JOINT IMAGE WATERMARKING, COMPRESSION AND ENCRYPTION BASED ON COMPRESSED SENSING AND ENTROPY CODING

Mohab Mostafa[1] and Mohamed Waleed Fakhr[2]

[1]Computer Science Department, College of Computing, Arab Academy for Science and Technology, Cairo, Egypt
[2]Computer Engineering Department, College of Engineering, Arab Academy for Science and Technology, Cairo, Egypt

ABSTRACT

Image usage over the internet becomes more and more important each day. Over 3 billion images are shared each day over the internet which raise a concern about how to protect images copyrights? Or how to utilize image sharing experience? This paper proposes a new robust image watermarking algorithm based on compressed sensing (CS) and quantization index modulation (QIM) watermark embedding. The algorithm capitalizes on the CS to compress and encrypt images jointly with Entropy Coding, Arnold Cat Map, Pseudo-random numbers and Advanced Encryption Standard (AES). Our proposed algorithm works under the JPEG standard umbrella. Watermark embedding is done in 3 different locations inside the image using QIM. Those locations differ with each 8-by-8 image block. Choosing which combination of coefficients to be used in QIM watermark embedding depends on selecting a combination from combinations table, which is generated at the same time with projection matrices using a 10-digits Pseudo-random number secret key SK_1. After quantization phase, the algorithm shuffles image blocks using Arnold's Cat Map with a 10-digits Pseudo-random number secret key SK_2, followed by a unique method for splitting every 8x8 block into two unequal parts. Part number one will act as the host for two QIM watermarks then goes through encoding phase using Run-Length Encoding (RLE) followed by Huffman Encoding, while part number two goes through sparse watermark embedding followed by a third QIM watermark embedding and compression phase using CS, then Huffman encoder is used to encode this part. The algorithm aims to combine image watermarking, compression and encryption capabilities in one algorithm while balancing how those capabilities works with each other to achieve significant improvement in terms of image watermarking, compression and encryption. 15 different images usually used in image processing benchmarking were used for testing the algorithm capabilities and experiments show that our proposed algorithm achieves robust watermarking jointly with encryption and compression under the JPEG standard framework.

KEYWORDS

Robust watermarking, Compressed sensing, JPEG standard, Arnold Cat Map, Quantization index modulation (QIM), Data hiding, Advanced Encryption Standard, Pseudo-random number

1. INTRODUCTION

With the rapid development of digital communication and technologies, digital watermarking has been widely used in various applications such as authenticity and identification of digital files, track digital products and usage control [1, 2]. Digital watermark is a signal which can be embedded into the host image and it can be visible or invisible. Many researchers focus on four main features; imperceptibility, confidentiality, robustness and watermarking capacity [3]. Another important issue is how to guarantee communication efficiency and save network bandwidth while embedding more information into the image. Most watermarking algorithms

focus on improving the main four features without taking into consideration the size of the image file while others rely on compression techniques which decrease the size of the image file but affects watermark detection. Moreover, photo-sharing has always been a popular practice on the Internet. It is very easy for a user to collect a large amount of multimedia data from different sources without knowing the copyright information of those data, not to mention that the data may be misused by the user. Therefore, securing the multimedia and the watermark must be taken into consideration
.
Recently, some researchers proposed watermarking algorithms based on compressive sensing (CS) [4, 5, 6], while others proposed compression algorithms also based on CS [7, 8, 9]. They rely on the fact that CS both compresses and randomly projects a sparse representation of an image [10]. Recently, researchers used CS for encryption and its information hiding capability has been analysed with high robustness [10, 11]. CS states that it is possible for a signal to be reconstructed with only a few samples under certain circumstances; Samples should be collected randomly, and signal must be sparse [12]. Inside JPEG algorithm, after we quantize sparse DCT coefficients, three main questions arise: Firstly, what is the best technique to embed watermark without sacrificing image quality and being robust against different attacks at the same time? Secondly, which coefficients should we compress using CS to get the smallest image size while preserving image information? Finally, how can we achieve high compression ratio while embedding a watermark usually expands image size and while a high compression ratio may cause losing watermark information?

In this paper, we show a unique combination of watermark host coefficients selection, CS projection, Entropy coding, Pseudo-random numbers and Arnold shuffling leads to a novel algorithm that can offer robust watermarking, high compression ratio and strong encryption in one algorithm that gives better performance compared to other standalone algorithms. Moreover, a sparse watermark as in [13] is used to hide information about the sign of the watermark, and as a pointer to which set of host coefficients are used for the robust QIM watermark embedding.

This paper has 3 main objectives. The first objective is to achieve a highly robust watermark against Additive White Gaussian Noise attack (AWGN). The second objective is to achieve high compression ratio as in JPEG algorithm by depending on CS/Huffman encoding after a unique splitting technique applied on DCT coefficients. The third objective is to partially encrypt images by using random CS projection as-well-as generating random projection of CS using Pseudo-random number secret key SK_1 as a first level encryption. Arnold Cat Map is then used to shuffle image blocks with Pseudo-random number secret key SK_2 as a second level encryption. Moreover, hiding DC-Values within CS coefficients in addition to using Advanced Encryption Standard (AES) with 256-bit key for encrypting Huffman tables as a third level encryption. An algorithm that achieves the above objectives is proposed and 15 different images usually used in image processing benchmarking were used for testing the algorithm capabilities.

In summary, the contributions of this work are the following:

1. The present work is designed to be the first to consider robust watermarking jointly with secure encryption and compression. This research introduces the idea of selecting certain image coefficients to be encoded by Huffman while encoding the rest of image coefficients using Compressive Sensing. This idea enhances effectiveness of both encoders.

2. This paper advances quantized Gaussian sensing matrix which leads to better compression ratio. In addition to resulting in a good performance of robustness compared with existing sensing matrices types.

The rest of this paper is structured as follows; section 2 discusses the related work. Section 3 discusses the security of Compressive Sensing under a one-time use. Section 4 has the details of the proposed algorithm while section 5 shows the experimental results and discussion. Finally, section 6 has the conclusions and future work.

2. RELATED WORK

2.1. Compressive Sensing Theory

CS was introduced in [12, 14]. CS state that it is possible for a signal to be reconstructed perfectly with only a few samples under certain circumstances. A summary about compressive sensing concept can be located in [14, 15, 16, 17]. Consider x to be the signal with dimensions N×1; $x \in R^N$ To obtain M linear measurements (non-adaptive) from x, we multiply x by matrix φ. This sampling mechanism is represented as:

$$y = \varphi x \tag{1}$$

Here, φ is called measurement matrix with dimensions M×N. y is a compressed measurement vector of dimension M ×1 where M < N as shown in Fig. 1.

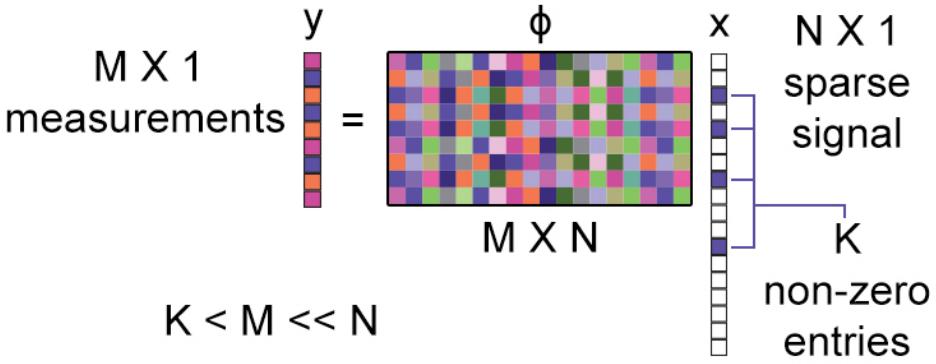

Figure 1. CS matrices structure.

2.2. CS in Watermarking

In [13] the author proposed a sparse audio watermark embedding and recovery technique using CS. The one non-zero sparse watermark vector is projected by a random matrix and added to the host signal, which is made sparse in a specific transform domain. The technique had three main advantages; firstly, the embedding is secure and distributed as opposed to specific coefficients embedding. Secondly, it is robust to additive noise as opposed to QIM based techniques, and finally, it has the capability of recovering the host signal perfectly in clean conditions. Wavelet based CS watermarking was proposed in [18], a pseudo random sequence watermark is embedded in a certain region of the image obtained using second step Haar wavelet decomposition. The technique can reconstruct image in a good quality (PSNR above 30dB) with only 25.9 % of total coefficients measurements in addition to being able to detect watermark successfully. This technique suffers from lack of security as well as visually degraded image quality when trying to minimize the number of measurements, not to mention that the watermark might be vulnerable to different attacks. In [19] a watermarking scheme is proposed for color digital image based on compressive sensing and chaos theory in DCT domain and singular value decomposition (SVD) domain. The scheme shows improvement in watermark capacity and robustness but did not take image size into consideration.

Another CS-based watermarking algorithm proposed in [20]. This algorithm adopts CS for compression and encryption, original image goes through 2-D DWT to highlight the important part and unimportant part. After that, the important LL2 coefficients are divided into blocks, marked to get a sequence as the watermark position key, then encrypted by traditional stream cipher. Another wavelet coefficients get simultaneously encrypted and compressed using CS. Finally, watermark is embedded into high frequency coefficients and scrambled to enhance security. This watermarking scheme provides robust and secure watermark, but it does not offer high compression ratio. More CS-based techniques were explored in [4, 21, 37].

The main issue with previous watermarking techniques is the image size. Usually embedding more data in the image expands its size. Trying to compress the image might damage the embedded watermark making it unrecoverable. Moreover, lack of security.

2.3. CS in Compression

Recently, several researchers adopted the concept of CS-based image compression [22, 23, 24, 25, 26, 27, 28]. As reported by CS theory, the signal must be sparse in transform domain i.e. DCT to be perfectly recovered from small number of measurements. Moreover, inside compressive sensing scheme, input signal should be a column vector. A random matrix is used for measurements reduction. But since distribution of coefficients become more flat, it leads to increasing memory storage and computational cost thus Entropy coding turns out to be less efficient. Nevertheless, [22] proposed a CS-based compression algorithm. At first, the images were splitted into non-overlapping blocks, then a sampling process using the same measurement matrix is performed per block. To reconstruct the image and reduce blocking artifacts, Wiener filtering and a hard thresholding Projected Landweber (PL) was used. Block-based compressive sensing shrinks measurement matrix dimension in addition to sparse vector length. This leads to better performance regarding computational work. The main problem with these schemes is that image data statistical structure is not entirely explored [23].

In [24] a block-based DCT sampling process for images was introduced. This technique uses a weighted Gaussian matrix for sampling. This technique relies on the fact that human eyes are sensitive to high energy (low-frequency) components than low energy (high-frequency) components in an image. To obtain the weighting coefficients, inverse JPEG quantization table entries were used. The weighted sampling matrix results in effectively quality increase of the reconstructed image. Different sampling matrix for each block made this a complex scheme. A re-weighted CS-based sampling was proposed in [25]. By adjusting sampling coefficients using calculated weights from statistical properties of the image results in better performance. Another CS-based sampling scheme was proposed in [26]. In this scheme, a random coefficients permutations in block-based DCT was adopted in addition to an energy contributions adaptive sampling matrix which uses different DCT coefficients frequency components per block. But this scheme suffers from different permutation orders for different vectors. Other CS-based compression techniques were explored in [27, 28].

Even with previous compression techniques reaching out new standards using CS technique, it was not enough to compete with JPEG compression which is adopted by 73.9 % of all websites making it the most popular lossy image compression standard [29].

2.4. CS in Encryption

Computational secrecy of CS-based encryption depends on the hardness of computation to find the precise measurement matrix from many candidates. Recovering signal using different measurement matrix will results in an incorrect signal, this was proven in [10]. Thus, CS

measurements encryption is possible. In [7], a new CS-based hybrid image compression-encryption algorithm using random exchanging of pixels was demonstrated. Where encryption and compression phases are done simultaneously. Image is broken down into four blocks. After that a random exchanging of pixels is offered by the algorithm to scramble blocks. Similar techniques were explored in [30].

Most previous encryption techniques depend on the compression-encryption ability of CS which might not be enough to secure the signal nor to prove the owner copyrights. In addition to deficiency of watermarking and compression.

3. SECURITY OF COMPRESSIVE SENSING

In one-time key scenario, Compressive Sensing encryption resistance against chosen-plaintext key recovery were discussed in [31]. The secrecy of CS depends on the randomness of sensing matrix. While all the values of compressed measurement vector are random, the adversary needs to exhaustively search for all possible combinations in order to decrypt the signal. Instead of exchanging sensing matrix over a secure channel, parties may rely on exchanging a secret key which is used later as a seed to generate sensing matrix. In that case, the secrecy of system reduces to the length of shared secret key.

In addition to image blocks being shuffied, the proposed algorithm uses the same sensing matrix for all blocks. However, it uses a brute force technique to heavily compress each block by finding minimum number of sensing matrix length needed to perfectly recover each block. This results in measurement vector for each block in the image with different length. Moreover, the CS coefficients are entropy coded into a variable-length bit stream.

So, the adversary must break both the entropy coding structure to get the real-valued CS coefficients which is encrypted by AES with 256-bit key. Then, the adversary must try to deduce the variable number of CS coefficients for each block which is also encrypted by AES with 256-bit key. This means the adversary has $2^{256} \times 2^{256}$ tries left to make. This multiple variable-length encoding makes the estimation of sensing matrix very difficult. Even with this information the adversary still must overcome the blocks shuffing obstacle. All this combined make complexity of proposed algorithm encryption extremely high. More details on proposed algorithm encryption strength can be found in Section 5.

4. PROPOSED ALGORITHM

4.1. Proposed Algorithm Description

This paper presents a CS-based algorithm to achieve 3 main objectives, those 3 objectives are as follows:

1. To embed a highly robust watermark against Additive White Gaussian Noise attack (AWGN).
2. To Achieve high compression ratio for images as in JPEG algorithm.
3. To secure CS encryption.

The proposed algorithm takes advantage of Quantization Index Modulation (QIM) to achieve the first objective. While splitting the DCT coefficients into CS coefficients (which holds the sparse watermark along with one QIM watermark), and Entropy coefficients (which are used as a host for two QIM watermarks embedded inside different locations) to achieve the second objective. Finally, the third objective can be achieved by generating random CS projections using Pseudo-

random number secret key SK_1 and shuffing image blocks with Arnold Cat Map using another Pseudo-random number secret key SK_2, in addition to hiding DC-Values within CS coefficients and encrypting Huffman tables using Advanced Encryption Standard (AES) with 256-bit key. The proposed algorithm is implemented within the JPEG framework after the quantization and rounding stage, which makes the blocks sparse.

The proposed algorithm works as follows: at first, we generate PHI, PSI, 2 combination tables for QIM watermark (the first for embedding two QIM watermarks within Entropy coefficients and the second for embedding one QIM watermark within CS coefficients) and two sparse watermark tables (positive and negative). All are generated using SK_1. Then after JPEG quantization stage, image blocks are shuffled using Arnold Cat Map by SK_2. DC coefficients are coded using Differential Pulse Code Modulation (DPCM). Each DPCM coded DC coefficient is represented by (SIZE, AMPLITUDE). After block splitting, the SIZE is inserted within CS coefficients in a fixed place. Each 8×8 block is sliced into two parts, 58 coefficients act as Entropy coefficients which includes 17 coefficients representing the QIM watermark pool, while 6 coefficients act as CS coefficients which will be used for the sparse and one QIM watermarks embedding as well as compression-encryption stage using CS as shown in Fig.2.

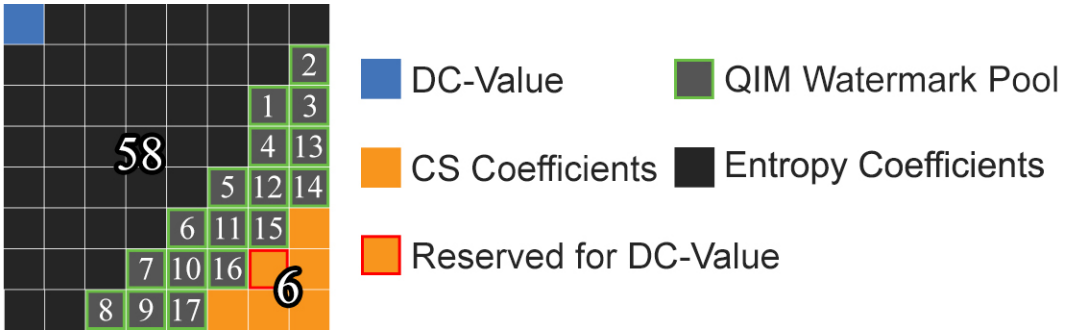

Figure 2. Proposed segmentation of DCT-Blocks.

After that, a more crucial stage comes next; in case of positive embedded watermark, we try to heavily compressing CS coefficient as well as embedding positive sparse watermark from the positive sparse watermark table. the proposed algorithm tries all sparse watermarks within the table and with each one it tries to compress CS coefficients as much as possible by reducing number of rows in the sensing matrix. after finding the sparse watermark which allows for high compression ratio, the proposed algorithm forces QIM watermark to be embedded inside Entropy and CS coefficients using combination corresponding to the non-zero value within sparse watermark from QIM combination tables. In case of negative embedded watermark, the algorithm uses the same previous steps but with negative sparse watermark. at the end, proposed algorithm encrypts Huffman tables using AES with 256-bit key.

An improved technique in embedding QIM watermark is used by the encoder which minimize the risk of losing too much information from the image. This unique technique embeds QIM watermark inversely if number of coefficients needed to be changed is more than a certain threshold. Moreover, the negative sparse watermark (which contains -1 value) is used to inform the decoder that QIM watermark within both Entropy and CS coefficients is embedded inversely.

In watermark detection, the proposed algorithm uses a voting system to enhance the robustness of embedded watermark by using the 3 QIM watermarks embedded within Entropy and CS Coefficient

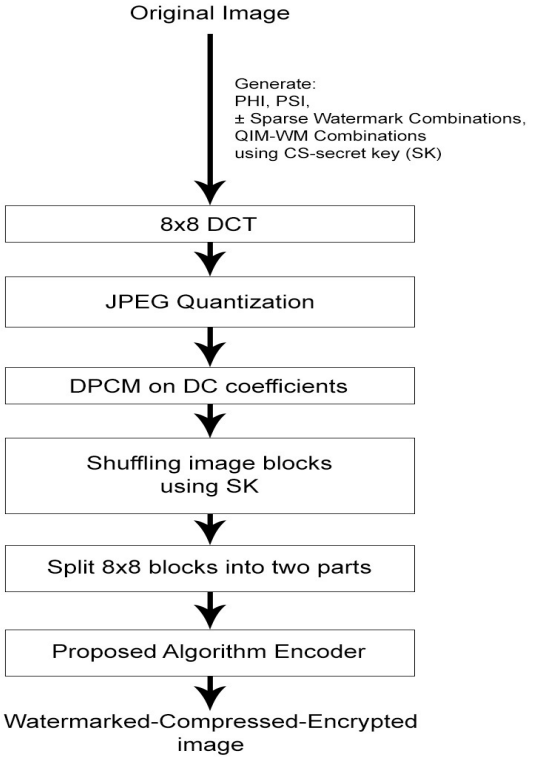

Figure 3. Block Diagram for the proposed algorithm.

4.2. QIM Robust Watermark Description

In our proposed algorithm, we pre-defined 17 coefficients (QIM watermark pool) to be used for watermark embedding. Choosing how many coefficients to be used in watermark embedding was challenging as it will affect the size of image file and the watermark robustness. After experimenting different number of coefficients within the QIM watermark pool to be used in watermark embedding, we choose to embed the QIM watermark in 10 out 17 coefficients, as 10 coefficients give best results in terms of compression ratio and watermark robustness.

The QIM watermark embedding is applied on 10 coefficients out of 17 coefficients (QIM watermark pool) which are picked in a block-by-block basis as shown in Fig.2 so as not to affect the image quality as well as to maximize watermark robustness. Those 17 coefficients located within the Entropy coefficients which consists of 58 quantized DCT coefficients. In detection, the 10 watermarked coefficients will be treated as two separate QIM watermarks (5 coefficients each).

Based on the QIM partitioning, the proposed algorithm decides whether to make a change in the actual coefficients or not to embed the QIM watermark. This partitioning is created by taking the maximum absolute value among all Entropy parts and create incrementing 1 step partitions starting from 0 to the maximum absolute value. Each section in that partition represent +1 or -1 watermark. For example, if the maximum absolute value is 3 then the partitions will be as shown in Fig.4.

Partitions	0	1	2	3
Watermark	1	-1	1	-1

Figure 4. Example for QIM partition dictionary.

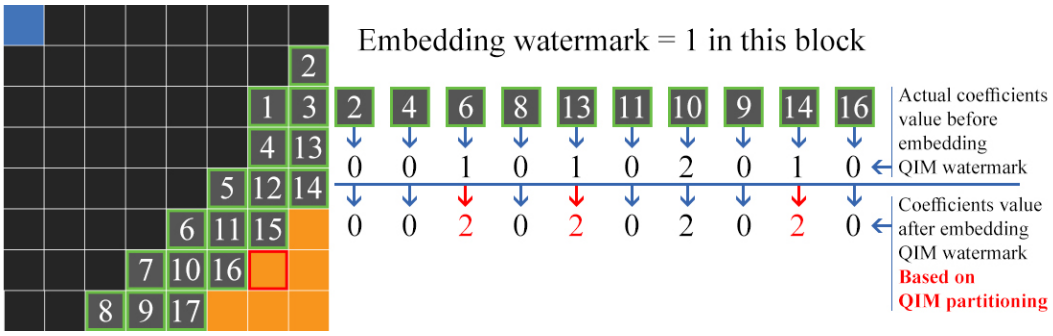

Figure 5. Example for QIM watermark embedding within Entropy coefficients of 1 block

.

The previous example for QIM partitioning is used in Fig.5 to illustrate how the QIM watermark embedding is done.

The same technique is used in embedding the third QIM watermark within CS coefficients. This time our pre-defined pool is all CS coefficients except for coefficient reserved for DC-value to avoid damaging image quality. Next, the proposed algorithm chooses only 3 coefficients out of the remaining 5 to be used for the third QIM watermark embedding depending on chosen combination.

4.3. Sparse Watermark Description

4.3.1. Sparse Watermark Embedding

After picking the best combination for the Entropy coefficients, we come to the next stage where we apply sparse watermark, CS compression and encryption simultaneously. Consider N1 to be the number of CS coefficients (in our proposed algorithm, N1 = 6 as shown in Fig.2), then we generate 2 random matrices; φ with dimensions N2 × N1 and ψ with dimensions N2×N3 (in our proposed algorithm, we use N2 = 6 and N3 = 6). For each block, a brute-force approach is used to get the lowest number of rows for acceptable reconstruction of coefficients and sparse watermark. The sparse watermark embedding and the CS coefficients projection are explained by the following equation:

$$x_w = \varphi\, x + \psi\, w_{sparse} \quad (2)$$

x is vector of selected host coefficients, w_{sparse} is sparse watermark vector that contains a single non-zero value and x_w is the watermarked-compressed encrypted coefficients vector. The partial encryption comes from both the random projection of the selected host coefficients and the addition of the sparse watermark random vector (ψw_{sparse}).

We start simultaneously watermarking, compressing and encrypting CS coefficients with a full sized φ and ψ. We gradually decrease the number of rows in both φ and ψ until we find the minimum acceptable number of rows (compressed vector dimension) for block and sparse watermark reconstruction.

4.3.2. Sparse Watermark Recovery

With the watermarked-compressed-encrypted coefficients vector x_w, we apply basis pursuit denoising (BPDN) algorithm:

$$\begin{aligned} &\text{minimize } \|\beta\|_1 \\ &\text{subject to } \|\alpha\times\beta - x_w\|^2{}_2 \leq \epsilon \end{aligned} \tag{3}$$

Where $\alpha = [\varphi\ \psi]$ and $\beta = [x\ w_{sparse}]$. From estimating β, we recover x and w_{sparse}. The first K elements of estimated β are those of x and the remaining L elements are those of w_{sparse}. Our proposed algorithm uses Sparse Modeling Software (SPAMS) [32] for solving the basis pursuit formulation.

4.3.3. Sparse Watermark Table and Combinations Tables Link

Fig.6 illustrates the link between indices of combinations and sparse watermark tables, choosing one sparse watermark enforces the algorithm to choose combination for the 1st, 2nd and 3rd QIM watermarks embedding with the same index. This constrain is made to minimize the size of side information needed for recovery and watermark detection.

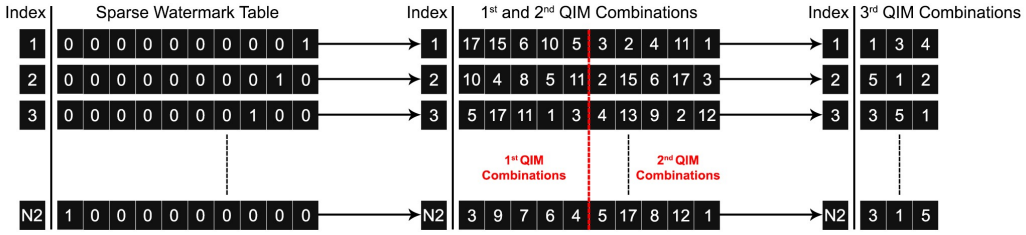

Figure 6. This figure shows how the combinations and the sparse watermark tables are linked.

5. Experimental Results and Analysis

To evaluate performance of the proposed algorithm, we use test images from USC-SIPI Miscellaneous image data set [33] commonly used in benchmarking. The experiment environment platform is Windows 10 operating system of Sony notebook, CPU is Intel Core i5, 4GB of ram, MATLAB R2015a.

We take Baboon image as an example to elaborate the algorithm performance. Testing watermark robustness against AWGN attack, file size, encryption strength and original host image recovery. Encryption strength is calculated by the complexity of finding the right Pseudo-random number SK_1, Pseudo-random number SK_2 and the AES key as-well. For the Pseudo-random number SK_1 and SK_2, each one contains 10 digits, each digit may contain numbers from 0 to 9. This costs the adversary to try 10^{10} combinations to find 1 right key. Not only that, but he/she must decrypt the signal itself which is encrypted with AES using 256-bit key. Breaking a symmetric 256-bit key by brute force attack require to check 2^{256} combinations. The fastest computer built so far is Sunway TaihuLight, a supercomputer developed by China's National University of Defense Technology,

with a performance of 93 petaflops (quadrillions of calculations per second) on the Linpack benchmark. Petaflop is about 10^{15} or 2^{50} floating point operations per second. Thus, this super computer can compute 93×2^{50} operations per second, So, it will be cracked in $2^{256} \div 93\times2^{50}\times365\times24\times60\times60 = 3.5\times10^{52}$ years.

We generate 2 random matrices; φ with dimensions 6 × 6 and ψ with dimensions 6 × 6, both are quantized orthogonal matrices which produces CS coefficients which are more suitable for Entropy coding.

Different types of sensing matrices (ex. Gaussian, Bernoulli, Fourier, etc.) were experimented using our proposed algorithm. Our technique in generating quantized φ and ψ results in better compression ratio with higher PSNR than all other types of sensing matrices due to pre-quantized values of φ and ψ. Using any other type of sensing matrix affect Entropy coding badly, for example; in case of using Gaussian sensing matrix, compressed CS coefficients must be quantized before encoding them using Huffman coding. Quantization levels determine the compression ratio and PNSR of the image. As quantization levels increases, the compression ratio increases but PSNR decreases.

The problem is that CS coefficients were quantized before in JPEG quantization stage. This means that the image already lost some information. Quantizing coefficients one more time before Entropy coding means losing more information. This will result in failing to recover image blocks while maintaining compression ratio and PSNR.

Generating quantized φ and ψ works as follows;

- Generate orthogonal φ with dimensions 6 × 6, and orthogonal ψ with dimensions 6×6.
- Quantize φ and ψ values into 7 levels.

While quantizing φ and ψ enhance the performance of proposed algorithm, more quantization levels will affect badly Huffman coding performance. As more quantization levels leads to more Huffman codes. On the other hand, less quantization levels will produce weaker φ and ψ as well as lowering the compression ratio of proposed algorithm. Less quantization levels mean that the proposed algorithm will need more samples to perfectly recover each block. After experimenting different quantization levels, we got best results using 7 quantization levels.

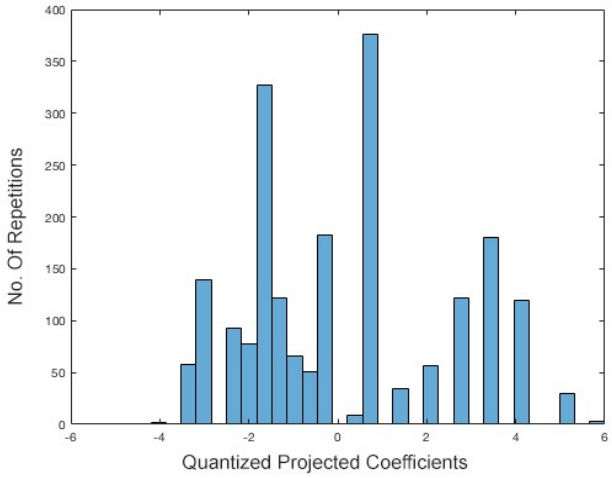

Figure 7. This figure shows histogram of the quantized projected coefficients (Average of 100 realizations on Baboon image).

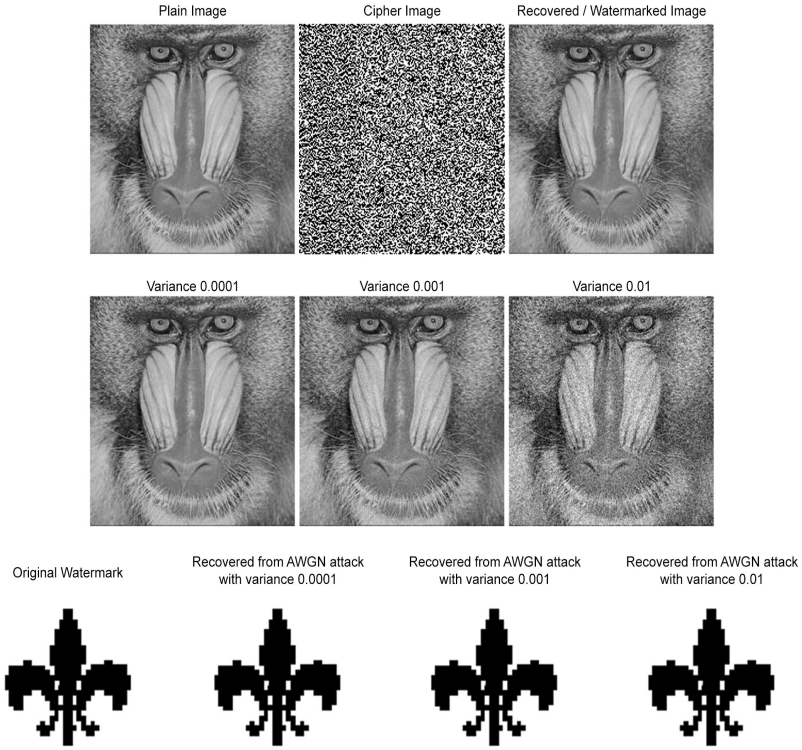

Figure 8. This figure elaborates algorithm performance through an experiment on Baboon image.

As our proposed algorithm works after the JPEG quantization stage and sparse watermark contains only ±1, that results in quantized DCT coefficients and sparse watermark. Quantizing φ and ψ was the next high priority step to enhance Huffman coding performance which produced quantized projected coefficients. Histogram is shown in Fig.7.

All images are resized to 256×256 pixels and we are embedding a 32×32 binary watermark image (1024 bits per image).

5.1. Watermark Analysis

5.1.1. Robust Watermark

(Table. 1) shows that the proposed algorithm can recover embedded watermark perfectly under AWGN attack with different variances by using voting system which takes advantage of all 3 QIM watermarks embedded within the image.

Even under very high AWGN attack (with variance 0.01), proposed algorithm takes advantage of the voting system to successfully detect embedded watermark with 100 %.

Embedding 1,024 bits per 256×256 gray-scale image; which means embedding 0.0156 bpp, is not the highest embedding rate compared to other schemes which uses CS for applying watermarking approach. However, embedding 3 QIM watermarks to be used in watermark detection voting

system, and sparse watermark as a side information gives the proposed algorithm the edge in term of watermark robustness.

Table 1. Watermark Detection.

Before Attack		
Test Image	**Watermark Detection**	**PSNR (dB)**
Baboon	100 %	28.7
Plane	100 %	33.4
Lena	100 %	34.2
Peppers	100 %	34.9
House	100 %	35.6
Lake	100 %	31.4
Splash	100 %	37.2
Tree	100 %	31.2
After AWGN Attack with 0.01 variance		
Baboon	100 %	19.4
Plane	100 %	20
Lena	100 %	19.9
Peppers	100 %	20
House	100 %	19.9
Lake	100 %	19.8
Splash	100 %	20.1
Tree	100 %	19.8

5.1.2. Fragile Watermark

In case of embedding fragile watermark, proposed algorithm can increase watermark embedding capacity up to 6 times. This can be done by making each QIM watermark hold a different value, in addition to increasing number of non-zero values inside the sparse watermark up to 2 values. Which means that sparse watermark can point to 2 different combinations within each set of coefficients (Entropy and CS coefficients). So proposed algorithm can embed up to 6,144 bits per 256×256 gray-scale image.

5.2. Compression Analysis

To test the proposed algorithm compression ratio performance versus PSNR, a comparison between proposed algorithm and JPEG was conducted and tabulated in (Table. 2). The proposed algorithm shows a very close to identical compression ratio and PSNR with JPEG. These results were conducted by using the proposed algorithm as a compression/encryption technique without embedding watermark in either CS or Entropy parts.

Embedding watermark usually expands image file size. In proposed algorithm watermark is embedded using four watermarks, three QIM watermarks and one sparse watermark.

Table 2. Compression ratio for proposed algorithm vs JPEG.

Size in bits		
Test Image	**Proposed**	**JPEG**
Baboon	79155	79171
Plane	56242	56241
Lena	51941	51942
Peppers	53954	53953
House	38858	38857
Lake	70478	70477
Splash	37650	37649
Tree	69878	69877
PSNR (dB)		
Baboon	29	29
Plane	33.5	33.5
Lena	34.3	34.3
Peppers	34.9	34.9
House	35.6	35.6
Lake	31.9	31.9
Splash	37.2	37.2
Tree	31.2	31.2

(Table. 3) shows that after embedding all this amount of information, expansion rate does not exceed 31% from JPEG image file size. Proposed algorithm expands image file size to provide security and watermarking capabilities. To do that, it costs proposed algorithm expansion rate up to 31%.

Table 3. Proposed algorithm watermarked, compressed and encrypted image file size.

Test Image	**Proposed (Size in bits)**	**Expansion Rate vs. JPEG**
Baboon	89595	13 %
Plane	67489	20 %
Lena	61980	19 %
Peppers	64329	19 %
House	50903	31 %
Lake	81445	15.6 %
Splash	48927	30 %
Tree	81541	16.7 %

Table 4. Comparison between compression ratio and PSNR for proposed algorithm and another scheme

.

Test Image	Compression Ratio		
	Proposed	[7]	[35]
Lena	8:1	4:1	-
Cameraman	7:1	4:1	-
	PSNR (dB)		
Lena	34.2	25.9	-
Cameraman	29.5	22.6	-
Baboon	29	-	28.9
Lake	31.9	-	29.2
Peppers	34.9	-	32.2

Table. 4 lists PSNR and compression ratio for proposed algorithm, scheme proposed in [7] and scheme proposed in [35]. According to Table. 4, it is evident that the proposed algorithm outperforms the scheme proposed in [7] and [35] in terms of compression ratio and PSNR.

5.3. Encryption Analysis

To verify proposed algorithm encryption strength, the number of changing pixel rate (NPCR) and the unified average changed intensity (UACI) is calculated and tabulated in Table. 5. Those tests are two of the most common quantity used to evaluate the strength of image encryption algorithms/ciphers with respect to differential attacks [34].

The results shown in Table. 5 indicates that the algorithm can resist differential attacks as the NPCR and UACI values are close to the theoretical values of 99.61% and 33.46% respectively. From Table. 5 Proposed algorithms scored a very close NPCR and UACI values compared to other schemes with the advantage of watermarking capability.

Table 5. NPCR and UACI analysis.

Test Image	NPCR		
	Proposed	[35]	[36]
Baboon	99.6214 %	99.8322 %	99.6735 %
Boat	99.5321 %	99.3500 %	99.6674 %
Lake	99.6678 %	99.8718 %	99.6735 %
Man	99.6821 %	99.6704 %	99.5972 %
Peppers	99.5750 %	99.8840 %	99.5911 %
Average	99.6157 %	99.7217 %	99.6405 %
	UACI		
Baboon	33.4606 %	34.1922 %	33.6889 %
Boat	33.2162 %	36.5397 %	33.4745 %
Lake	33.4115 %	38.9861 %	33.8226 %
Man	33.5286 %	30.1514 %	33.3759 %
Peppers	33.2472 %	35.3155 %	33.3975 %
Average	33.3728 %	35.0370 %	33.5519 %

Table. 6 shows PSNR of the test images Baboon, Boat, Lake, Man and Peppers compared with the schemes proposed in [35] and [36]. It is noticeable that proposed algorithm achieves high PSNR compared to other schemes which leads to better reconstructed image quality.

Table 6. PSNR for different test images.

Test Image	PSNR (dB)		
	Proposed	**[35]**	**[36]**
Baboon	28.7475	28.9701	20.5268
Boat	32.4195	28.5048	22.2174
Lake	31.8490	29.2761	19.9962
Man	30.8848	29.0090	20.8783
Peppers	34.9214	32.2387	23.0676
Average	31.7644	29.5997	21.3373

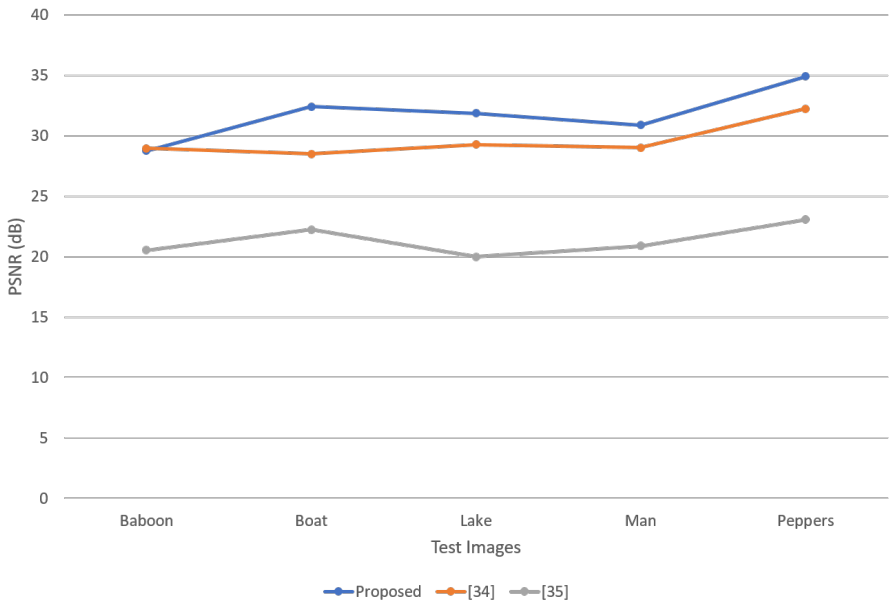

Figure 9. Comparison of PSNR

In Fig.9 a comparison between proposed algorithm and other schemes in term of PSNR is shown using a line graph. This graph clearly shows that proposed algorithm has the capability to achieve higher PSNR with almost every test image.

Table 7. Proposed algorithm compared with other algorithms.

Algorithms	**Watermarking Data Hiding**	**Compression**	**Secure CS Encryption**
Di Xiao et al. [4]	Yes	-	Yes
Rachlin Y et al.. [10]	-	Yes	-
Jelena Musi et al.. [18]	Yes	-	-
Mengmeng Li et al.. [19]	Yes	-	-
Xiao, D., et al.. [20]	Yes	-	Yes
Qia Wang et al.. [21]	Yes	-	-
Gan L [22]	-	Yes	-
Muhammad Ali Qureshi et al.. [23]	-	Yes	-
Yang Y et al.. [24, 25]	-	Yes	-
Gao Z et al.. [26]	-	Yes	-
Dipiti Bhantnagar et al.. [27]	-	Yes	-
Yuan Yuan et al.. [28]	-	Yes	-
Muhammad Yousuf Baig et al.. [9]	-	Yes	-
Nanrun Zhou et al.. [7]	-	Yes	Yes
Valerio Cambareri et al.. [30]	-	-	Yes
Ponuma, R., and R. Amutha [35]	-	Yes	Yes
Proposed Algorithm	Yes	Yes	Yes

(Table. 7) shows that most researchers focus on using CS technique to achieve one or two objectives out of the previously stated objectives (watermarking, compression, secure CS encryption). Some of them takes advantage of CS secrecy and appoint that as a joint compression-encryption algorithm, while others strengthen the encryption by adding some scrambling technique to achieve secure CS encryption.

Our proposed algorithm is the first algorithm that can combine watermarking, compression and secure CS encryption in one algorithm.

6. CONCLUSION

This paper introduces a new algorithm which combines image watermarking along with high image compression ratio and strong image encryption capability based on Quantization Index Modulation, Sparse Watermarking, Compressive Sensing, Arnold Cat Map, Pseudo-random numbers, Advanced Encryption Standard (AES) and Entropy coding.

The proposed algorithm is based on the JPEG standard and operates after the quantization and rounding stage. Experiments shows that our proposed algorithm can provide same compression ratio (without watermarking capability) in comparison to JPEG with an advanced encryption based on secrecy of CS, Pseudo-random numbers, AES and Arnold shuffling. While using full proposed algorithm capabilities expands size by an average of 15% in comparison with JPEG. Also proposed algorithm provides acceptable capacity watermarking with 1024 bits per 256×256 image and is highly robust against AWGN attack due to embedding 3 QIM watermarks. Future work should include doubling the embedding capacity by using sparse watermark with two non-zero values and 4 different combinations per block

REFERENCES

[1] X. Tao, C. Zhao-quan, A novel semi-fragile watermarking algorithm for 3d mesh models, in: Proc. Control Engineering and Communication Technology (ICCECT), 2012 International Conference on, IEEE, 2012, pp. 782–785.

[2] L. Xu-dong, Image digital watermarking algorithm in dct domain for resisting brightness-and-contrast adjusting attack [j], Journal of Optoelectronics. Laser 6 (2013) 027.

[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital watermarking and steganography, Morgan Kaufmann, 2007.

[4] D. Xiao, H. Cai, Y. Wang, S. Bai, High-capacity separable data hiding in encrypted image based on compressive sensing, Multimedia Tools and Applications 75 (21) (2016) 13779–13789.

[5] X. LIU, J. YU, Y. YUE, Y. WEI, A double encrypted digital image watermarking algorithm based on compressed sensing, Journal of Computational Information Systems 10 (12) (2014) 5113–5120.

[6] Q. Zhang, Y. Sun, Y. Yan, H. Liu, Q. Shang, Research on algorithm of image reversible watermarking based on compressed sensing, J Inf ComputSci 10 (3) (2013) 701–709.

[7] N. Zhou, A. Zhang, F. Zheng, L. Gong, Novel image compression encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing, Optics & Laser Technology 62 (2014) 152–160.

[8] S. M. R. Islam, X. Huang, K. L. Ou, Image compression based on compressive sensing using wavelet lifting scheme, The International Journal of Multimedia & Its Applications 7 (1) (2015) 1.

[9] M. Y. Baig, E. M. Lai, A. Punchihewa, Compressed sensing-based distributed image compression, applied sciences 4 (2) (2014) 128–147.

[10] Y. Rachlin, D. Baron, The secrecy of compressed sensing measurements, in: Proc. Communication, Control and Computing, 2008 46th Annual Allerton Conference on, IEEE, 2008, pp. 813–817.

[11] A. Orsdemir, H. O. Altun, G. Sharma, M. F. Bocko, On the security and robustness of encryption via compressed sensing, in: Proc. Military Communications Conference, 2008. MILCOM 2008. IEEE, IEEE, 2008, pp. 1–7.

[12] E. J. Cand`es, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE475 Transactions on information theory 52 (2) (2006) 489–509.

[13] Fakhr, M., 2012. Robust watermarking using compressed sensing framework with application to MP3 audio. The International Journal of Multimedia & Its Applications (IJMA), 4(6), pp.27-43.

[14] D. L. Donoho, Compressed sensing, IEEE Transactions on information theory 52 (4) (2006) 1289–1306.

[15] R. G. Baraniuk, Compressive sensing [lecture notes], IEEE signal processing magazine 24 (4) (2007) 118–121.

[16] R. G. Baraniuk, E. Candes, M. Elad, Y. Ma, Applications of sparse representation and compressive sensing [scanning the issue], Proceedings of the IEEE 98 (6) (2010) 906–909.

[17] E. J. Cand`es, M. B. Wakin, An introduction to compressive sampling, IEEE signal processing magazine 25 (2) (2008) 21–30.

[18] J. Musi´c, I. Kneˇzevi´c, E. Franca, Wavelet based watermarking approach in the compressive sensing scenario, in: Proc. Embedded Computing (MECO), 2015 4th Mediterranean Conference on, IEEE, 2015, pp. 315–318.

[19] M. Li, C. Han, A dct-svd domain watermarking for color digital image based on compressed sensing theory and chaos theory, in: Proc. Computational Intelligence and Design (ISCID), 2014 Seventh International Symposium on, Vol. 1, IEEE, 2014, pp. 35–38.

[20] D. Xiao, Y. Chang, T. Xiang, S. Bai, A watermarking algorithm in encrypted image based on compressive sensing with high quality image reconstruction and watermark performance, Multimedia Tools and Applications 76 (7) (2017) 9265–9296.

[21] Q. Wang, W. Zeng, J. Tian, A compressive sensing based secure watermark detection and privacy preserving storage framework, IEEE transactions on image processing 23 (3) (2014) 1317–1328.

[22] L. Gan, Block compressed sensing of natural images, in: Proc. Digital Signal Processing, 2007 15th International Conference on, IEEE, 2007, pp.403–406.

[23] M. A. Qureshi, M. Deriche, A new wavelet based efficient image compression algorithm using compressive sensing, Multimedia Tools and Applications 75 (12) (2016) 6737–6754.

[24] Y. Yang, O. C. Au, L. Fang, X. Wen, W. Tang, Perceptual compressive sensing for image signals, in: Proc. Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, IEEE, 2009, pp. 89–92.

[25] Y. Yang, O. C. Au, L. Fang, X. Wen, W. Tang, Reweighted compressive sampling for image compression, in: Proc. Picture Coding Symposium, 2009. PCS 2009, IEEE, 2009, pp. 1–4.

[26] Z. Gao, C. Xiong, L. Ding, C. Zhou, Image representation using block compressive sensing for compression applications, Journal of Visual Communication and Image Representation 24 (7) (2013) 885–894.

[27] D. Bhatnagar, S. Budhiraja, Image compression using dct based compressive sensing and vector quantization, International Journal of Computer Applications 50 (20).

[28] Y. Yuan, O. C. Au, A. Zheng, H. Yang, K. Tang, W. Sun, Image compression via sparse reconstruction, in: Proc. Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 2014, pp. 2025–2029.

[29] W. W. W. T. Surveys, Usage of image file formats for website, https://w3techs.com/technologies/overview/image_format/all (2017).

[30] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, G. Setti, Low complexity multiclass encryption by compressed sensing, IEEE Transactions on Signal Processing 63 (9) (2015) 2183–2195.

[31] R. Fay, C. Ruland, Compressive sensing encryption modes and their security, in: Proc. Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for, IEEE, 2016, pp. 119–126.

[32] Sparse modeling software (spams), http://spams-devel.gforge.inria.fr/ (2017).

[33] The usc-sipi image database, http://sipi.usc.edu/database/ (1977).

[34] Y. Wu, J. P. Noonan, S. Agaian, Npcr and uaci randomness tests for image encryption, Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT) (2011) 31–38.

[35] R. Ponuma, R. Amutha, Compressive sensing-based image compression encryption using novel 1d-chaotic map, Multimedia Tools and Applications (2017) 1–26.

[36] G. Hu, D. Xiao, Y. Wang, T. Xiang, an image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications, Journal of Visual Communication and Image Representation 44 (2017) 116–127.

[37] Hagras, E.A., El-Mahallawy, M.S., Eldin, A.Z. and Fakhr, M.W., 2011. Robust secure and blind watermarking based on dwt dct partial multi map chaotic encryption. The International Journal of Multimedia & Its Applications, 3(4), p.37.

AUTHORS

Mohab Mostafa obtained his BSc from the Computer Science Department in 2010 from Modern Academy in Egypt. He obtained his MSc from Arab Academy for Science and Technology and Maritime Transport, Egypt in 2017. He is currently teaching assistant in Modern University for Technology and Information in Cairo, Egypt.

Mohamed Waleed Fakhr Had his Ph.D. at University of Waterloo, Canada, 1994 in the field of "minimum complexity neural networks". Worked at NORTEL speech recognition research Lab, Montreal, Canada from 1994 to 1999 where he was applying HMM and machine learning techniques in speech recognition and language modeling. He joined AAST in 1999 where he has taught, done research and published numerous papers in watermarking, neural networks, pattern recognition, time series prediction, compressed sensing applications, image retrieval and secure computing.