# A Proposed Data Security Algorithm Based on Cipher Feedback Mode and its Simulink Implementation

Adnan Mohsin Abdulazeez, Farah Shleemon Khamo

**Abstract**— The issue of security in the process of data transformation is very important so that in this paper we proposed a new two-stage security algorithm which combines cryptography and steganography to increase the secrecy. In the first stage; cryptography stage, the characters of the text (plaintext) are encoded using Cipher Feedback (CFB) mode which uses International Data Encryption Algorithm (IDEA) instead of Data Encryption Standard (DES). In the second stage; steganography stage, the ciphertext is embedded into the cover image using two least significant bits of the middle and high frequency subbands coefficients of the integer to integer lifting wavelet transform of the green and blue components. In this work the proposed algorithm is implemented using MATLB Simulink. Through the software and hardware simulations of the proposed algorithm; the most important properties are improved such as imperceptibility, security, and hiding capacity. Also the results of the software and hardware implementation of the proposed algorithm are approximately similar.

**Index Terms**— — Steganography, Cryptography, Lifting Wavelet Transform (LWT), Cipher Feedback (CFB), International Data Encryption Algorithm (IDEA ), Peak Signal to Noise Ratio (PSNR).

## 1 INTRODUCTION

THE data security can be obtained by using two techniques: cryptography and steganography. A combination of the two techniques can be used to increase the data security. Cryptography is the art and science of protecting information from undesirable individuals by converting it into a non-recognizable form by its attackers while stored and transmitted. Data cryptography mainly is the scrambling of the content of data, such as text, image, audio, video and so forth to make the data unreadable, invisible or unintelligible during transmission or storage, this is called encryption [1]. Steganography is the art and science of hiding information into a host data set so that its presence cannot be detected. In steganography, the secret message is embedded into an image (or any media) called cover image, and then sent to the receiver who extracts the secret message from the cover image. After embedding the secret message, the cover image is called a stego-image. This image should not be distinguishable from the cover image. So that the attacker cannot discover any embedded message [2]. Steganography does not alter the structure of the secret message, but hides it inside a cover image so that it cannot be seen [3]. In this paper, , a new two stage security algorithm which combines cryptography and steganography is presented to enhance the security in data transfer. In the first stage the pay-load (plaintext) is encoded using Cipher Feedback (CFB) mode which uses International Data Encryption Algorithm (IDEA) instead of Data Encryption Standard (DES) as encryption algorithm. This modification gives many enhancements such as increasing the key size from 56- bits key into 128-bit key to increase the security. In addition, it increases the confusion and diffusion of the algorithm. While in the second stage the ciphertext is embedded in cover image using two least significant bit-plain(s) of the integer to integer lifting wavelet transform of the middle and high frequency subbands of the green and blue components of the cover image. The proposed algorithm allows hiding 96 KB in the cover image, with maintaining Peak Signal to Noise Ratio (PSNR) as high as possible. Finally the proposed algorithm is implemented using MATLAB Simulink which provides facilities to convert the design to the Very High Speed Hardware Description Language (VHDL), to implement it on Field Programmable Gate Arrays (FPGAs) device. Furthermore the results of the software and hardware implementation of the proposed algorithm have been compared.

- *Adnan Mohsin Abdulazeez **is** currently assistant professor at Duhok Polytechnic University-Duhok City-Kurdistan Region of Iraq E-mail: Adnan_brifcani@mail.com*
- *Farah Shleemon Khamo is currently assistant lecturer at Department of Computer Science, Faculty of Science, University of Duhok E-mail: Farahshleemon@yahoo.com*

## 2 RELATED WORK

In [4] Guorong Xuan and et al. proposed a novel distortionless image data hiding algorithm based on Integer Wavelet Transform (IWT) that can invert the stego-image into the original image without any distortion after the hidden data are extracted. This algorithm hides data in one (or more) middle bit-plane(s)of the integer wavelet transform coefficients in the middle and high frequency subbands. In [5] KB Raja and et al presented a high capacity, lossless, secure wavelet steganographic algorithm in which pay-load bit stream is encrypted and embedded in the wavelet coefficients of the cover image to derive a stego-image. The pay-load is embedded in the approximation band of the wavelet domain that increases its robustness.

In [6] Shu-Guo Yang and et al. proposed a novel scheme for text information hiding. Firstly, it encodes the characters of the text information by ASCII code and receives plaintext information codes. Secondly, it encrypts the plaintext information

codes by chaotic encryption method and receives encryption information codes. Then it encodes the encrypted information codes by BCH code again before it transmits them, and embeds the ciphertext information in the selected wavelet coefficients of a carrier image in DWT domain.

In [7] R.O. EI Safy and et al. proposed a novel data hiding scheme that hides data in the integer wavelet coefficients of an image. The system combines an adaptive data hiding technique and the optimum pixel adjustment algorithm to increase the hiding capacity of the system compared to other systems. The proposed system embeds secret data in a random order using a secret key only known to both sender and receiver. It is an adaptive system which embeds different number of bits in each wavelet coefficient according to a hiding capacity function in order to maximize the hiding capacity without sacrificing the visual quality of resulting stego image.

In [8] Debnath Bhattacharyya and et al. proposed a security model which imposes the concept of secrecy over privacy for text messages. The model combines cryptography, steganography (taken as security layers) along with that an extra layer of security imposed in between them. This newly introduced extra layer of security changes the format of the normal encrypted message and the security layer followed by it embeds the encrypted message behind a multimedia cover object.

In [9] Vijay Kumar and Dinesh Kumar intended to observe the effect of embedding the secret message in different bands such as horizontal coefficients (CH), vertical coefficients (CV), and diagonal coefficients (CD) of Discrete Wavelet Transform (DWT) in the performance of stegano image in terms of Peck Signal to Noise Ratio (PSNR). Experimentation has been done using six different attacks. Experimental results reveal that the error block replacement wit (CD) gives better PSNR than doing so with other coefficients(CH,CA).

In 2010 Rahman Tashakkori and Christopher D. Sholar [10] presented an information hiding technique that utilizes lifting schemes to effectively hide information in color images. This paper hides the data in three color components of the image this increase the hiding capacity.

## 3 The Proposed Algorithm

The basic structure of the proposed algorithm consists of four major parts, encryption, embedding, extraction and decryption processes. The encryption process encrypt the data (plaintext) using CFB mode to get the cipher text, so the encryption process takes the plaintext, 128-bit key and 64-bit initial vector as input and produces ciphertext as output. The ciphertext is used in addition to the cover image as input to the embedding process which hides the ciphertext in the cover image to produce the stego-image as output. While in extraction process the ciphertext is extracted from stego-image then the decryption process decrypts the extracted ciphertext to produce plaintext. Data encryption part goes through several processes, it begins with reading the plaintext and firstly the characters of the plaintext are encoded by ASCII code and denote it as 7 binary bits secondly, in order to avoid malicious decryption, the plaintext bits are encrypted by using CFB mode which consists several sub-processes. The ciphertext embedding part goes through several processes it begins with reading the cover image then the cover image is decomposed into four subbands such as approximation coefficients CA, CH, CV and CD using integer to integer lifting wavelet transform. Then the detailed coefficients (CD, CH, and CV) are used to hide the ciphertext by replacing the two least significant bits of every detailed coefficient with two bits of the message while the approximation subband (CA) remains unaltered. While the extraction process starts with reading the stego-image and calculates the LWT for this image to get CA, CH, CV, and CD subbands. Then the ciphertext extracted from the CD, CV, CH subbands then the ciphertext goes through the decryption process (CFB) to find the plaintext. The proposed algorithm includes the following:-

### 3.1 Encryption Process

The first stage in the proposed Algorithm is the encryption process in which the original data (pay-load) is encrypted using CFB mode to get ciphertext. The CFB mode procedure consists of several steps to encrypt the plaintext. It takes 64-bits initial vector, 128-bits key and plaintext with any size as input and produces ciphertext with the same size as output. The proposed encryption process starts with opening the text file that contains the plaintext to be encrypt. The plaintext is converted into stream of bits first by taking the ASCII code of each character of the plaintext to produce array of ASCII values then each value is converted into seven binary bits to get a stream of bits. The initial vector is encrypted using IDEA and 128-bits key. Then the leftmost (most significant) seven bits from the encrypted initial vector are selected to be XORed with first seven bits (first unit) of plaintext (P1) to produce the first unit of ciphertext (C1). This is put in the text file to be hidden. In addition, the contents of the initial vector are shifted to the left by seven bits and C1 is placed in the rightmost (least significant) seven bits of the initial vector. This process continues until all plaintext units have been encrypted. The encryption process is shown in Fig.1.

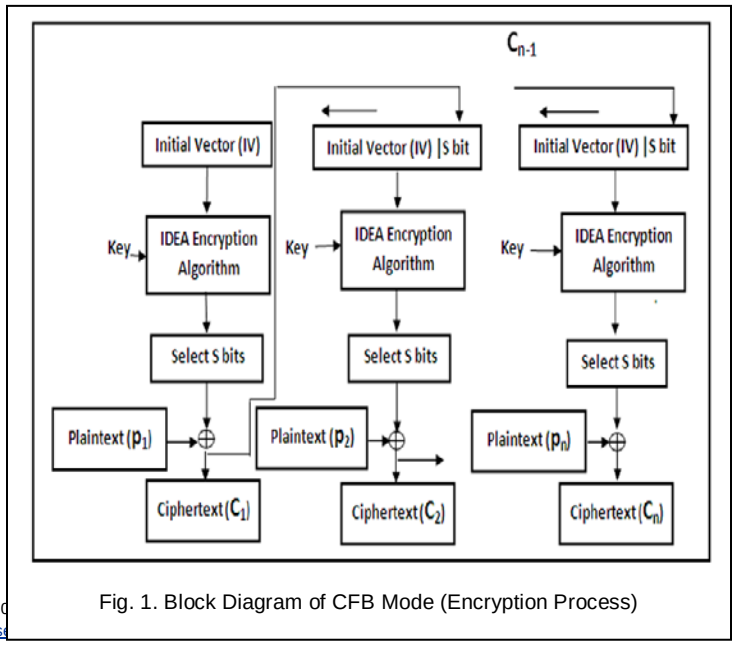


Fig. 1. Block Diagram of CFB Mode (Encryption Process)

### 3.2 Embedding Process

The second stage in the proposed algorithm is the embedding process in which the ciphertext gained from the encryption process is hidden in the details coefficients of the wavelet domain of the cover image (grayscale or color image) to get the stego-image. So the structure of the embedding process is made up of three components: the cover image, the secret message (ciphertext), and the stego-image. This algorithm considers eight-bit grayscale images as well as 24-bit color images as carrier to hold the secret message. This process denotes the least significant bit-plane by the first bit-plane, and the most significant bit-plane by the last bit-plane, and uses second generation wavelet transform which maps integer to integer and the CDF (2, 2) filter. This process is based on the lifting scheme. This process includes eight steps:

i. Extracting three components (red, green, blue) from the color image.
ii. Converting the secret message into stream of bits.
iii. Performing single level wavelet decomposition on green and blue components of the cover image using lifting scheme and CDF (2, 2) to get (LL, LH, HL, and HH) sub-bands for each component.
iv. Checking whether the size of high and middle subbands of the green and blue components is enough to hold the message stream of bits.
v. Partitioning the message stream of bits into two parts; the first part is hidden in high and middle subbands of the green component, while the second part is hidden in high and middle subbands of blue component.
vi. Hiding the message bits in HH, HL, LH, by replacing two least significant bits of HH coefficients with first two bits from the message, and the two least significant bits of HL coefficients with the second two bits from the message, and two least significant bits of LH coefficients with the third two bits from the message, and so on till all bits of message are embedded.
vii. Performing inverse of the lifting wavelet transform to get green and blue component of the stego-image.
viii. Merging the red, green, and blue components to create the stego- image.

### 3.3 Extraction Process

The third stage in the proposed algorithm is the extraction process which takes the stego-image with a number of hiding characters as input to produce the ciphertext as output. This process includes seven steps:

i. Extracting three components (red, green, and blue) from the cover image.
ii. Performing single level wavelet decomposition on green and blue components of the stego-image using lifting scheme and CDF (2, 2) to get (LL, LH, HL, and HH) subbands for each component.
iii. Computing number of hiding bits by multiplying number of characters by seven according to ASCII code.
iv. Dividing the number of hidden bits by two to know the number of bits hidden in high and middle sub-bands of green component, and the number of bits hidden in high and middle sub-bands of blue component.
v. Extracting the message bits from HH, HL, LH, by taking the first two bits of the message from two least significant bits of HH coefficients, and the second two bits from the two least significant bits of HL coefficients, and third two bits from two least significant bits of LH coefficients. This process continues until all message bits extracted from HH, HL, and LH sub-band.
vi. Merging the bits extracted from high and middle subbands of green component, with the bits extracted from high and middle subbands of blue component to create the whole message bits.
vii. Converting message bits into message characters by converting each seven bits into one character.

3.

### 3.4 Decryption Process

The last stage in the proposed algorithm is the decryption process in which the extracted ciphertext is decrypted using CFB to get the original data (pay-load). After extracting the ciphertext from the stego-image, CFB decryption procedure decrypts the ciphertext to get the original data. The CFB decryption procedure is the same as encryption procedure while the decryption procedure uses the ciphertext, 64-bit initial vector and 128-bit key as input to produce the plaintext as output. Decryption process is shown in Fig. 2.

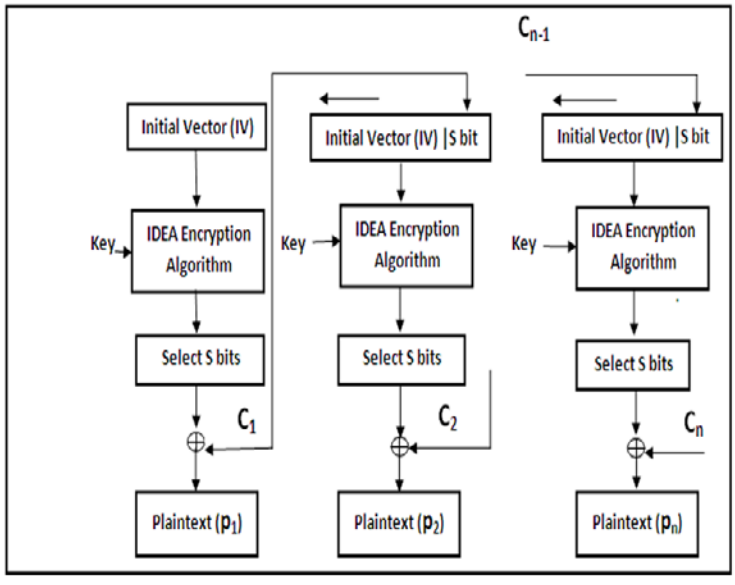


Fig. 2. Block Diagram of CFB Mode (Decryption Process)

## 4 Hardware Implementation of the Proposed Algorithm

In addition to the software implementation of the proposed algorithm, the proposed algorithm is implemented by using MATLAB Simulink which provides many blocksets and many facilities to build any hardware design and convert this design to VHDL language to implement on programmable hardware devices (FPGAs). The design of the proposed algorithm as whole is very complex, and it contains many long operations. The proposed Simulink consists of four major subsystems, the first subsystem encrypts the data (plaintext) by using CFB mode, and the second subsystem hides the text in cover image. The cover image can be a color or a grayscale image. The third subsystem extracts the ciphertext from the stego-image, while the last subsystem decrypts the ciphertext to get the original data (plaintext). The proposed subsystems are built using many blocksets such as math operations blockset, sink blockset, source blockset, user design functions blockset, video and image processing blockset, and data store blockset. So each subsystem consists of many blocks; each block performs one operation in the design.

### 4.1 Encryption Subsystem

The encryption subsystem encrypts the original data (plaintext) to get ciphertext. This subsystem starts with reading the 64–bit initial vector and 128-bit key from MATLAB workspace using "From workspace blocks" which are named Iv, Key respectively. In Simulink, we use 65-bit initial vector and 129-bit key because the "From workspace block" reads data from a workspace as a signal. The block's data parameter specifies the workspace data using two-dimensional matrix: The first element of each matrix row is a time stamp. The rest of each row is a scalar or vector of signal values. The leftmost element of each row is the time stamp of the value(s) in the rest of the row. However the first bit in the 65-bit initial vector is the time stamp and the rest 64-bits are the initial vector values. The encryption subsystem also opens the file which contains the plaintext to be encrypted, and convert the plaintext to a stream of bits by using embedded MATLAB function which is programmable to do this operation. After this operation, the 128-bits key go through the key generation subsystem which uses128-bit key as input using "In block" and produces fifty two (16-bit) sub-keys as output which are used in the encryption steps. Also using "Select row block" to partition the 64-bit into four (16 -bit) sub-blocks and "Bit to integer block" to convert these sub-blocks of bit into four integer numbers to go through the steps of IDEA first round. The encryption subsystem also has "For iterator subsystem" which iterates seven times, "For iterator subsystem" includes the IDEA round steps, then the output of "For iterator subsystem' goes through steps of final transform. The "Selector block" is used to select seven bits from the output of final transform to XORed with seven bits selected from the plaintext stream of bits. The results of "Xor block" represent the first unit of ciphertext. In addition, the initial vector is shifted to the left by seven bits using "shift left subsystem block" which contains two "selectors and vector concatenate blocks". And ciphertext unit is placed in the rightmost (least significant) seven bits of the initial vector using vector concatenate. This process continues until all plaintext units have been encrypted. The encryption subsystem is shown Fig. 3.

### 4.2 Embedding Subsystem

This subsystem starts with loading the cover image by using "Image from file block" as well as opening the text files which contains the ciphertext. Then the ciphertext characters are converted into the stream of bits by using "Transpose and convert 2-D to 1-D blocks". Then the cover image is decomposed into four subbands using integer to integer lifting wavelet transform using the "Embedded MATLAB function block" which is programmable to do this work. In this subsystem, also "If Action subsystem block" is used to check whether the size of the high and middle subbands enough to hold the message bits. "If Action subsystem block" takes the message bits and the CD, CV, and CH coefficients as inputs and uses "Data store memory blocks" which names are CD, CV, CH to store the value of subband coefficients in them. "If action subsystem block" includes the steps of hiding bits into the high and middle subbands. Then the inverse lifting wavelet transform is taken for the CV, CH, CD, and CA to get the stego-image. The embedding subsystem is shown in Fig.5.

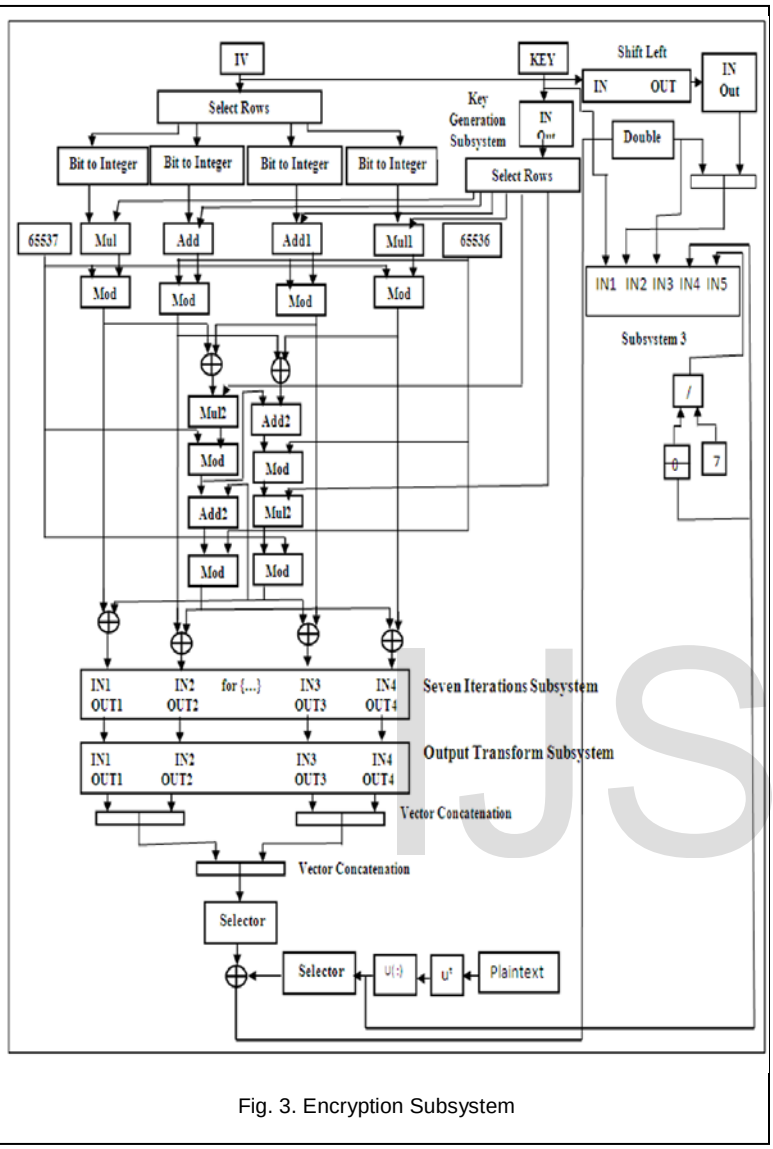


Fig. 3. Encryption Subsystem

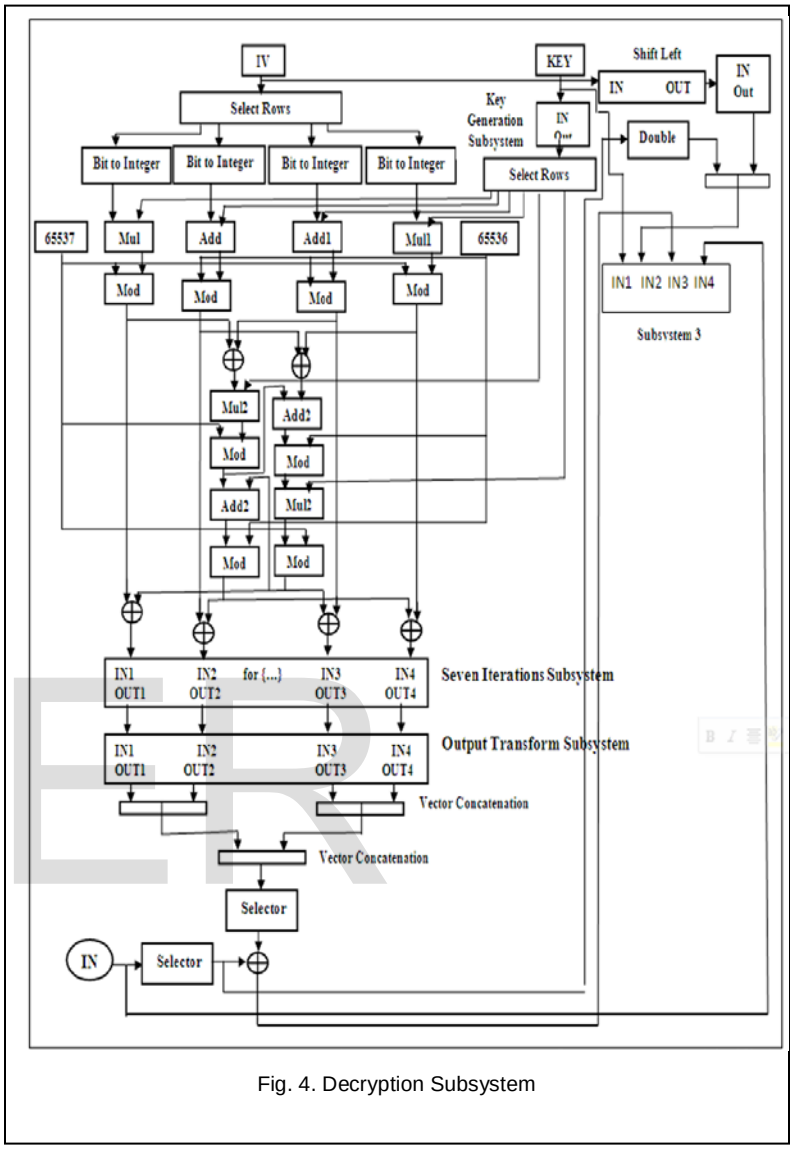


Fig. 4. Decryption Subsystem

subbands coefficients. The extraction subsystem is shown in Fig. 6.

### 4.3 Extraction Subsystem

The extraction subsystem starts with loading the stego-image by using the "Image from file block" and separates the stego-image into three components (red, green, and blue).Then the green and blue components are decomposed into LL, LH, HL, and HH subbands using integer to integer lifting wavelet transform. Extraction subsystem also read the number of characters from workspace by using "From workspace block" which are named No. then compute the number of bits by multiply the number of characters by seven using "constant and Multiplication blocks". Then numbers of bits and the HH, HL, and LH subband coefficients go through the subsystem block which uses the "data store memory" and other blocks to extract the message bits from the high and middle

### 4.4 Decryption Subsystem

For decryption, the same steps of encryption subsystem are used, except that the ciphertext units are XORed with the output of the final transform steps to produce the plaintext units. The decryption subsystem is shown in Fig. 4.

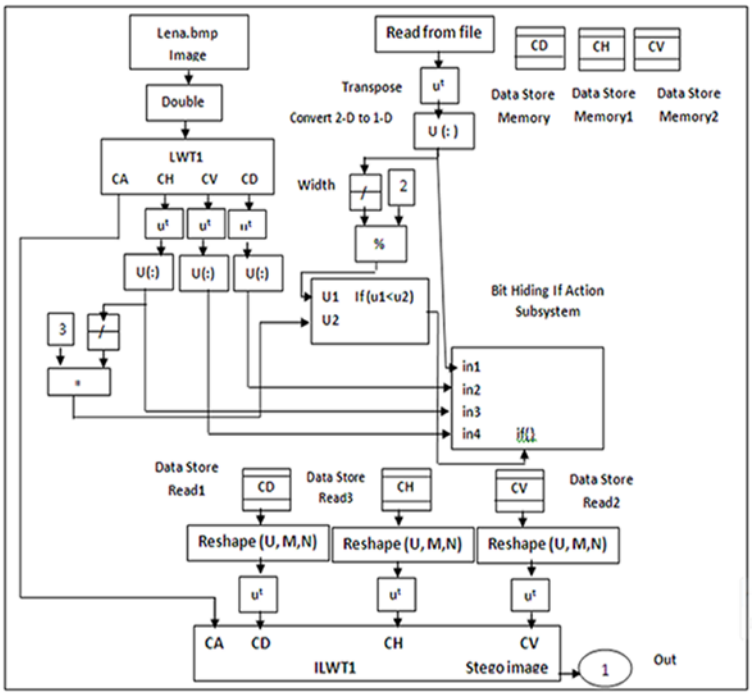


Fig. 5. Extraction Subsystem

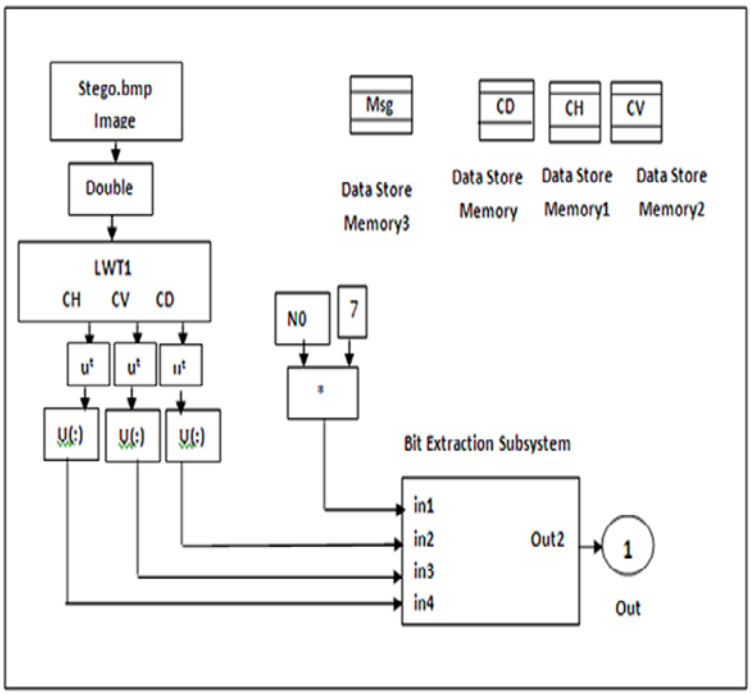


Fig. 6. Extraction Subsystem

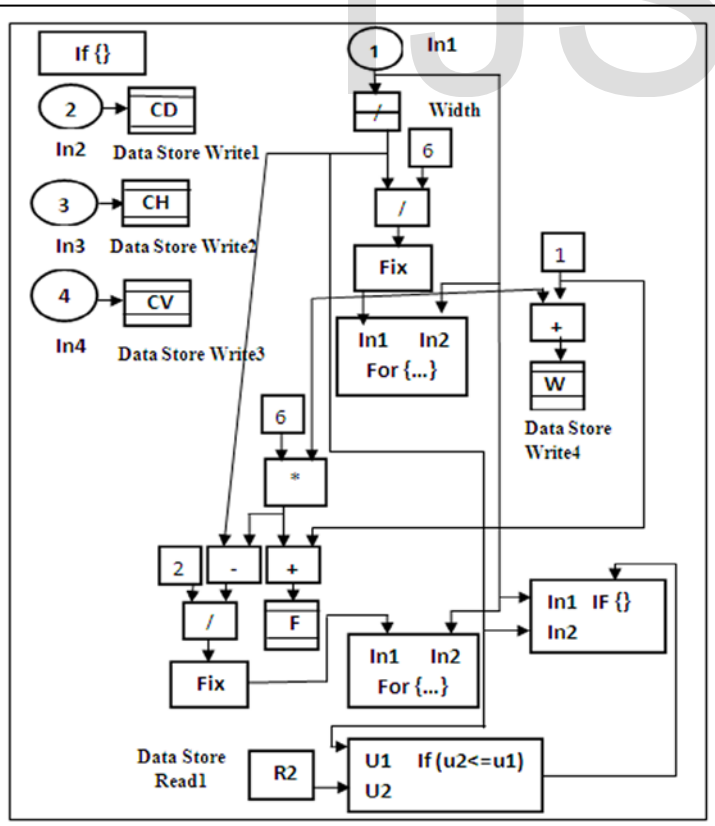


Fig. 7. Bit Hiding If Action Subsystem

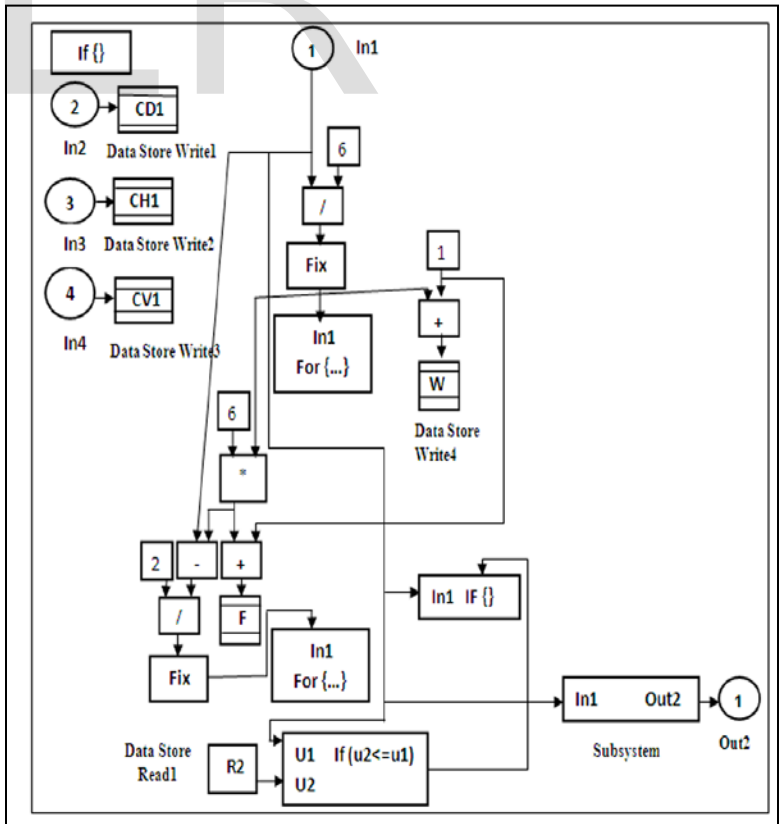


Fig. 8. Bit Extraction Subsystem

## 5. EXPERIMENTAL RESULTS

To test the imperceptibility and security of the proposed information hiding algorithm; one of the visual quality metrics which provides the facility of measuring the distortion in the stego-image comparing to the cover image has been used. The PSNR is the standard quantitative measurement to compare the image with the original image. The proposed information hiding algorithms have been tested using eight typical 512x512 8-bit grayscale images and two 512x512 24-bit color images as shown in Fig. (9 and 10) respectively. All tests have been done by executing application on a PC with, 2 GHz Core 2 duo CPU, 2 GB RAM, and windows XP professional edition-SP2. The algorithm is implemented with MATLAB R2010a and MATLAB Simulink. Table I shows the results of hiding the data using two least significant bits from CD, CH, and CV subband coefficients. From Table I it is obvious that the maximum size of the message hidden in the cover image using this method is (56173) characters, (393211) bits with PSNR ranging between (44.0990 dB, 44.8884 dB) for the eight test images. So the proposed algorithm could increase the capacity and at the same time keeping the PSNR as high as possible. Table II presents the results of hiding data in two least significant bits of high and middle subband coefficients of green and blue components of Lena_rgb.tif and test4.bmp images. The results of this table show that the maximum capacity in this method is (112140) characters (784980) bits with a PSNR 46.6756 dB and 46.7073 for Lena_rgb.tif and test4.BMP respectively. And the Simulink results of hiding the data using the grayscale and color images as a cover images are presented in Tables (III and IV) respectively. From the results of the four Tables (I, II, III, and IV), it is obvious that the results of the software and hardware simulation are approximately similar.

TABLE 1
Experimental Results Using Grayscale Images

| Images (512*512*8) | Message Size (Characters) | Pay-load (Bits) | PSNR (dB) |
|---|---|---|---|
| Lena.JPG | 56173 | 393211 | 44.5779 |
| Barbara.GIF | 56173 | 393211 | 44.4776 |
| Girl.BMP | 56173 | 393211 | 44.0990 |
| Pentagon.JPG | 56173 | 393211 | 44.8591 |
| Man.GIF | 56173 | 393211 | 447928 |
| Harbour.GIF | 56173 | 393211 | 44.7000 |
| Boat.BMP | 56173 | 393211 | 44.8147 |
| Tank.BMP | 56173 | 393211 | 44.8884 |

TABLE 2
Experimental Results Using Color Images

| Images (512*512*24) | Message Size (Characters) | Pay-load (Bits) | PSNR (dB) |
|---|---|---|---|
| Lena_rgb.TIF | 112140 | 784980 | 46.6756 |
| Test4.BMP | 112140 | 784980 | 46.7073 |

TABLE 3
Simulink Results of Hiding Pay-load in High and Middle Subband Coefficients Using Grayscale Images

| Images (512*512*8) | Message Size (Characters) | Pay-load (bits) | PSNR (dB) |
|---|---|---|---|
| Lena. JPG | 56173 | 393211 | 44.5779 |
| Barb.GIF | 56173 | 393211 | 44.0462 |
| Girl.BMP | 56173 | 393211 | 44.0989 |
| Pentagon.JPG | 56173 | 393211 | 44.0777 |
| Man.GIF | 56173 | 393211 | 44.7928 |
| H.GIF | 56173 | 393211 | 44.5005 |
| Boat.BMP | 56173 | 393211 | 44.3431 |
| Tank.BMP | 56173 | 393211 | 44.8884 |

TABLE 4
Simulink Results of Hiding Pay-load in High and Middle Subband Coefficients Using Color Images

| Images (512*512*24) | Message Size (Characters) | Pay-load (bits) | PSNR (dB) |
|---|---|---|---|
| Lena_rgb.TIF | 112140 | 784980 | 46.6550 |
| Test4.BMP | 112140 | 784980 | 46.7071 |

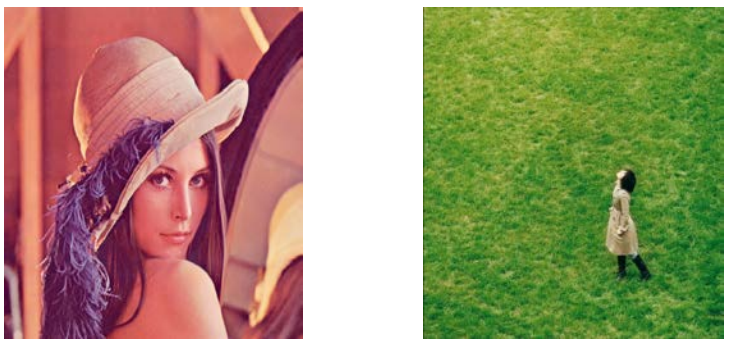

Fig. 9. The Color Images Used For Testing

A. Barbara.GIF

B. Lena.JPG

C. Girl.BMP

D. Pentagon.JPG

E. Man.GIF

F. Harbour.GIF

G. Boat.BMP

H. Tank.BMP

## 6. CONCLUSION

From the results of the software and hardware simulations of the proposed algorithm, the conclusions are summarized as follows:

1. For cryptography stage, the proposed algorithm improves the CFB mode by using IDEA instead of DES as encryption algorithm. This modification increases the security of the CFB mode by increasing the key size from 56-bit key to 128-bit key, also increases the confusion of the algorithm due to mixing three different operations.
2. For steganography stage, the proposed algorithm gives better performance, and better imperceptibility regarding to the high PSNR value, also the pay-load is retrieved at the receiver without any error so that the algorithm is lossless data recovery.
3. From the capacity point, the proposed algorithm provides high hiding capacity.
4. The design of the proposed algorithm using MATLAB Simulink is very complex. Therefore many embedded MATLAB functions which are programmed to do some functions that are not available in MATLAB Simulink blocksets such as lifting wavelet transform block and shift to the left block have been used.
5. The results of software and hardware simulation of the proposed algorithm are approximately similar.

## REFERENCES

[1] M. Abomhara, Omar Zakaria and Othman O. Khalifa , "An Overview of Video Encryption Techniques", International Journal of Computer Theory and Engineering, Vol.2, No.1, pp 103-110, 2010.

[2] Debnath Bhattacharyya, Arpita Roy, Pranab Roy, and Tai-hoon Kim, "Receiver Compatible Data Hiding in Color Image", International Journal of Advanced Science and Technology, Vol. 6, pp.15-24,2009.

[3] Amin M.M., Salleh M., Ibrahim S., Katmin, M.R. and Shamsuddin M.Z.I., "Information Hiding Using Steganography", IEEE National Conference on Telecommunication Technology, pp. 21 – 25, 2003.

[4] Guorong Xuan, Jiang Zhu, Jidong Chen, Yun Q. Shi, Zhicheng Ni and Wei Su, "Distortionless Data Hiding Based on Integer Wavelet Transform", Electronics Letters, Vol.38, No.25, pp.1646-1648,2002.

[5] Raja K.B., Vikas, Venugopal K.R. and Patnaik L.M., "High Capacity Lossless Secure Image Steganography using Wavelets", IEEE International Conference on Advanced Computing and Communications, pp.230-235, 2006.

[6] Shu-Guo Yang, Chun-Xia Li and Sheng-He Sun," Text Information Hiding Method Based on Chaotic Map and BCH Code in DWT Domain of a Carrier Image", IEEE the First International Symposium on Data, Privacy, and E-Commerce, pp.239-241, 2007.

[7] R.O. El Safy, H. H. Zayed and A. EI Dessouki, "An Adaptive Steganographic Technique Based on Integer Wavelet Transform", International Conference on Networking and Media Convergence, pp. 111–117, 2009.

[8] Debnath Bhattacharyya, Poulami Das, Samir Kumar Bandyopadhyay and Tai-hoon Kim, " Text Steganography: A Novel Approach", International Journal of Advanced Science and Technology, Vol.3, 2009, pp.79-86.

[9] Kumar V. and Kumar D., "Performance Evaluation of DWT Based Image Steganography", IEEE 2nd International Advance Computing Conference, pp. 223 – 228,2010.

[10] Tashakkori R. and Sholar C.D., "Message Encoding in Images Using Lifting schemes", IEEE Proceedings of the SoutheastCon, pp. 444-447, 2010.