Balancing Compression and Encryption of Satellite Imagery

Ali J. Abboud[1], Ali N. Albu-Rghaif[2], Abbood Kirebut Jassim[3]
[1,2]Department of Computer Engineering, College of Engineering, Diyala University, Iraq
[3]Department of Computer Science, College of Science for Women, University of Baghdad, Iraq

Article Info

Article history:

Received Feb 6, 2018
Revised May 12, 2018
Accepted May 19, 2018

Keyword:

Balanced resources
Hybrid compression
Hybrid security
Image entropy
Satellite imagery

ABSTRACT

With the rapid developments in the remote sensing technologies and services, there is a necessity for combined compression and encryption of satellite imagery. The onboard satellite compression is used to minimize storage and communication bandwidth requirements of high data rate satellite applications. While encryption is employed to secure these resources and prevent illegal use of image sensitive information. In this paper, we propose an approach to address these challenges which raised in the highly dynamic satellite based networked environment. This approach combined compression algorithms (Huffman and SPIHT) and encryptions algorithms (RC4, blowfish and AES) into three complementary modes: (1) secure lossless compression, (2) secure lossy compression and (3) secure hybrid compression. The extensive experiments on the 126 satellite images dataset showed that our approach outperforms traditional and state of art approaches by saving approximately (53%) of computational resources. In addition, the interesting feature of this approach is these three options that mimic reality by imposing every time a different approach to deal with the problem of limited computing and communication resources.

Corresponding Author:

Ali J. Abboud,
Department of Computer Engineering,
College of Engineering,
Diyala University,
Mail Box No. 1, College of Engineering Branch, Baquba Mail Office, Diyala, Iraq.
Emails: ali.j.abboud@gmail.com

1. INTRODUCTION

Satellites are powerful machines to observe rapidly large areas around the earth. The onboard camera sensors enable these artificial moons to collecting high resolution images to be used by remote sensing applications including earth monitoring, agriculture, military, commercial, industrial, astronomy, space, weather, mineral, geology, ocean, water, etc [1]. These applications are restricted by the scarcity of storage, energy and downlink communication resources of satellites. To relax these limitations, compression algorithms are employed to play vital role in reducing the size of extremely large satellite images [2]. Moreover, nowadays terrorists threat the security of many safe nations especially in countries like Iraq and Syria. They can benefit from satellite imagery in their attacks on innocent civilians. Hence, ensuring secure storage and transmission of compressed images through communication channels down to the authorized users is another important issue like compression. There are many proposed algorithms proposed to compress or protect satellite images, however all these algorithms consume high amounts of computational and energy resources for example works in the references [2]-[3]. Also, there is a fast growing trend in the hybrid compression, hybrid encryption or joint compression and encryption approaches for digital images. The main reason for such growth is the benefits that can reap from such combined collaboration between compression and encryption algorithms. For instance**,** Setyaningsih *et al*. [4] presented a good survey about hybrid compression algorithms that show the revenues of mixing compression techniques for reducing the size of images. While Mahmood [5] proposed a hybrid encryption algorithm for medical images to reduce consumed resources. Furthermore, Setyaningsih and Wardoyo [6] proposed a survey about image compression and encryption algorithms. Massoudi [7] proposed an overview that classified joint compression and encryption algorithms into precompression (i.e. encryption and then compression), incompression (compression and encryption are done simultaneously) and postcompression (i.e. compression and then encryption). The aforementioned three surveys agree on the following statements: (1) precompression algorithms class increase the size of encrypted file and badly influence compression performance (2) incompression algorithms class is not format compliance since require modification of encoder and decoder of compression algorithms (3) postcompression algorithms class is compression friendly while earlier two classes are not compression favorable (4) image selective encryption is not secure enough and vulnerable to attacks. Based on these proved statements, postcompression (i.e. compression and then encryption) algorithms class are used in our developed approach. To sum up, the main objective of this paper is to develop an approach that balance satellite image quality and system performance under different load conditions. In other words, it optimizes simultaneously image security and compression adaptively to make them appropriate for specific condition. We can achieve this aim by using hybrid compression and hybrid security together for satellite imagery. According to the best of our knowledge, in contrast to the research works in the literature [8]-[10] which proposed either adaptive compression or adaptive security but not both, this is the first research investigates the usage of adaptive compression and adaptive security simultaneously to provide secure compression of satellite imagery.

1.1. Image compression systems

Image compression is the technology to remove redundant data. It is regarded as a tool to remove irrelevant data from wished image. The image redundancy is classified into three main categories [11]:

a. Coding redundancy: every pixel in the image represented by codes that consists of several bits. In these code bits, there are redundant bits.
b. Spatial and temporal redundancy: this redundancy in the image represents similar pixels in the same column or row. There is no need to store all of pixels but we have to store one these similar pixels.
c. Irrelevant information: this class of redundancy presents in the majority of images. It can be regarded as the data not important for observer in the scene, photo or image.

The image compression algorithms are two main categories: **lossless** and **lossy**. The lossless compression is used to return original image with no error in the image reconstruction process. Medical, archrival, text data and cinema. are examples of applications that need high quality reconstructed images. In contrast, the lossy image compression is used to obtain high compression ratios with indistinguishable loss in the quality of the reconstructed image. Such type of compression algorithms is useful in streaming applications such as multimedia, video, images and audio. In this paper, predictive and Huffman coding algorithms were used for lossless compression, while the lossy version of set partitioning in hierarchical trees (SPIHT) algorithm was used for the lossy compression to obtain high compression ratios.

1.1.1. Predicative coding [11]

Predicative coding is the compression technique to remove irrelevant data based on the principle of coding the difference error between pixel and its predicted value using the following equations:

$$e(n) = f(n) - \tilde{f}(n) \qquad (1)$$

$$d_{i,j} = x_{i,j} - x_{i-1,j} \qquad (2)$$

The predicted pixel value is calculated from neighboring pixels of the desired pixel using anyone of the following example predictors:

$$x_{i,j} = x_{i-1,j} \qquad (3)$$

$$x_{i,j} = x_{i-1,j} + x_{i,j-1} + x_{i-1,j-1} \qquad (4)$$

For simplicity in this paper, predictor in Equation (3) was used in our experiments for its simplicity and efficiency.

1.1.2. Huffman coding algorithm [11]

Huffman coding algorithm is one of the most known algorithms for lossless data compression. It is used mainly for removing unwanted coding data from each image pixel. Then represent all pixels in the image with optimal and efficient number of symbols for each coded pixel. The steps of Huffman algorithm

can be summarized as follows:
a. Calculate the probabilities of unique pixels in the image by dividing the frequency of each unique pixel on the total number of pixels in the image.
b. Sorting probabilities of unique pixels from highest to the lowest. Then reduce these probabilities by combining the lowest probability pixels into single pixel. This process of reduction continues until reach finally into two probabilities only.
c. Coding the reduced pixels with smallest probability to the original raw probabilities before combination.

1.1.3. SPIHT algorithm

SPIHT algorithm is a wavelet transform based compression algorithm. It consists of three main steps (1) transform signal (2) quantize signal coefficients (3) code coefficients. The wavelet transform decomposes the image into four subands (LL, LH, HL and HH) at the first level of decomposition. The transform can continue towards depth at further levels as required to obtain more detailed information about image. The tree structure of wavelet subands at various scales can be exploited to develop efficient and robust compression encoders and decoders as SPIHT have done. The main idea of SPIHT algorithm is to transform most significant coefficients based on the sorted magnitudes of these coefficients after partitioning the sets of coefficients at different scales according to their degree of significance. After that, the sorted coefficients are compared with a series of decreasing thresholds to determine whether each one of these coefficients is significant or not. To sum up, the core concepts in the SPIHT algorithm are (1) transform image and sort coefficients partially (2) splitting coefficients into sets according to their importance based on predetermined thresholds (3) transforming sorted bits (4) similarity among wavelet subband scales. For more information about this algorithm please refer to [12].

1.2. Image security systems

Image security is the process of protecting contents of digital images. The information security algorithms are used to secure images by providing confidentiality, authentication and integrity to them [13]. The image confidentiality is transforming image into another form which is only understandable by sender and receiver. While image authentication is testing the origin of the image if it is from expected originator or not. Lastly, image integrity is checking the contents of the image is reached as it (i.e. in intact) without any modification to the desired receiver. In this paper, encryption and decryption technologies were used to provide image confidentiality. RC4, Blowfish and AES well known security algorithms were used in our experiments to encrypt and decrypt satellite images. These algorithms are explained briefly as follows:

1.2.1. RC4 algorithm

RC4 algorithm is the lightweight prominent symmetrical stream cipher. It is developed by Ronald Rivest for RSA security company to be used in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) standards. Also, it is used in the IEEE 802.11 wired equivalent privacy (WEP) protocol standard. This algorithm is fast in encrypting data with the generated key using stream byte generator. It key length from 1 to 256 bytes. For more details about this algorithm can refer to the reference [13].

1.2.2. Blowfish algorithm

Blowfish algorithm is symmetrical block cipher algorithm developed by Bruce Schneier in 1993. The main characteristics of this algorithm are:
a. Fast implementation on microprocessors
b. Compact in use of memory
c. Simple structure eases analysis and implementation
d. Variable security by varying key size from 32 bits to 448 bits.

It has a 64-bit block size and a variable key length. It has also 16 round Feistel cipher and uses key-dependent S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. Besides, the algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. We have to mention that Twofish and Threefishes are new versions of this algorithm. For more details about this algorithm can refer to the reference [14]

1.2.3. Advanced encryption standard (AES) algorithm

Advanced encryption standard (AES) algorithm is 128-bits symmetrical block cipher. It is selected in competition which hold by the national institute of standards and technology (NIST) in 2001 as the best algorithm among several cryptographic algorithms. Security, cost and implementation were the criteria for selection best security algorithm in the competition. It is non-Feistel cipher uses key size 128, 192, or 256 bits depending on the number of rounds which might be 10, 12, or 14. AES consists of four transformations to convert 16-bytes data block successively in aim to provide best confusion and diffusion. These transformations are substitution, shift rows, mixing and key adding. All these transformations use key size of 128-bits. However, there is a key size expansion algorithm that can convert keys sizes less 128-bits into 128. AES repeat these steps into a defined number of times based on the number of rounds. This algorithm is reversible like other cryptographic algorithms. We have to mention that AES algorithm is called also (RIJNDAEL) according to its inventors (Vincent Rijmen and Joan Daemen). For more details about this important standardized algorithm refer to [15].

The rest of the paper is organized as follows: Section 2 is used to describe satellite image dataset, Section 3 is devoted to explain proposed approach, the results and their analysis presented in Section 4, security analysis is explained in Section 5. Finally, Section 6 presents conclusion.

2. SATELLITE IMAGES DATASET

We have collected a set of satellite images from earth resources observation and science (EROS) center at (https://eros.usgs.gov/). These images have different characteristics and they grouped into three main folders named (Art, Art 2 and Art3). The format of these images is Thousands of Incompatible File Formats (TIFF) and they are in raw format (i.e. uncompressed). The size of each image is approximately (8000 x 7000) pixels. Figure 1 shows some examples of these images. The number of images in the dataset which are used in our experiments is (126) hundred and twenty-six images. The images have been acquired by (Landsat 5, MODIS, Landsat 7 and ASTER) satellites around the earth.

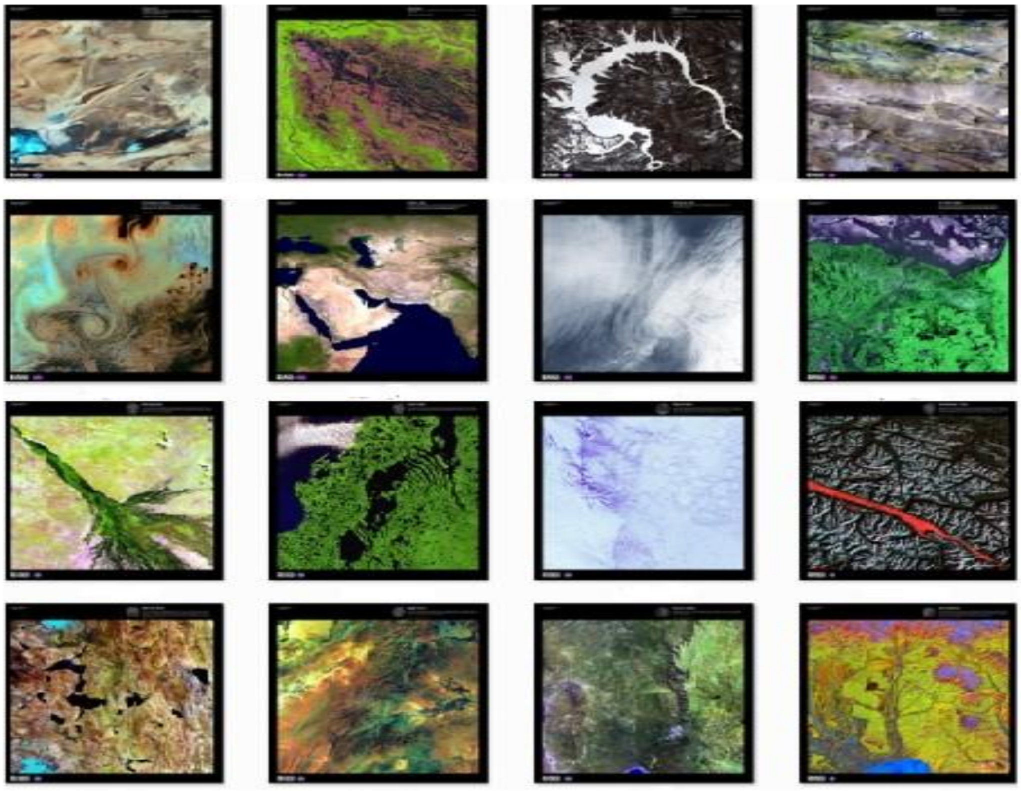


Figure 1. Examples of satellite images dataset

3. RESEARCH METHOD

We propose multiple options approach to jointly compress and encrypt the satellite imagery. It consists from three options (1) secure lossless compression, (2) secure lossy compression and (3) hybrid secure compression. The first option proposed to provide secure lossless compression for remote sensing applications that strictly require no loss in the information of satellite image (i.e. best quality images). While the second option proposed to satisfy the needs for secure lossy compression for satellite application working under strict network load conditions (less resources available) that require high compression ratios and fastest data transfer rates and acceptable low quality images. Lastly, the third option proposed to balance adaptively hybrid compression and encryption of satellite images. The last option, it means dynamic use of compression and security algorithms depending on the local characteristics of satellite images. Eventually, the third option can be regarded as an appropriate choice to obtain good quality satellite images quickly in case of scarce resources. In the following, the unified pseudo code and Figure 2 are used to thoroughly describe proposed method:

```
PROPOSED METHOD : Balancing Compression and Encryption of Satellite Imagery
1: Input: Uncompressed Satellite image.
2: Divide satellite image into a master grid of equal size blocks.
3: Determine importance of areas within satellite image automatically by measuring the information contents of each block using entropy measures.
4: Select one of three options below based on the available communication bandwidth, transfer speed, storage requirements and visual quality specifications.
    4.1: Option 1 (secure lossless compression): Compresses each block of satellite image using lossless compression algorithm (Huffman algorithm). After compressing all blocks of satellite image, use the same encryption/decryption algorithm to secure all compressed blocks. The encryption algorithm will be one of three adopted security algorithms (RC4, Blowfish or AES).
    4.2: Option 2 (secure lossy compression): Compresses each block of satellite image using lossy compression algorithm (SPIHT algorithm). After compressing all blocks of satellite image, use the same encryption/decryption algorithm to secure all compressed blocks. The encryption algorithm will be one of three adopted security algorithms (RC4, Blowfish or AES).
    4.3: Option 3 (hybrid secure compression): Select either lossless Huffman or lossy SPIHT compression algorithm for each block depends on its entropy level, available storage and communication resources. After compressing the blocks of satellite image, different encryption/decryption algorithm is selected adaptively based on the entropy level of each block to secure its compressed bit stream. The encryption algorithm will be one of three adopted security algorithms (RC4, Blowfish and AES).
5: Output: Compressed and encrypted bitstream of input satellite image.
```

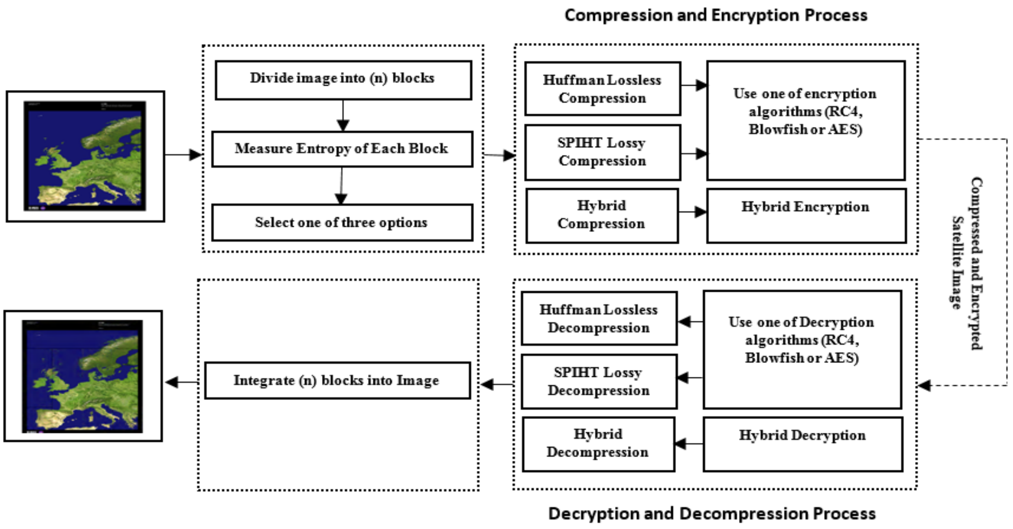

Compression and Encryption Process
Divide image into (n) blocks
Measure Entropy of Each Block
Select one of three options
Huffman Lossless Compression
SPIHT Lossy Compression
Hybrid Compression
Use one of encryption algorithms (RC4, Blowfish or AES)
Hybrid Encryption
Compressed and Encrypted Satellite Image
Huffman Lossless Decompression
SPIHT Lossy Decompression
Hybrid Decompression
Use one of Decryption algorithms (RC4, Blowfish or AES)
Hybrid Decryption
Integrate (n) blocks into Image
Decryption and Decompression Process

Figure 2. Block diagram of the proposed method

4. RESULTS AND ANALYSIS

A comprehensive set of experiments were conducted in aim to assess and compare our proposed method with traditional methods and state-of-art approaches. We obtained experimental results by applying proposed method on the satellite image data set explained earlier in section 2 by measuring them quantitively using the measures described in subsection 4.1. The analysis of these results is divided into three subsections. Secure lossless compression option is explained in subsection 4.2. While secure lossy compression option is described in subsection 4.3. In addition, the last subsection 4.4 is devoted to explain the secure hybrid compression option. All experiments were implemented on the Intel Core i7 processor PC of 2.7 GHz speed and 8GB RAM memory.

4.1. Performance measures

In this section, we elaborate the performance metrics which have been used to quantify system performance quantitively (or objectively). In addition, we presented the subjective assessment of our method in the Figure 8 using some examples of satellite images.

The performance measures are explained as follows:

4.1.1. Information entropy measure

Information entropy measures quantify the amount of information contained inside symbol. Also, it can be defined as a measure of randomness or uncertainty of random variable. The concept of this measure suggested by Claude E. Shannon to represent randomness of information in the communication systems. The mathematical model of this measure for information source (m) of probability (p) is:

$$H(m) = \sum_{i=0}^{2N-1} p_i \ \log_2 \frac{1}{p_i} \tag{5}$$

4.1.2. Peak-to-signal-ratio measure

The Peak-to-Signal-Ratio (PSNR) is well known metric to quantify objectively the quality of reconstructed image after compression or encryption. It measures the difference between original and reconstructed image pixels. The mathematical models of this measure are:

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right) \tag{6}$$

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2 \tag{7}$$

Where mean square error (MSE) measures the error difference between original image I (i, j) and reconstructed image K (i, j). Also, the variables M and N represent the dimensions of the images I and K and (i, j) represent the indices of these images.

4.1.3. Structural similarity index

Structural similarity (SSIM) index is a new metric to measure the similarity between original image and reconstructed image based on the concepts of human visual system (HSV) [16]. It quantifies quality of reconstructed image by measuring the difference in contrast, luminance and structure. The mathematical model of this measure as follows:

$$SSIM\ (x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8}$$

Where x and y are windows of images X and Y and the explanation of the statistical parameters of this model can be found in reference [16].

$$MISSIM\ (X,Y) = \frac{1}{n} \sum_{i=1}^{n} SSIM\ (x_i, y_i) \tag{9}$$

Mean SSIM is between two images X and Y over (n) windows.

4.1.4. Saved time (%) measure

The percentage amount of saved compression (or encryption) time of proposed method can be quantified objectively using the following metrics [17]:

$$Saved\ Compression\ Time\ \% = \frac{Huffman\ Time - Proposed\ Time}{Huffman\ Time} \times 100\ \% \tag{10}$$

$$Saved\ Encryption\ Time\ \% = \frac{AES\ Time - Proposed\ Time}{AES\ Time} \times 100\ \% \tag{11}$$

Where Huffman time is the amount of consumed time by Huffman lossless compression algorithm while AES time is the amount of consumed time by AES encryption algorithm.

4.2. Information entropy anaylsis

To enable developing hybrid compression (or encryption) approaches, there is a need to analysis the information contents (entropy) of satellite images. In these experiments, we found that there are three main entropy levels for each satellite image. These levels were calculated by entropy measure described in subsection 4.1.1. and they are classified into low, medium and high entropy levels. The low entropy level represents image regions that contains less amount of information and features. While, the medium level represents image regions that contains moderate amount of information and details. Lastly, high level contains the highest amount of information, features and details in the satellite image. Figure 3 shows the three entropy levels of 126 satellite images dataset. It is apparent from this Figure that Huffman compression algorithm produced the biggest compressed image size for all entropy levels among compared compression algorithms. The main reason for this result is that Huffman algorithm removes only bits coding redundancy of image pixels. We have noticed also that the relation between entropy level and compressed image size length is linearly dependent for all compression algorithms. In addition, the SPIHT lossy compression algorithm produced the smallest compressed image size among compression algorithms. Lastly, it is apparent from this Figure that the adaptive (or hybrid) compression algorithm produced moderate images sizes because it removes images redundancy dynamically (i.e. using hybrid of compression algorithms).

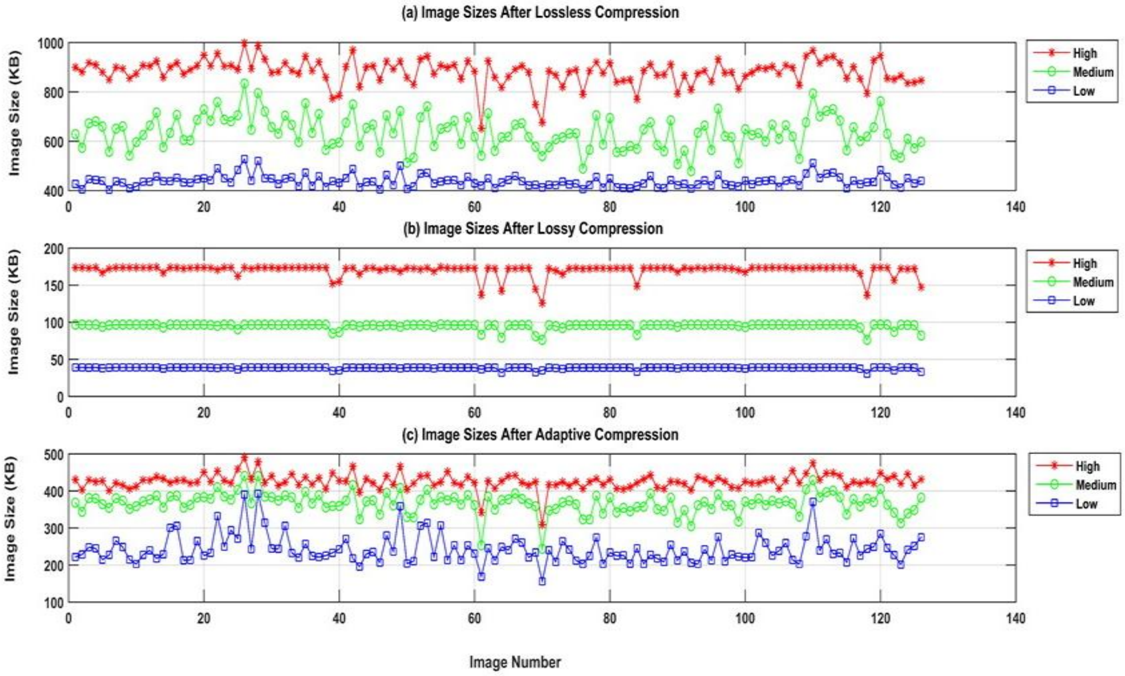


Figure 3. Satellite images sizes for three entropy levels after compression using three options

4.3. Secure lossless compression (Option 1)

The main objective of these experiments is to investigate the performance of secure lossless compression in terms of speed performance and satellite image quality. Huffman lossless compression algorithm was used to reduce size of images. After compression, one of cryptographic algorithms RC4, Blowfish or AES is applied on the compressed satellite image.

Figure 4(a) and Figure 4(b) show that the encoding and decoding times of Huffman algorithm are increasing gradually. Such harmonic behavior of compression and decompression times is a result of increasing number of important features and details in the image gradually from low level towards high entropy level. We also found that the decoding time (170 msec) of Huffman algorithm is three times more than its encoding time (48 msec) because this algorithm need more time to reconstruct original satellite image.

The results of encryption and decryption times for all adopted security algorithms are shown in the Figure 4(c)-(h). Based on the analysis of results in the Figure 4(c) and Figure 4(d), we can notice that the trend of these results show that RC4 consumes least encryption/decryption times than Blowfish and AES algorithms. The conventional clarification of this result is the low computational complexity of RC4 algorithm and the small program code of this stream cipher. In addition, the direction of the RC4 encryption and decryption times is monotonically increasing with entropy level of the satellite image and the decryption time is almost identical to encryption time for all tested satellite images (700 msec).

Let’s now explain the results of Blowfish algorithm as illustrated in Figure 4(e) and Figure 4(f). The obvious result that the encryption/decryption time of this algorithm is three times more than RC4 algorithm

(700 Vs. 2500 msec). The computational complexity and the number of rounds (16) is the main cause of this boost in the execution time of this algorithm. However, the security of Blowfish algorithm is better than RC4 algorithm. In addition, we found that the equality of encryption and decryption times of the Blowfish algorithm. Also, they have the same trend of RC4 algorithm of increasing monotonically with entropy level. This is a normal result because Blowfish algorithm is symmetric Feistel Block cipher algorithm.

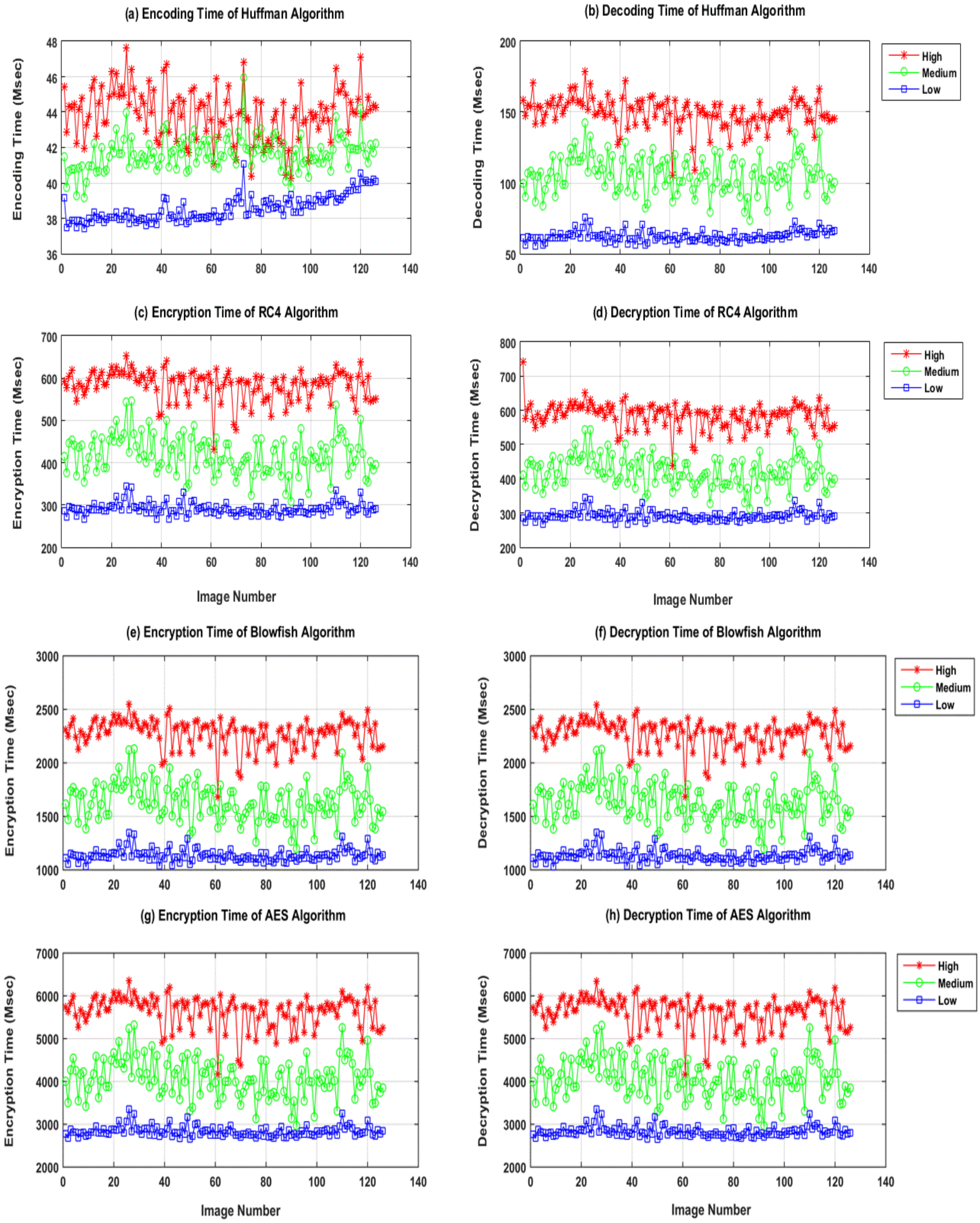


Figure 4. Secure Huffman lossless compression

Finally, we explain the results of AES algorithm as shown in the Figure 4(g) and Figure 4(h). The encryption and decryption times of AES algorithm is nearly three times more than their counterpart times of Blowfish algorithm. Conclusively, the AES algorithm cryptographic times are nine times more than encryption and decryption times of RC4 algorithm. The direction of AES curves likewise the trend of RC4 and Blowfish algorithms of increasing monotonically with entropy level. To sum up, AES algorithm consumes more resources than other two algorithms but it is regarded more secure than RC4 and Blowfish algorithms.

4.4. Secure lossy compression (Option 2)

The aim of these experiments is to investigate the performance of secure lossy compression in terms of speed performance and satellite image quality. SPIHT lossy compression algorithm was used to reduce size of images significantly with loss in the quality of satellite image. After compression, one of cryptographic algorithms RC4, Blowfish or AES is applied on the compressed satellite image. The encoding and decoding times of SPIHT lossy compression algorithm as shown in Figure 5 (a) and Figure 5(b) are progressively escalate in steadily way. Such regular harmonic behavior of compression and decompression times is attributed to the increased number of important features of image gently from low to high entropy levels. We also noticed that the decoding time (25 msec) of SPIHT compression algorithm is less than encoding time (35 msec). The logical illustration of this result is the inherit design of SPIHT algorithm that makes the encoding and decoding times have negligible difference.

Figure 5(c) and Figur 5(d) show encryption/decryption times results of RC4 algorithm on compressed satellite images. The trend of these results demonstrate certainly that RC4 consumes least encryption and decryption times (200 msec) among cryptographic algorithms. The conventional clarification of this result is the low computational complexity of RC4 algorithm and the small program code of this stream cipher as mentioned earlier in the subsection 4.3. In addition, the direction of the RC4 encryption and decryption times is monotonically increasing with entropy level of the satellite image and the decryption time is almost identical to encryption time for all tested satellite images (150 msec). The consumed time resources of this secure compressor are three times less than its counterpart RC4 times resources of lossless compression in subsection 4.3. This significant reduction can be attributed to the less compressed image length code produced by the SPIHT that led to minimize the RC4 encryption and decryption times.

Let's now explain the security times results of Blowfish algorithm as illustrated in Figure 5(e) and Figure 5(f). The obvious result that the encryption/decryption of this algorithm is four times more than RC4 algorithm (150 Vs. 600 msec). The computational complexity and the number of rounds (16) is the main cause of this boost in the execution time of this algorithm as mentioned earlier in the subsection 4.3. However, the security of Blowfish algorithm is better than RC4 algorithm. In addition, we noticed the equality of encryption and decryption times of the Blowfish algorithm. However, these times have the same trend of RC4 algorithm security times of increasing monotonically with entropy level. Hence, the Blowfish speed performance results of this secure compressor is four times less its counterpart in subsection 4.3 (600 msec vs. 2500 msec).

Finally, we explain the results of AES algorithm speed performance in the terms of security execution times as shown in the Figure 5(g) and Figure 5(h). The encryption and decryption times of AES algorithm is nearly three times more than encryption and decryption times of Blowfish algorithm (600 vs. 1600 msec). Conclusively, the encryption and decryption times of AES are nine times more than RC4 algorithm. The direction of AES curves is similar to the trend of RC4 and Blowfish algorithms that increasing monotonically with entropy level. Moreover, AES algorithm in this secure compressor option consumes less resources than its counterpart in subsection 4.3 (1600 vs. 6000 msec). To sum up, we can add another evidence that AES algorithm consumes more resources than other two algorithms but still more secure than RC4 and Blowfish ciphers.

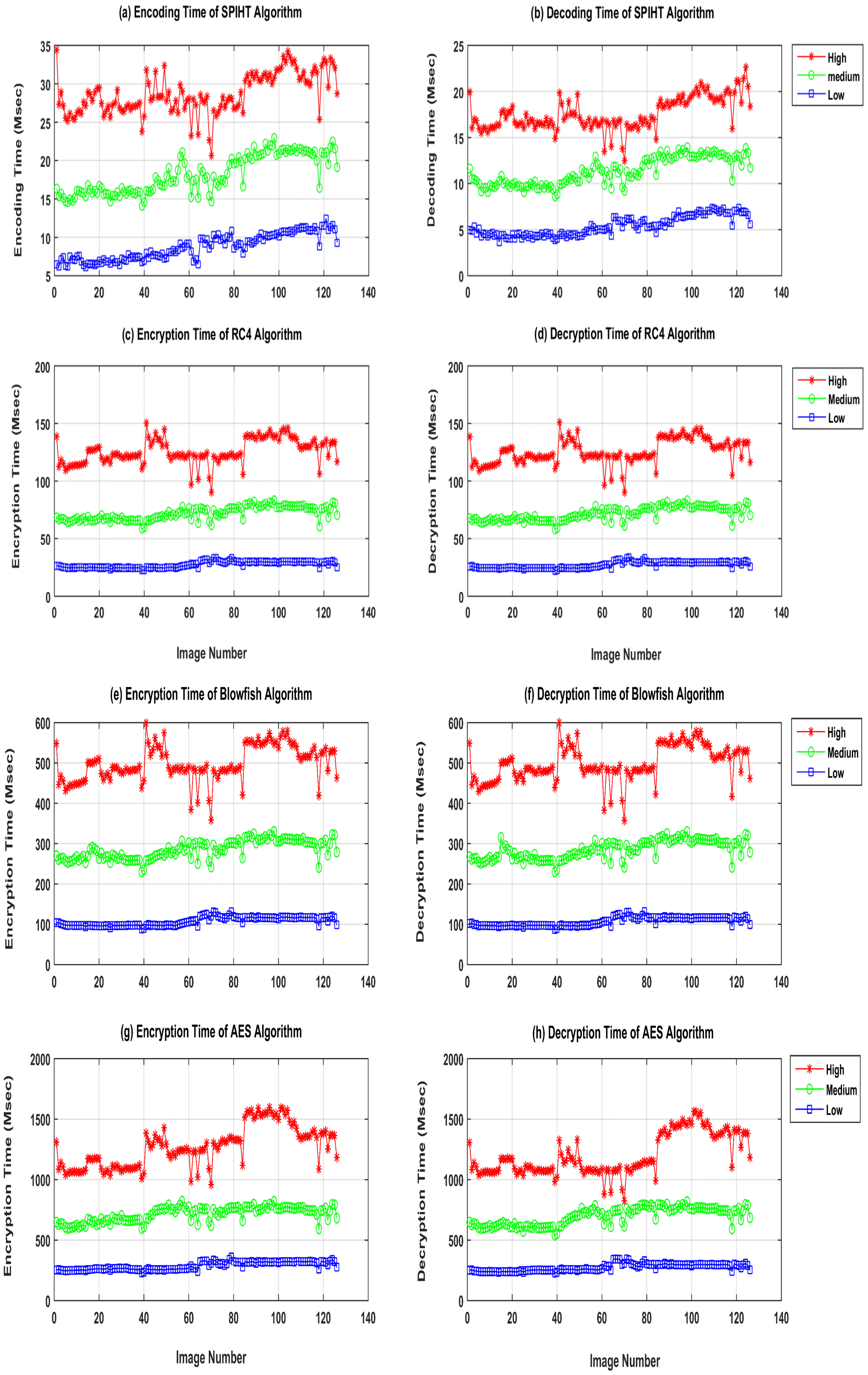


Figure 5. Secure SPIHT lossy compression

4.5. Secure hybird compression (Option 3)

The aim of these experiments is to investigate the performance of secure hybrid compression in terms of speed performance and satellite image quality. The first option of this approach provides apparent proof that secure lossless compression absorbs highest amount of computing resources and simultaneously provides the best satellite image quality. While the second option of secure lossy compression spent least computing resources and usually produce low quality satellite images at high compression ratios. In other words, those two schemes represent the end of two extremes of secure compression. Hence, there is a need to design new option that mix those two options into one option to obtain hybrid (or adaptive) secure compression. This mixed option can reap advantages of first and second options in aim to balance compression and security processes and simultaneously produces good quality satellite images. The balance between those two dominate processes in the computer networks is extremely important for the limitations of computing resources and also increasing number of users of these networks for high quality images. Figures 6 and 7 are used to explain thoroughly the experimental results analysis of secure hybrid compression in the following manner. Figure 6 is employed to show the results of hybrid use of compression algorithms (Huffman and SPIHT). Then this hybrid compression immediately followed by non-hybrid use of security algorithms by applying each one of them (RC4, Blowfish or AES) individually on compressed satellite images. While Figure 7 is employed to present the results of hybrid use of compression algorithms (Huffman and SPIHT) and also hybrid use of security algorithms (RC4, Blowfish and AES) together.

Figure 6(a) and Figure 6(b) show that the encoding and decoding times of secure hybrid compression are progressively escalating in steadily way. Nevertheless, such regular harmonic behavior is almost close for 'High' and 'Medium' entropy levels curves while the 'Low' entropy level is kept separated from them and its shape similar to the 'Low' curve of options 1 and 2. One logical interpretation of these results is the dynamic use of compression algorithms that made compression times almost equal at high and medium entropy levels.

Figure 6(c) and Figure 6(d) show security execution time results of RC4 cryptographic algorithm on compressed satellite images. The consumed time resources of RC4 algorithm in this option is half time less than its counterpart of first option (≈ 350 msec vs. ≈ 700 msec) while it is one time more than its counterpart of second option (≈ 350 msec vs. ≈ 150 msec). Hence, we can conclude that adaptive use of lossless and lossy compression algorithm consumes moderate amount of timing resources. Such kind of use of encryption and compression algorithms make a balanced use of available resources. In addition, we emphasis on the same earlier proved results in the last two options that the direction of the RC4 encryption and decryption times is still monotonically increasing with entropy level of the satellite image. Such assured behavior in our experiments is important and can be exploited in developing adaptive algorithms to solve different kinds problems in image and video processing.

Let's now explain the results of Blowfish algorithm as shown in Figure 6(e) and Figure 6(f). The noticeable result is that the encryption/decryption of this algorithm is four times more than execution times of RC4 algorithm in this option (350 Vs. 1500 msec). As clarified in the last two options, the main reason of this boost in this execution time of Blowfish algorithm is the computational complexity and the number of rounds of this algorithm (16). In addition, we noticed the equality of encryption and decryption times of the Blowfish algorithm and the trend of these cryptographic times is monotonically increasing with entropy level. Finally, the Blowfish encryption/decryption time results of this option are less than of its counterpart of first option (1500 msec vs. 2500 msec) while they are more than its counterpart of second option (1500 msec vs. 600 msec). Lastly, we analyze the encryption/decryption execution time results of AES security algorithm as shown in the Figure 6(g) and Figure 6(h). The execution times of AES algorithm is nearly half execution time of its counterpart in the first option (3500 vs. 6300 msec). However, these execution times are one time more than its identical part in the second option and they are ten times more than RC4 algorithm execution time (350 vs. 3700 msec) in the same option. Also, the direction of AES curves is increasing monotonically with entropy level in similar relevance trend of RC4 and Blowfish algorithms. To sum up, we can add another strong evidence that AES algorithm consumes more resources than other two algorithms but it is regarded recently good security algorithm.

Figure 7 shows the results of hybrid use of security algorithms to secure compressed satellite images. In this Figure, we have found that the adaptive encryption and decryption times are less than average encryption and decryption times of AES algorithm for various image sizes produced by Huffman and SPIHT compression algorithms (adaptive = 2000 msec vs. non-adaptive=3800 msec). However, these times are greater than average encryption and decryption times of blowfish and RC4 security algorithms.

Tables 1 and Table 2 show the performance comparison of second and third options of our developed approach in terms of PSNR and SSIM quality metric values for three entropy levels. We have noticed that the values of image quality measures increase gradually with entropy level. It means that the 'High' entropy level has highest measures values while 'Low' entropy level has the lowest measures values.

In addition, PSNR and SSIM values prove that the quality of images in the secure hybrid option is higher than the quality of images in the secure lossy option. In addition, Figure 8 is added to show the subjective performance evaluation of the proposed approach of three options.

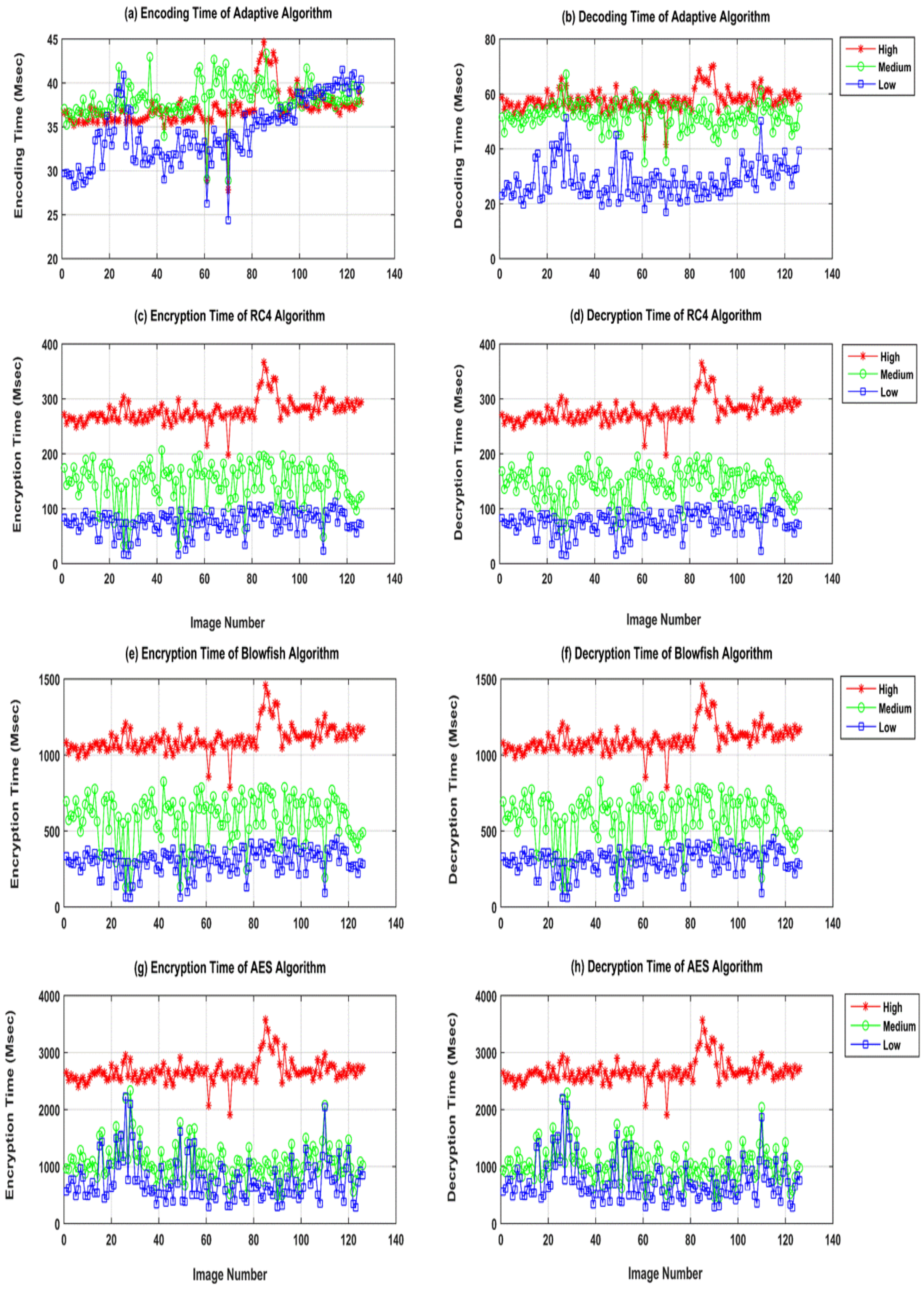


Figure 6. Secure hybrid (or adaptive) compression

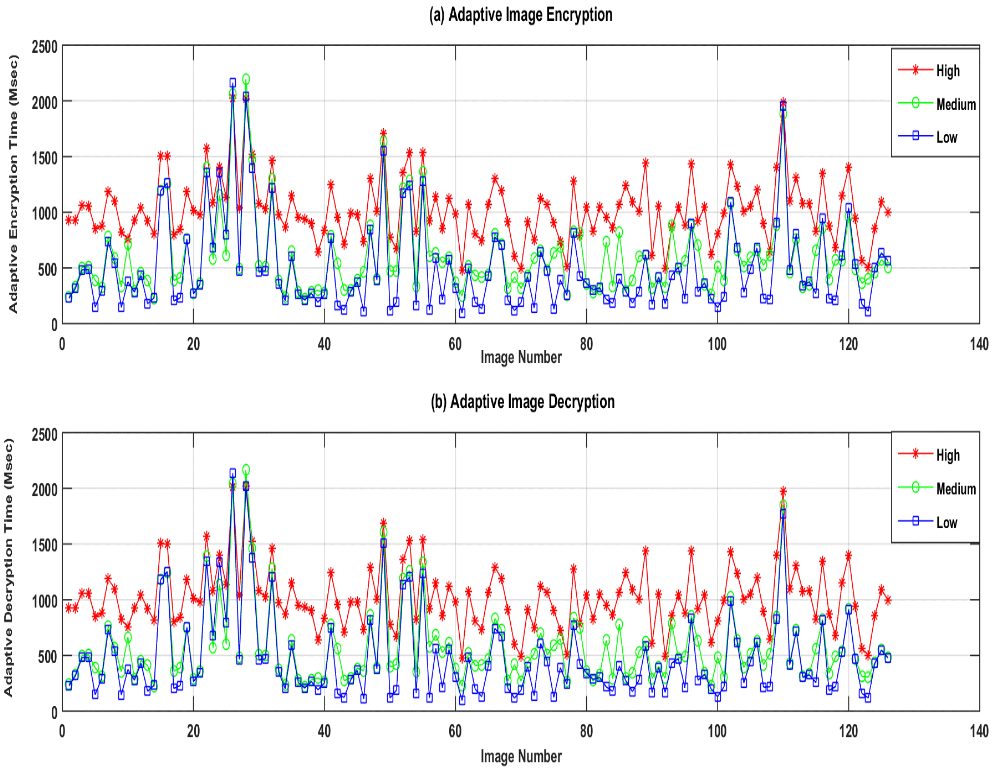


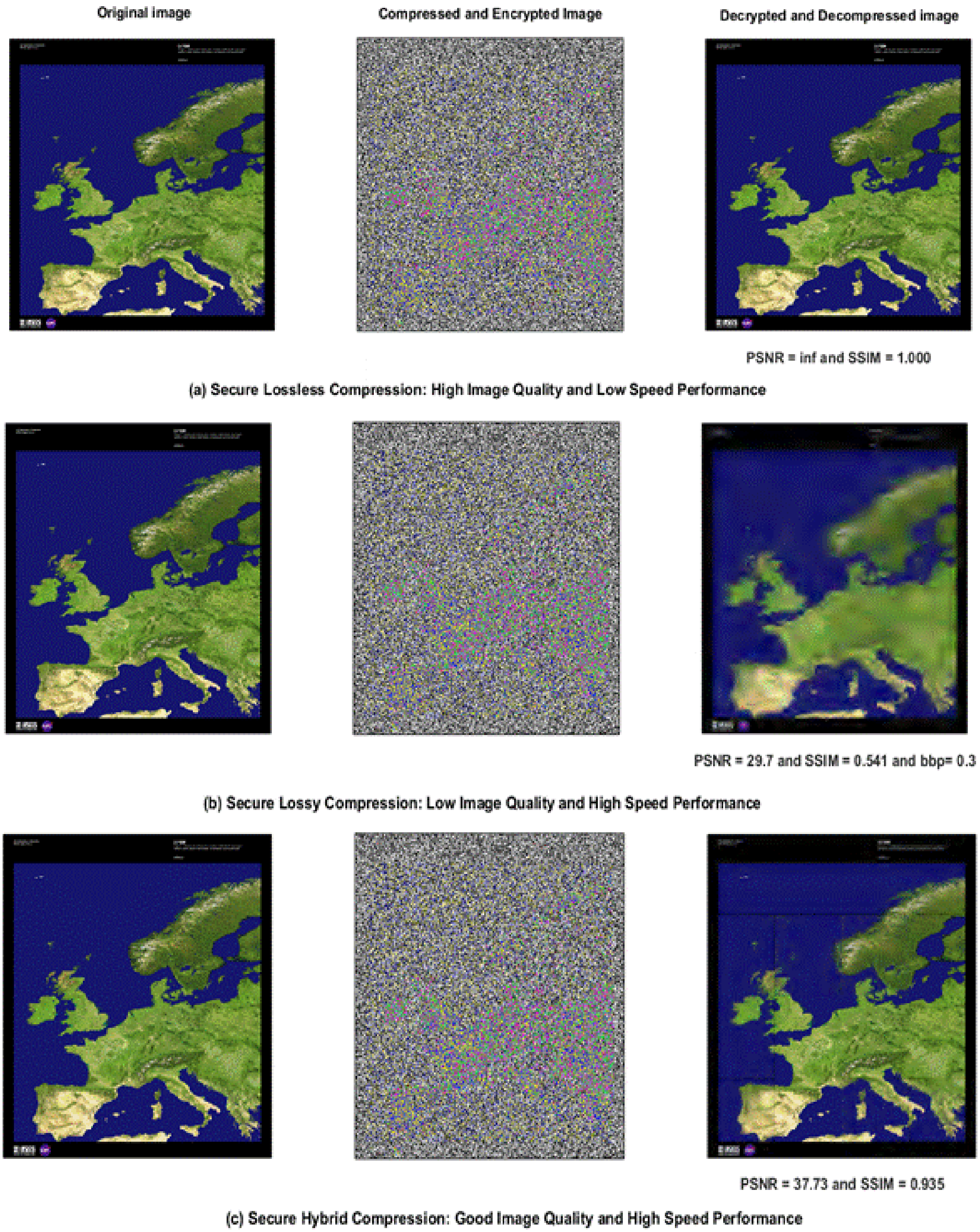
Figure 7. Hybrid security plus hybrid compression

Table 1. The Performance Comparison of Second and Third Options in Terms of PSNR Metric

Measure/ Scenario	Secure lossy compression (option 2)			Secure hybrid compression (option 3)		
	High	Medium	Low	High	Medium	Low
Maximum (Max)	47.438	45.624	42.595	58.310	56.890	49.878
Minimum (Min)	24.112	21.882	19.012	48.415	45.972	28.314
Average (Avg.)	32.492	29.989	27.085	50.362	48.280	38.716
Standard Deviation (Std.)	04.226	04.200	04.005	01.673	01.250	03.361

Table 2. The Performance Comparison of Second and Third Options in Terms of SSIM Metric

Measure/ Scenario	Secure lossy compression (option 2)			Secure hybrid compression (option 3)		
	High	Medium	Low	High	Medium	Low
Maximum (Max)	0.674	0.585	0.475	0.994	0.910	0.828
Minimum (Min)	0.377	0.268	0.168	0.701	0.714	0.447
Average (Avg.)	0.547	0.436	0.287	0.970	0.865	0.656
Standard Deviation (Std.)	0.056	0.055	0.048	0.032	0.040	0.068


Original image
Compressed and Encrypted Image
Decrypted and Decompressed image
PSNR = inf and SSIM = 1.000
PSNR = 29.7 and SSIM = 0.541 and bbp= 0.3
PSNR = 37.73 and SSIM = 0.935

(a) Secure Lossless Compression: High Image Quality and Low Speed Performance

(b) Secure Lossy Compression: Low Image Quality and High Speed Performance

(c) Secure Hybrid Compression: Good Image Quality and High Speed Performance

Figure 8. Subjective evaluation of the proposed approach of three options

4.6. Performance comparative analysis

Table 3 below outline the performance comparative analysis of the proposed approach and traditional methods and state-of-art approaches. The performance of all compared approaches assessed objectively by using earlier explained performance metrics in subsection 4.1. To obtain fair and unbiased performance evaluation comparison, we did not use the direct value of compression or encryption time in the research papers. However, we used the saved time measure of subsection 4.1.4 to calculate the percentage of saved compression or encryption time. In the literature, we found that some articles concerned hybrid compression and other researches concerned adaptive security and other researchers interested joint compression and encryption. We have selected the most relevant and recent papers among them for comparison purposes.

We noticed from Table 3 that our proposed approach outperforms the traditional compression algorithms (Huffman and SPIHT) in terms of saved compression time. Also, it has higher SSIM and PSNR values than lossy SPIHT compression algorithm. Eventually, these results indicate that the proposed approach return good quality satellite images. Furthermore, the saved encryption time by proposed approach is definitely more than saved time by traditional ciphers (Blowfish and AES). To summarize, the proposed approach outperforms significantly the traditional compression and encryption methods.

Ahmed *et al.* [17] developed an adaptive security algorithm for medical images. They employed genetic algorithms, segmentation methods, entropy and several encryption algorithms to achieve their aims of reducing encryption time and also robustness. The percentage of saved encryption time of this algorithm is approximately (37%) while saved time by our approach is (62%). Accordingly, we can insert another proof that our proposed approach is the *best* in saving the encryption and decryption times.

Usama *et al.* [18] proposed to use a family of chaos functions to secure satellite imagery. They compared their approach with traditional DES, 3 DES and AES security algorithms since this research only concerned with security issues. By looking at Table 3, we notice that this approach saved approximately (30%) of encryption time while our proposed approach saved (62%). Hence, our approach *outperforms* significantly this research method in terms of saved encryption time.

Table 3. Performance Comparative Analysis of the Proposed Approach

Approach/Measure	Type of Method (s)	Compression Performance			Encryption Performance	Joint Compression and Encryption Performance
		PSNR	SSIM	Saved Compression Time (%)	Saved Encryption Time (%)	Saved Compression and Encryption Time (%)
Huffman	Traditional Compression	infinity	01.000	00.000 %	-	00.000 %
SPIHT		29.856	0.550	00.000 %	-	00.000 %
RC4	Traditional Encryption	-	-	-	00.000 %	00.000 %
Blowfish		-	-	-	00.000 %	00.000 %
AES		-	-	-	00.000 %	00.000 %
Ahmed Ref [17]	State-of-art Research	-	-		36.900 %	-
Usama Ref [18]		-	-	-	29.540 %	-
Chandan Ref [19]		34.847	0.843	-	-	-
Xiang Ref [20]		31.920	0.869	-	-	-
Morsi Ref [21]		33.000	-	20.000 %	20.000 %	20.000 %
Sahoo Ref [22]		38.870	0.954	-	-	-
Vaish Ref [23]		39.614	0.771	-	-	-
Proposed Method	Proposed	49.878	0.994	33.957 %	62.065 %	53.012 %

Rawat and Meher [19] proposed a hybrid compression algorithm to reduce size of digital images. They have used discrete cosine transform (DCT) and fractal compression to implement their hybrid compression algorithm. In this research, the authors did not mention to the compression and decompression times of their algorithm but they have used PSNR and SSIM metrics to evaluate the performance of their compression algorithm. By comparing the values of those metrics of this algorithm and our proposed approach in Table 3, we notice that our approach outperforms this hybrid algorithm in terms of compression performance and image quality (PSNR=34.847 vs. 49.878 and SSIM=0.843 vs. 0.994).

Xiang *et al.* [20] developed a joint selective encryption and compression approach. They secure compressed image by scrambling part of this image during SPIHT compression coding. By comparing compression performance of this algorithm and our approach in terms of PSNR and SSIM values, we notice that our approach has higher metric values than this algorithm. Also, we could not compare encryption performance for insufficient information about security execution times. Finally, we should emphasis the same point that we have declared in the introduction of our paper that such kind of encryption that modify encoder and decoder of SPIHT algorithm is not compression friendly.

Mostafa and Fakhr [21] proposed joint compression and encryption algorithm based on the entropy measurements and compressive sensing. The DCT coefficients splitted between entropy coding and compressive sensing by using five different algorithms. The security of the compressed image is achieved by random projection of compressive sensing. In this paper only PSNR used as performance measure and also the authors did not mention encryption and decryption times. By comparing results, we found that the percentage of compression time saving and PSNR value are higher than what the authors achieved in their paper (saved compression time = 20% vs. 33.957% and PSNR = 33.000 vs. 49.878).

Sahoo and Das [22] proposed compression algorithm based on the concepts of over training dictionary and sparse representation of residuals. Every block can be predicted from its neighboring blocks and then the residuals encoded via sparse representations. They used JPEG and JPEG2000 as reference algorithms for performance comparison. The authors did not mention the compression and decompression times of their algorithm in this paper. By comparing compression performance in terms of PSNR and SSIM, we notice that the PSNR value of our approach is the *higher* while SSIM value is almost close.

Vaish *et al*. [23] developed joint compression and encryption method for fused images in the wavelet domain. The coefficients of fused images in the wavelet domain are divided into significant and least significant coefficients groups, then each group of these coefficients protected by using different scheme. In this paper, the authors did not mention compression and decompression times however they measure performance of their approach using PSNR and SSIM metrics. We notice by inspecting Table 3 that our approach has higher PSNR and SSIM metric values than proposed method in this paper.

5. SECURITY ANALYSIS

Several statistical security analysis methods are employed to assess and analyze the security of the proposed approach to protect satellite imagery. These methods including histogram analysis, information entropy, correlation coefficient and differential analysis.

5.1. Histogram analysis

Histogram analysis is used to analysis the distribution of pixel intensities among plain and enciphered images. The image histogram is formed by calculating the frequency of each distinct pixel intensity divided by the total number of distinct pixel intensities in the image. The histograms of plain and ciphered images should have no statistical similarities to prevent the flow of any information to the attackers. Figure 9 shows the histograms of plain and encrypted images. This Figure proves that attacker cannot exploit histogram of encrypted image to reveal any information about original image since both histograms are totally different and the histogram of ciphered image is almost flat.

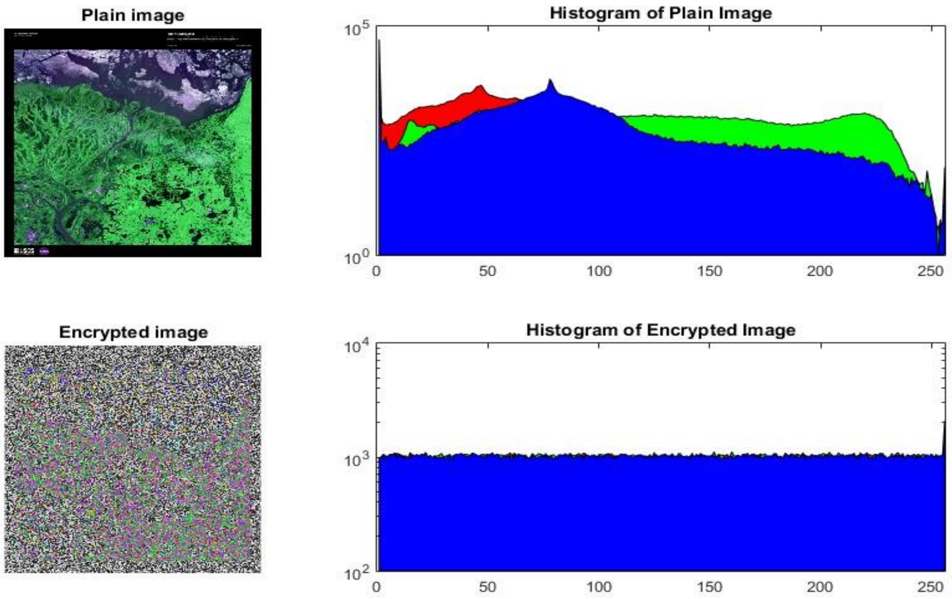


Figure 9. Histograms of plain and encrypted images

5.2. Information entropy

Information entropy measure quantifies the randomness of image information as clarified earlier in subsection 4.1.1. It can also be regarded as a measure of the amount of uncertainty in the random pixel intensities information of both plain and ciphered images. Table 4 below presents the entropy values of an example satellite image. The data in the Table 4 reveals that the color channels of encrypted image have approximately the same entropy value (i.e. the information is distributed equally and randomly among all color channels of an enciphered image). Eventually, our proposed approach is resistance to entropy attacks.

Table 4. Information Entropy

Color Channel	Plain Image	Encrypted Image
R	6.2717	7.9971
G	7.0226	7.9974
B	6.4026	7.9968

5.3. Correlation coefficient

Correlation coefficient is a very powerful tool to determine the strength and direction of the relationship between two random variables. It can be used to specify the correlation between adjacent pixels of the same image. Also, this measure can determine the correlation between plain and ciphered images in vertical or horizontal or diagonal direction. The equations of this statistical method are:

$$r_{xy} = \frac{|\mathrm{Cov}(x,y)|}{\sqrt{D(x)}\times\sqrt{D(y)}} \tag{12}$$

$$\mathrm{Cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\big(x_i - E(x)\big)\big(y_i - E(y)\big) \tag{13}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{14}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{15}$$

Table 5. Correlation Coefficient between Several Encrypted Images

Image 1	Image 2	Correlation Coefficient
Encrypted Image A	Encrypted Image B	0.00073
Encrypted Image A	Encrypted Image C	0.00019
Encrypted Image B	Encrypted Image C	0.00374

Table 5 shows the correlation coefficient between pixels of several encrypted images using slightly changed encryption keys. The results in this table reveal that the correlation among different encrypted images is minimal. Thus, the correlation measure results prove that our proposed approach resistant against statistical attacks.

5.4. Differential analysis

The differential analysis is a statistical method to test the sensitivity of security algorithms. This analysis can sense any change in the plain image (or encryption key) such that incurs big change in the encrypted image. Also, it achieves this objective using two main metrics. First metric is called number of pixels change ratio (NPCR) while second metric is called unified average change intensity (UACI). They are calculated as follows:

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{H\times W} \times 100\% \tag{16}$$

Where H and W are the height and width of an image and D (i, j) is the difference between two images.

$$D(i,j) = \begin{Bmatrix} 1 & E1(i,j) \neq E2(i,j) \\ 0 & E1(i,j) = E2(i,j) \end{Bmatrix} \tag{17}$$

Where E1 is encoding original image and E2 is encoding original image after change one pixel.

$$\mathrm{UACI} = \frac{1}{H\times W}\left[\sum_{i,j}\frac{|E_1(i,j)-E_2(i,j)|}{255}\right] \times 100\% \tag{18}$$

We obtained the data in Table 6 from calculations of NPCR and UACI on three different encrypted images. The high values of NPCR and low values of UACI indicate that the proposed approach is highly sensitive to any change in the input image or encryption key. Eventually, the proposed approach is resistant to the differential attacks.

Table 6. NPCR and UACI Measure Percentages (%)

Image 1	Image 2	NPCR %	UACI %
Encrypted Image A	Encrypted Image B	0.99614	0.33588
Encrypted Image A	Encrypted Image C	0.96605	0.33589
Encrypted Image B	Encrypted Image C	0.96603	0.33495

6. CONCLUSION

To address the challenges of increasing needs for secure transmission of large size satellite images through the internet and satellite based networks, we propose in this paper a balanced approach for secure compression of satellite imagery. The proposed approach consists of three complementary options including secure lossless compression, secure lossy compression and secure hybrid compression. First option is the non-hybrid secure lossless compression that consumes the highest amount of processing resources. However, it produces the best quality satellite images for applications require lossless or near lossless satellite images. Second option is non-hybrid secure SPIHT lossy compression that consumes least resources among proposed options with loss in the quality of satellite image. Also, this approach can be used when there are scarce resources available for satellite applications that need high data transfer rates. Third option is the hybrid secure compression which balances the usage of processing resources and return good quality satellite images. In our opinion, these options mimic reality by imposing every time a different approach to deal with the problem of limited resources. We have to remind that an extensive set of experiments were conducted to test performance, speed and security of the proposed approach. Finally, the proposed approach is not limited to satellite imagery however we can be used for different types of applications that need to store and secure multimedia data over the internet and shared networks. We can also extend this research by implementing its algorithms on the hardware platforms (i.e. FPGA) similar to reference [24] and we may attempt to use this approach for video watermarking and hiding technologies like study in reference [25].

REFERENCES

[1] Petrou M, *et al.*, “Region-based Image Coding with Multiple Algorithms”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 562-570, Mar. 2001.

[2] Faria L. N., *et al.*, “Performance Evaluation of Data Compression Systems Applied to Satellite Imagery”, *Journal of Electrical and Computer Engineering*, pp. 1-18, Jan. 2012.

[3] Barnes R. D. and Gemperline R.C., “Inventors; General Electric Co, Assignee”, Image Data Compression Employing Optimal Subregion Compression, United States patent US 6,792,151. 2004 Sep 14.

[4] Setyaningsih E and Harjoko A, “Survey of Hybrid Image Compression Techniques”, *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 4, pp. 2206-22014, Aug. 2017.

[5] Mahmood A. B. and Dony R. D., “Adaptive Encryption using Pseudo-noise Sequences for Medical Images”, *In 2013 IEEE Third International Conference on Communications and Information Technology (ICCIT)*, vol. 19, no. 1, pp. 39-43, Jun 2013.

[6] Setyaningsih E and Wardoyo R, “Review of Image Compression and Encryption Techniques”, *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, pp. 83-94, Feb. 2017.

[7] Massoudi A, *et al.*, “Overview on Selective Encryption of Image and Video: Challenges and Perspectives”, *EURASIP Journal on Information Security*, vol. 2008, no. 1, pp.179-290, Dec. 2008.

[8] Kolo JG, *et al.*, “An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks”, *Journal of Sensors*, 2012.

[9] Marcelloni F and Vecchio M, “An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks”, *The Computer Journal*, vol. 52, no. 8, pp. 969-987, Nov. 2009.

[10] Mahmood A, *et al.*, “Improving the Security of the Medical Images”, *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 9, pp. 137-149, Sep. 2013.

[11] Gonzalez R. C., Woods R. E., “Digital Image Processing”, Prentice M Hall, 2006.

[12] Said A and Pearlman W. A., “A New, Fast, and Efficient Image Codec based on set Partitioning in Hierarchical Trees”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, Jun. 1996.

[13] Stallings W, “Cryptography and Network Security: Principles and Practices”, Pearson Education India; 2006.

[14] Schneier B, “Description of a new Variable-length key, 64-bit Block Cipher (Blowfish)”, In *International Workshop on Fast Software Encryption*, pp. 191-204, Dec. 1993.

[15] Pub N. F., “197: Advanced Encryption Standard (AES)”, *Federal Information Processing Standards Publication*, vol. 197, no. 441, pp. 03-11, Nov. 2001.

[16] Wang Z, *et al.*, “Image Quality Assessment: From Error Visibility to Structural Similarity”, *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[17] Mahmood A, *et al.*, “An Adaptive Encryption based Genetic Algorithms for Medical Images”, In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, vol. 1, pp. 1-6, Sep. 2013.

[18] Usama M, *et al.*, “Chaos-based Secure Satellite Imagery Cryptosystem”, *Computers and Mathematics with Applications*, vol. 60, no. 2, pp. 326-337, Jul. 2010.

[19] Rawat C and Meher S, “A Hybrid Image Compression Scheme using DCT and Fractal Image Compression”, *International Arab Journal of Information Technology*, vol. 10, no. 6, pp. 553-562, Nov. 2013.

[20] Xiang T, *et al.*, “Joint SPIHT Compression and Selective Encryption”, *Applied Soft Computing*, vol. 21, pp. 159-170, Aug. 2014.

[21] Mostafa M and Fakhr MW, “Joint Image Compression and Encryption based on Compressed Sensing and Entropy Coding”, *In 2017 IEEE 13th International Colloquium on Signal Processing and its Applications (CSPA)*, vol. 1, pp. 129-134, Mar. 2017.

[22] Sahoo A and Das P, “Dictionary based Image Compression via Sparse Representation”, *International Journal of Electrical and Computer Engineering*, vol. 7, no. 4, pp. 1964-1972, Aug. 2017.

[23] Vaish A, *et al.*, “A Wavelet based Approach for Simultaneous Compression and Encryption of Fused Images”, *Journal of King Saud University-Computer and Information Sciences*, 2017 Feb 11.

[24] Ou SC, *et al.*, “Improving the Compression and Encryption of Images using FPGA-based Cryptosystems”, *Multimedia Tools and Applications*, vol. 28, no. 1, pp. 05-22, Jan. 2006.

[25] Ahuja R and Bedi SS, “Robust Video Watermarking Scheme based on Intra-coding Process in MPEG-2 Style”, *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp. 3332-3343, Dec. 2017.

BIOGRAPHIES OF AUTHORS

Ali J. Abboud has received PhD degree in computer science from the University of Buckingham, UK in 2011. Aslo, he obtained M.Sc. degree in computer engineering from University of Technology, Iraq in 2005 and B.Sc. degree in computer and software engineering from Al-Mustansiriyah University, Iraq in 2001. His research intersets focuse on image processing, computer vision, biometrics, machine learning, cryptography and information security. He is working currently as a senior lecturer at department of computer engineering at the university of Diyala since 2005.

Ali Albu-Rghaif has received his PhD degree in Applied Computing from the University of Buckingham, UK in 2015. His work focuses on integrating the receivers of GNSS systems (GPS, Galileo & GLONASS) to enhance product implementation and user localization. He is working as a lecturer at university of Diyala since 2004.

Abood K. Jassim is a lecturer at the university of Baghdad/ College of Science for Women since 2008. He obtained his B.Sc. degree in computer science from university of Basrah in 1989 and M.Sc. degree in Computer Science from University of Technology in 2004. In addition, he obtained his Ph.D. degree in computer science from university of Babylon in 2015. He has taught (and still) different subjects in computer science department He published different research papers in different fields of computer science in variety of international reputed journals in the fields of data mining, data security and data Structures.