VCS) model was applied on the generated halftone image. In 2014, Ashwaq and Loay [2] introduce a security secret color image sharing based on transform coding, using wavelet or cosine transformation to produce secure secret shares by first compress the image using one of the transform coding techniques mentioned earlier. The compressed stream is then subject to a data diffuser followed by a random generator to shuffle the image into shares; a secret shares generator system is then applied on these shares to produce the secret shares.

3. THE TINY ENCRYPTION ALGORITHM

The Tiny Encryption Algorithm (TEA) is a block cipher known as a simple of description and easy to implement. This cipher was firstly introduced by (Wheeler and Needham 1994). TEA works on 64-bit plaintext at a time and using 128-bit key. It is a Feistel network of 64 rounds, typically applied on pairs termed rounds. The key schedule was extremely simple by mixing all of the key material in exactly the same way for each round. Various multiples of a magic constant have been utilized to prevent simple attacks on basis of the symmetry of the rounds. The magic constant, 2654435769 or 9E3779B916 has been chosen to be 232/ϕ, where ϕ was the golden ratio [12].

Figure 1 shows the structure of TEA algorithm. TEA is a Feistel cipher that uses different (orthogonal) algebraic groups - XOR, ADD and SHIFT in this instance. This is a truly ingenious way of saving Shannon's twin properties of diffusion and confusion which are important for a secure block cipher, without needing the explicitly of P-boxes and S-boxes respectively. It seems highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause about 32 bit differences in the cipher text) after only six cycles [13].

4. THE PROPOSED SYSTEM

An approach of multiple image secret sharing for secret gray images for set of participants has been proposed. In this approach, each participant can share a gray secret image with the other of participants in a way that of all them can reconstruct your secret gray image if only k out of n shares have been collected. The proposed approach is basis on: (i) Applying Haar DWT to reduce each input image to its quarter size, (ii) Using proposed pseudo random generator to distribute the combined compressed streams into r subblocks randomly and (iii) Performing the developed multiple image secret sharing based on linear system to generate unrelated shares. Algorithm (1) shows the steps of proposed system.

ALGORITHM 1: MULTIPLE SECRET IMAGE SHARING

```
Input: $I_1, I_2, ...,I_m$// m gray images of equal size.
r // Threshold value
n // Number of generated shares
Output: Shares // n shares
Step1: Applying haar DWT on input images secret images$I_1, I_2, ...,I_m$. The subbands$LL_1, LL_2, ...,LL_m$ of the transform images have been used to generate shares.
Step2: Combined the $LL_1, LL_2, ..., LL_m$ subbands in one total image $T$.
Step3: Separate the total image $T$ into $r$ sub blocks randomly using proposed pseudo random generator system
Step 4: Generate $n$ shares $Sh_1, Sh_2, ..., Sh_n$ using Linear System
```

4.1. Proposed Pseudo Random Generator

Digital cryptography relies greatly on randomness in providing the security requirements imposed by various information systems. Just as different requirements call for specific cryptographic techniques, randomness takes upon a variety of roles in order to ensure the proper strength of these cryptographic primitives. Block ciphers are the most popular cryptographic primitives; due to the standardization of DES followed by AES, but also due to the fact that block ciphers constitute some of the fundamental building blocks for pseudorandom number generators, stream ciphers, hash functions and message authentication codes.

A pseudo random generator has been proposed.thw length of random sequence that has been generated by proposed system is 192 bits. This step is to decorrelate the output result from compression stage; it has been distributed into *r*subblocks. By using a proposed pseudo random generator based on a secret key, it will be generated a random sequence of numbers each has a length equal to the length of combined secret data for all images after compression and the values of generated sequence are ranged to be 1..*r*, then the secret information will be permuted randomly into *n*subblocks according to generated sequence. Algorithm (2) illustrates the detail steps of the proposed pseudo random generator.

Figure 1 depicts the general structure of the pseudo random generator where the keystream generator (KSG) is a pseudo random number generator based on block cipher and the keys (i.e., K_1, K'_1) is the seed of the pseudo random generator.

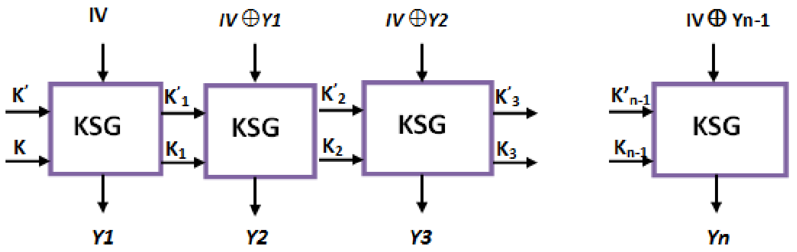


Figure 1. The proposed block cipher based pseudorandom generator

The KSG is a consecutive of two proposed block ciphers. Each of them is a cascaded design of TEA block cipher, denoted by Cascaded TEA1 and Cascaded TEA2. The Cipher Block Chaining encryption mode has been used in proposed block cipher where the input to the first Cascaded TEA is a public IV, and the inputs to each of them is one of two master keys, denoted $k_i$and k'_i respectively, these keys is considered as a seed for the pseudorandom generator. Figure 2 shows the block diagram of KSG**.**

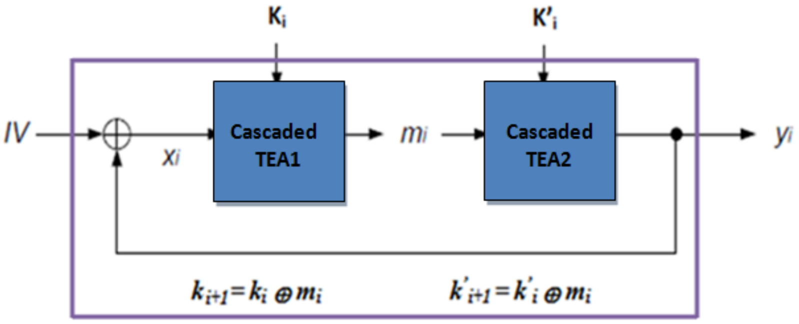
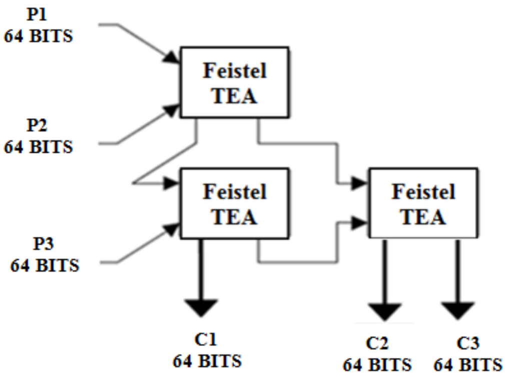


Figure 2. The proposed keystream generator KSG

As notice the x_i is the input to the first Cascaded TEA, and the m_i is an intermediate. Then the output of the KSG is y_i. The detail of Cascaded TEA has been showed in Figures 3 and 4.



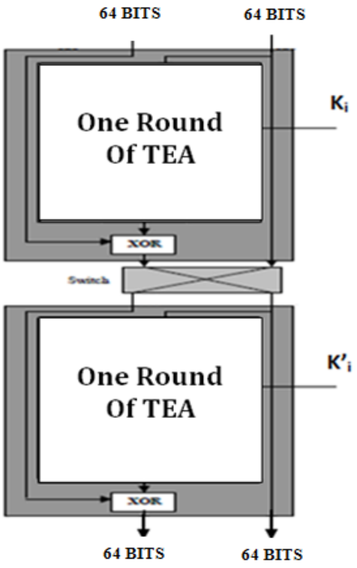


Figure 3. The proposed of Cascaded TEA

Figure 4. The proposed of Feistel TEA

ALGORITHM 3: PROPOSED PSEUDO RANDOM GENERATOR

```
Input:   T  // Combined the $LL_1, LL_2, ..., LL_m$subbands
         n  // Length of  T
         r// Threshold value
               K, K' // Master keys each of which 64 bits
Output: $S_{Bk}$   // Generated $r$subblocks
Step1:   The new input for the first Cascaded TEA is:

         $x_i$= IV
                                                              (3)
Step2:   The $m_i$ is computed as intermediate value:

         $m_i$= Cascaded $TEA_{ki}$ ($x_i$)                      (4)

Step3:   The KSG output $y_i$ is computed as:

         $y_i$ = Cascaded $TEA_{k'i}$($m_i$)                     (5)

Step4:   The new input for the next Cascaded TEA is:

         $x_{i+1}$= IV $\oplus y_i$.                             (6)

Step5:   In the KSG design, the internal state at each step has been used to update running keys such as
         following

         $k_{i+1} = k_i \oplus m_i$                               (7)

         and

         $k'_{i+1} = k'_i \oplus m_i$                              (8)

         The $k$ and $k'$ represented the master keys and the $k_i$ and $k'_i$ the running keys.
Step6:   Repeat step2 to step5 until generate $y_i$where $i=1..n$
Step7:   For each $y_i$ components (i.e.,$C_1$, $C_2$, and $C_3$) where $i=1..n$, Do the following:
         Step7-1:Let  $K = C_1, I=n-1$
         Step7-2: While 1>1
         $K \leftarrow (C_2 \times K + C_3)$ modular  $I$
         Swap b[I], b[K]
                        End loop I
Step8:   For I=1 → n
              $X \leftarrow b[I]$
              IF(X <> 0)
                 $X \leftarrow X \times n / r$
              ENDIF
         no←count[X]+1
         w←X+ no
         $S_{Bk}[w]=S[I]$
         end  loop I
```

4.2. Multiple Secret Image Sharing Based on Linear System

The input $I_1, I_2, ..., I_m$ images have been reduced (i.e., to its quarter size) and shuffled randomly into $r$subblocks. In the multiple secret images sharing, the input content is composed by a set of equal sized blocks (i.e., $S_{Bk1}, S_{Bk2}, ..., S_{Bkr}$). Each of the blocks can be any type of content. For a block of data $\{Q_j | j=1..r\}$ (i.e., each $Q_j$is a byte from j^{th} block), the i^{th} share is calculated by using the linear equations adapted from [2] and shown in (9):

$$Sh_i = c_{ij}Q_j + c_{ij}Q_j + ... + c_{ij}Q_j \ mod \ 255, \ i=1,...,n, j=1,..,r \quad (9)$$

Where, Sh_i is the i^{th} share which is generated for the subblock*Q()*, c_{ij} is the j^{th} coefficient belong to the linear equation denoted the i^{th} share.

5. EXPERIMENTAL RESULTS AND DISCUSSION

Different tests were implemented to evaluate the proposed system algorithm performance. Six grays of size 512×512 images (“Lena”, “Jet-plane”, “Splash”, “Peppers” and “Sailboat” and “House”) have been used. The test images have different visual properties, like a natural scene, having lot of edges and having high correlated sections. These six test images are shown in Figure 5.

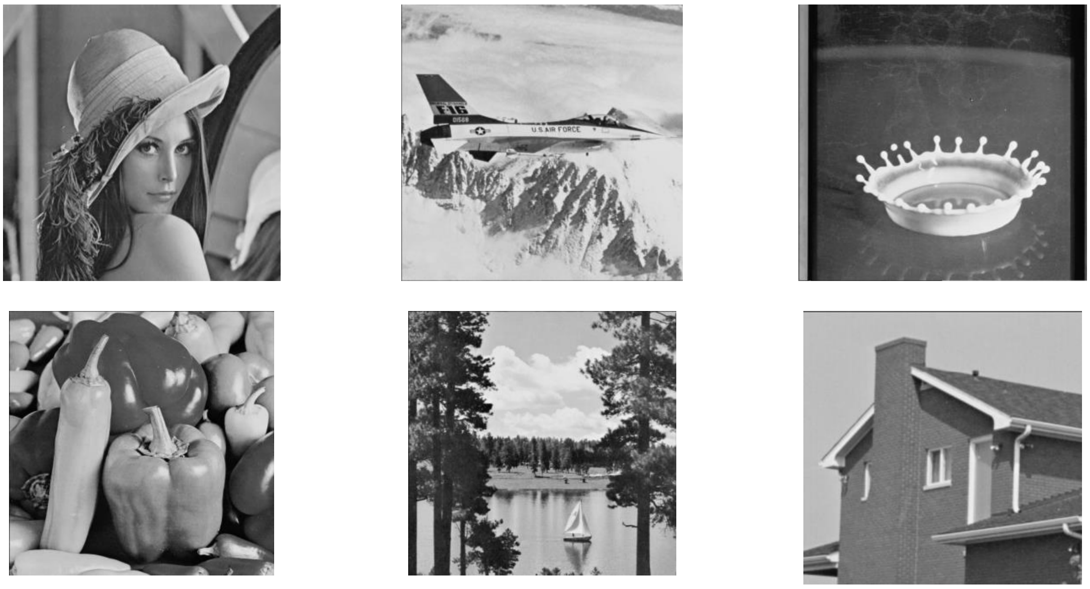


Figure 5. Test images

Table 1 lists the PSNR after applying haar DWT; the only distortion of the proposed multiple secret image sharing was due to the use of this step, no further loss was caused by the rest operations or functions of the proposed system.

Table 1. The CR, Number of bytes and PSNR of each image when applying proposed compression scheme

Image	PSNR
Lena	34.52
Jet_plane	30.01
Splash	33.35
Peppers	33.28
Sailboat	30.90
House	41.21

The results that have been listed in Table 1 showed that the size of each has been reduced, 1/4 of the size of the original secret image and the qualities of all images were at acceptable level (i.e., greater than30 dB). So that the total size for all input images have been reduced to 1/4r after applying a (r, n) threshold multiple secret image sharing.

The proposed Pseudo-random number generator depends on two master keys and an initial vector to generate and many random numbers as needed. Many statistical tests have been applied to test the randomness of a sequence, in general many sequences that considered random may be easy to predict. So it's important to test the generators to prove its efficiency. Table 2 shows first three samples generated using the specified initial vector and keys. Table 3 shows the differences between each two consecutive generated sequences. Table 4 shows the results of randomness test of 18 random sequences that are generated in table 2.

Table 2. Three consecutive random sequences that is generated using different initial and master keys

Sample no	Initial vector	Master Key 1	Master Key 2	First random sequence	Second random sequence	Third random sequence
1	0	0	0	0111110000000011	1011011000111011	0011100100000111
				0011001000100010	1010110100100000	0100111011111011
				0101111111101111	1010011010001100	0101010000101000
				0011000110100110	0111010111010111	1011101010110101
				0101110110110110	0101001110001000	1111001000001100
				1101000001101111	1110000101110010	0001110110001011
				0100000010001101	0000000010000000	1101011011000011
				0001001101101101	0010000111110110	0010101000011110
				1001100110000011	1000100101000000	1011000101110001
				0000001001010001	0010111111100001	1101000101000110
				1011110011101010	1101000001010010	1100001100001010
				1000001001010010	0000100000110111	1010100101001111
2	0	1	0	0010010110111110	0100001100100101	0001111111001001
				1100010001001101	1000101100011111	0101111111101110
				0010010011001100	1111001010110100	1101001001001001
				1010111001001011	0001010011101101	1100100111100000
				1001111000110111	1111101010000111	0011110110101011
				0111100110111001	1011110111101101	1001000111001110
				1101101111101000	1111000100000011	0111011100011000
				1101001100101111	1011011111110101	0011110111000101
				0101111110101110	0001000010101000	0111010111010000
				0111110100001001	1011101011100001	0100000100100001
				1110010011011101	1101110000000001	1101110011001100
				0101001100000110	0101001001010111	0101100011110001
3	0	0	1	1011110110001111	1010001011010100	1011101000001000
				1001110110010000	1111010001011100	0101100111101001
				0001011101010010	0010110010011100	0101010001110111
				1100010100101110	1100010111010000	1101101000110011
				0000010110110001	1110101000110010	1111010000101000
				1101010000010010	0110010111011101	1011101001000111
				1100111001101101	0011010000001001	1010100000101111
				0101011100100101	1111100100001111	0001000101000001
				0100000011010110	0100111100011110	1111111100110000
				1000101100000001	1001100100100010	0100011001101011
				0110100111100010	1101001111011100	0010001100101100
				1010110001110000	1110100110111011	1001000110001111
4	1	0	0	1000100110110101	0111100000100110	0011111001011001
				1110001010110110	0110010011111010	0011100111101110
				1001100101000000	0101100100110010	0110011101111111
				1011001101001010	1110100011101101	1101011000001100
				1001111000110111	1111111001100001	1110010000110110
				0111100110111001	0101100010000001	0000011100111101
				1101101111101000	0101111000101010	1000110000111100
				1101001100101111	0101001101000101	1010000110001011
				1010101000100111	1100001111011000	1001110100100111
				0111111010011111	0110100000100010	1101001011001011
				0001101001110011	0100110110100000	0110110000011000
				1100011110001110	0101000111010000	1000100101110010
5	0	11111111111	0	1000101000110110	1000011101111001	0111111111111100
		11111111111		1110001000111110	0111010111001011	0101100000000001
		11111111111		0011000101101111	1010010010101100	0110011110100101
		11111111111		1110100001001110	1010111010011111	1011100001011100
		11111111111		1001111000110111	0001011010011100	0110000010010111
		11111111111		0111100110101001	0100000101010100	0110011100100011
		11111111111		1101101111101000	0101100000010101	1010101100100110
		11111111111		1101001000111111	1101001001011010	0110101100100001
		11111111111		0011000001110011	0111010001010001	0110010001011010
		11111111111		1000011110000110	1000010101100010	0100111101111010
		11111111111		0001101000000101	1111001111011010	0001001001101100
		1111111		0101111010111011	0001111111110011	0001111111000011

Table 2. Three consecutive random sequences that is generated using different initial and master keys

Sample no	Initial vector	Master Key 1	Master Key 2	First random sequence	Second random sequence	Third random sequence
6	0	0	11111111111	0001111001111000	0010010111100101	1001010101110101
			11111111111	1100100010001101	0000110110110110	0111100101111001
			11111111111	0010011010011001	1111000101010001	0100000110101111
			11111111111	1010111011010000	1001011010011000	0010110100110100
			11111111111	1001111000110111	0011010111010010	0001111101000011
			11111111111	0111100110111001	0100001001010010	1011011010110010
			11111111111	1101101111100100	0000111111100001	1011111011011111
			11111111111	1101001100110001	0111101011010111	0101010111100111
			11111111111	1001011011000100	0110111100111011	1101101010110110
			11111111111	0000000101001100	0111111001011100	1110101000000100
			11111111111	0101100111001011	0100010011101110	1101000110100010
			111111	1011100101010010	0010101010100110	0000100100101001

Table 3. Difference between consecutive generated sequences

Sample no	Differences between 1st and 2nd	Differences between 2nd and 3rd
1	101 (52.6%)	99 (51.56%)
2	99 (51.56%)	92 (47.92%)
3	102 (53.13%)	97 (50.52%)
4	98(51.04%	103 (53.64%)
5	98 (51.04%)	102 (53.13%)
6	105 (54.69%)	94 (48.96%))

Table 4. Results of NIST Statistical of generated random sequences

Frequency (Monobit)	Frequency Test within a Block M=16	Runs Test	Longest Run Of Ones	Cumulative Sums	Serial M=3	Approximate Entropy M=2	Linear Complexity M=8
0.0123	0.2117	0.8537	0.3339	0.3981	0.9248	0.1860	0.8654
0.0233	0.0116	0.4577	0.1299	0.0267	0.7316	0.1423	0.0811
0.0417	0.5083	0.9023	0.4124	0.0432	0.9845	0.4119	0.5523
0.0405	0.0664	1.0000	0.4124	0.1175	0.9845	0.4119	0.8231
0.1472	0.6512	0.4232	0.7208	0.2543	0.8825	0.5218	0.0996
0.1157	0.0299	0.3422	0.7208	0.1783	0.6766	0.6006	0.3241
0.1371	0.6512	0.1907	0.7208	0.0781	0.8825	0.4646	0.0445
0.2527	0.0276	0.5752	0.7208	0.1046	0.6356	0.5760	0.2883
0.2146	0.6309	0.4230	0.7208	0.1772	0.2650	0.3002	0.8877
0.2777	0.0424	0.3286	0.7208	0.3251	0.4437	0.5427	0.7619
0.4563	0.4066	0.3269	0.7891	0.1362	0.1607	0.2709	0.3242
0.9542	0.0642	0.8124	0.6928	0.3524	0.4369	0.6103	0.0222
0.4871	0.5661	0.2540	0.7632	0.3697	0.1054	0.1664	0.4987
0.3222	0.0642	0.2348	0.7208	0.2765	0.1038	0.2532	0.3245
0.7234	0.5879	0.7236	0.7891	0.3555	0.1038	0.2990	0.7856
0.8234	0.0843	0.7655	0.8770	0.3241	0.2096	0.5089	0.4759
0.9271	0.6506	0.2767	0.8770	0.9977	0.3622	0.5500	0.8921
0.2333	0.6564	0.4766	0.6487	0.3777	0.4437	0.5919	0.5437
0.3532	0.4814	0.2435	0.9145	0.2341	0.2096	0.3294	0.7321
0.2821	0.6642	0.4210	0.6487	0.2342	0.4369	0.4737	0.5911

The result listed in Table 3 shows that most differences are greater than 50% of the length of generated random sequences while Table 4 shows that all the P-values are greater than α ($\alpha = 0.01$)value. So the proposed pseudo random generator output have passed NIST statistical test suite. Figures 6 and 7 show an example of coding and decoding of proposed multiple image secret sharing phases.

Table 5 illustrates the time required to recover the secret images of size 512×512 when applying (2,5) threshold multiple secret image using linear system and the sharing coefficient values for share1 are: a_{11}=131, a_{12}=132 and for share2 the coefficient values are: a_{21}=133, a_{22}=135.

Table 5. The time required to encode and recover multiple secret images using (2,5) threshold multiple secret image using linear system

No. of Secret images (m)	Size in bytes	Time in Sec. Haar (DWT)	Encoding of Linear system	Revealing of linear system	Inverse of Haar (DWT)	Pseudo Random Generator	TOTAL Encoding Phase	TOTAL Revealing Phase
2	131072	0.04	0.014	6.72	0.020	0.026	0.08	6.766
3	196608	0.06	0.022	10.08	0.032	0.019	0.101	10.131
4	262144	0.08	0.029	13.45	0.040	0.026	0.135	13.516
5	327680	0.10	0.039	16.81	0.053	0.033	0.172	16.896

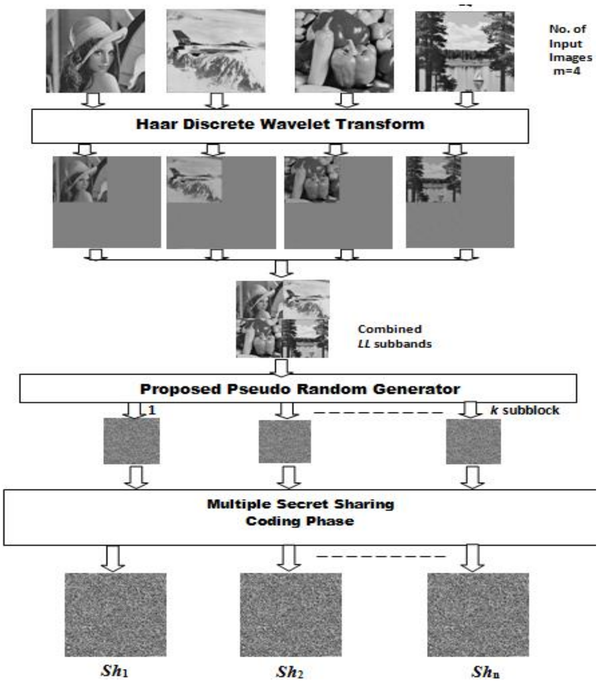

No. of Input Images m=4
Haar Discrete Wavelet Transform
Combined LL subbands
Proposed Pseudo Random Generator
1
k subblock
Multiple Secret Sharing Coding Phase
Sh_1
Sh_2
Sh_n

Figure 6. Example for encoding phase of proposed multiple secret image sharing

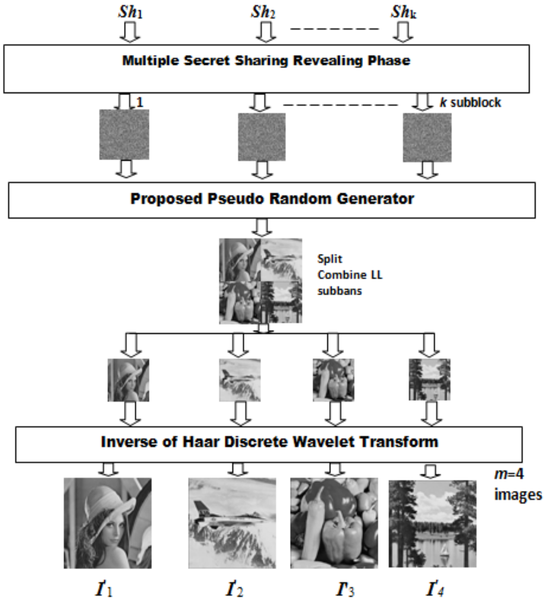

Sh_1
Sh_2
Sh_k
Multiple Secret Sharing Revealing Phase
1
k subblock
Proposed Pseudo Random Generator
Split Combine LL subbans
Inverse of Haar Discrete Wavelet Transform
m=4 images
I'_1
I'_2
I'_3
I'_4

Figure 7. Example for revealing phase of proposed multiple secret image sharing

6. CONCLUSION

In this paper, a (*r*, *n*)-threshold multiple image secret sharing based on linear system with the haar wavelet transform has been proposed. The benefit of a small computation time of reduced data has been exploited after applying haar wavelet. All secret images have been preserved of their quality (i.e., greater than 30 Db) after the reduced their sizes to quarter size. The proposed system employed pseudo random number generator to permute the transformed images data in pre-generated shares randomaly. The proposed pseudo random generator based on block cipher technique which is depended on a re-keying approach to overcome key schedule weakness in TEA algorithm and used the rounds of the TEA which is a highly random block cipher algorithm. The developed multiple secret image sharing has been applied to generate uncorrelated shares to store in different servers. If any *r* out of *n* shares have been collected, can be recovered the *m* secret images.

REFERENCES

[1] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, Oct 16, 1996.

[2] Ashwaq T. Hashim and Loay E. George; "Secret Image Sharing Based on Wavelet Transform", *International Conference on Information Technology in Signal and Image Processing (ITSIP-2013)*, Oct 18-19, 2013, Mumbai.

[3] Mohit Rajputa and Maroti Deshmukhb, "A Technique to Share Multiple Secret Images", *International Journal of Information Processing*, Vol. 10, No. 3, Pp. 35-44, 2016.

[4] A. Shamir, "How to Share a Secret", Communications of the ACM, Vol. 22, No. 11, PP. 612-613, 1979.

[5] C. Thien, J. Lin, "Secret Image Sharing", *Computers & Graphics*, Vol. 26, PP. 765-770, 2002.

[6] C. Yang, Y. Huang and J. Syue, "Reversible Secret Image Sharing Based on Shamir’s Scheme with Discrete Haar Wavelet Transform", *Electrical and Control Engineering (ICECE), International Conference*, Yichang, PP. 1250 - 1253, 2011.

[7] C. Huang and C. Li, "A Secret Image Sharing Method Using Integer Wavelet Transform", *Eurasip Journal on Advances in Signal Processing*, Vol. 2007, No. 2, PP. 1-13, 2007.

[8] J. Kong, Y. Zhang, X. Meng, Y. Zheng, Y. Lu, "A Scalable Secret Image Sharing Method Based on Discrete Wavelet Transform", *Lecture Notes in Computer Science*, Springer, Berlin, Vol. 4688, PP. 736-745, 2007.

[9] C. Huang and C. Li, "Secret Image Sharing Using Multiwavelet Transform", *Journal of Information Science and Engineering*, Vol. 27, PP. 733-748, 2011.

[10] S. Nerella, K. Gadi, R. Chaganti, "Securing Images Using Colour Visual Cryptography and Wavelets", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, No. 3, PP. 164-168, 2012.

[11] Hernández, Julio César; Isasi, Pedro; Ribagorda, Arturo. "An application of genetic algorithms to the cryptoanalysis of one round TEA". *Proceedings of the 2002 Symposium on Artificial Intelligence and its Application*, 2002.

[12] Hernández, Julio César; Sierra, José María; Ribagorda, Arturo; Ramos, Benjamín; Mex-Perera, J.C. (2001). "Distinguishing TEA from a random permutation: Reduced round versions of TEA do not have the SAC or do not generate random numbers", *Proceedings of the IMA Int. Conf. On Cryptography and Coding* 2001: 374–377.