ISSN: 0067-2904

Authentication of Digital Video Encryption

Asmaa Hasan Mohsen[1]*, Shaimaa Hameed Shaker[2]

[1]Department of Computer Sciences, University of Technolog, Baghdad, Iraq
[2]Department of Computer Engineering, University of Technolog, Baghdad, Iraq

Abstract

The security of multimedia data becoming important spatial data of monitoring systems that contain videos prone to attack or escape via the internet, so to protect these videos used proposed method combined between encryption algorithm and sign algorithm to get on authenticated video. The proposed encryption algorithm applied to secure the video transmission by encrypt it to become unclear. This done by extract video to frames and each frame separate to three frames are Red, Green, and Blue, this frames encrypt by using three different random keys that generated by a function for generating random numbers, as for sign algorithm applied for authentication purpose that enable the receiver from sure of the identity of the sender and provide secure communication between two communication parties.In the proposed method applied compression method before encryption algorithm on video by using discrete wavelet transform (DWT) to facilitation work of encryption algorithm because compression less size of video and this make execution time of encryption algorithm faster. The decryption algorithm and decompression method are big prove to success of proposed method that give target video without any noise, besides that has been verified from performance of proposed method using several quality measures performance of proposed method.

Keywords: Video encryption, Video compression, Digital signature, Mean Square Error(MSE), and Peak Signal to Noise Ratio (PSNR).

المصادقة على تشفير الفيديو الرقمي

أسماء حسن محسن[1]*، شيماء حميد شاكر[2]

[1] قسم علوم الحاسوب ، الجامعة التكنولوجية، بغداد، العراق
[2] قسم هندسة الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة

توفير الأمن لبيانات الوسائط المتعددة اصبح امراً ضروريا وخصوصاً بيانات أنظمة المراقبة التي تحتوي على فيديوات فهي أما معرضة للهجوم أو الأختراق عبر الانترنيت، ولحماية هذه البيانات أستخدمت طريقة مقترحة التي تجمع بين خوارزمية التشفير وخوارزمية التوقيع للحصول على فيديو مشفرو موقع، تم تطبيق خوارزمية التشفيرعلى الفيديو ليصبح غير واضح ولتأمين نقل الفيديو،وهذا تم بواسطة فصل الفيديو الى Frames وكل Frameتفصل الى Three Frames هي Red, Green, and Blue تشفرباستخدام ثلاثة مفاتيح عشوائية مختلفة التي ولدت بواسطة دالة لتوليد الارقام العشوائية . أما خوارزمية التوقيع طبقت لغرض التخويل التي تمكن المتلقي من التأكد من هوية المرسل وتوفير اتصال آمن بين طرفي التواصل .

*Email: asmaa.hasan92@yahoo.com

في الطريقة المقترحة طبقت طريقة ضغط قبل تطبيق خوارزمية التشفيرعلى الفيديو بواسطة استخدام(DWT) لتسهيل عمل خوارزمية التشفيرلأن الضغط يقلل حجم الفيديووهذا يجعل وقت التنفيذ لخوارزمية التشفيرأسرع ،خوارزمية فك التشفيروطريقة فك الضغط هم أكبر اثبات لنجاح الطريقة المقترحة التي تعطي الفيديوالمطلوب بدون اي تشويه، وبالاضافة الى ذلك تم التحقق من اداء الطريقة المقترحة بواسطة استخدام العديد من طرق قياس معايير الجودة.

1. Introduction

The multimedia data such as images and videos become common and heavily used in our lives and in many areas such as surveillance systems and phones cameras, these data contain important and accurate information, which are vulnerable to attacks, whether from inside or outside the system. Cryptography is science that converts data such as (text, audio, image, video) from understandable to non-understandable to keeping data secure from unauthorized attackers and this done by using encryption algorithm. The video encryption is used against attackers during transmission video and storage, while retrieving the original video the decryption algorithms is applied [1].

The compression techniques are important corresponding to the core of multimedia data to stored and transferred efficiently. Since digital video contain a large amount of redundant data the compression reducing the quantity of data by removes unnecessary data from file (video) without make change in the quality of the original data, and this will reduce storage size, and reduce bandwidth to transfer data in faster away with save time [2].

Steganography is the science of hiding data such as text inside images or video for security by using Least Significant Bit (LSB) technique that hidden data inside lower bits of the pixels of the image [3]. In this paper the steganography used for hiding text inside frames of video, this text refers to the signature of the sender that added into frames for authentication purpose because the receiver known identification of sender from this signature.

The compression, encryption, and signature techniques are applied in the proposed method to protect video during storage it or during send it to final receiver.

In this paper Section 2 related works, Section 3 the proposed method which contains compression method, encryption algorithm by using three different keys, signature the video, the signature verification, decryption algorithm by using the same keys that used in encryption algorithm because the keys are symmetric, decompression method, Section 4 describes experimental result, Section 5 evaluation of the proposed system, Section 6 describes the conclusion.

2. Related Works:

Al-Ani M S, Hammouri T A, 2011, [4], they applied Discrete Wavelet Transform (DWT) to compressed the video during the transfer or download operations over the network. The obtained results reduce both the video size and transfer time. DWT is an efficient method that can be used to perform an efficient compression technique. The video encryption also applies to obtain high level security.

A. Swathi, S.A.K Jilani, 2012, [5], they apply steganography technique for hiding data inside the video. The Least Significant bit (LSB) method was used, which is considering a good technique for hiding all letters of the message with frames of the video. These frames selected in random away.

Kulkarni A, Kulkarni S, et al, 2013, [6],was present proposed algorithm for encryption of compressed video to safely exchange highly confidential videos and by this proposed method could prevent unauthorized viewing of the video file. These make a balance between security and computational time by applying compression method on video before encryption because the compression method reduces storage space and save bandwidth. The authors not used steganography, so to increase the security of video can use steganography for hiding data within the frames of encryption video.

Jaspreet Kaur , Jagroop kaur, 2016,[7], the video steganography is a very important task in real life where the users want to keep data secret. The problem occurs when traditional text and image based steganography techniques are not plentiful .They are able to carry only small files. So there is a problem, how to get much enough files to hide our message. This becomes a very tedious task for carry large amount of data. Here, comes the need of video steganography. The use of video as a carrier cover for the secure message is overcame the capacity problem. Information can be hidden in any frame of video. That means the video has a large capacity to store information.

3. Proposed method

3.1 Extraction video into frames phase

The proposed method is extraction video into frames then applied compression, encryption, and steganography on each frame respectively. The proposed method implemented by using C# program, the aim of this paper is representing an efficient method to provide multimedia (videos) with high level of security and protects it from attacks. Figure-1(a) shows a flow chart of sending process. Figure-1 (b) shows a flow chart of receiving process.

The proposed method used camera connected into a base station (personal computer). This camera record video that consist from series of frames 25 frame per second and transfer into the base station to apply set processes on video. These process are the compression method, encryption algorithm, and video signature, the video extraction into frames to implementation this processes on each frame, algorithm (1) explains extraction video into frames.

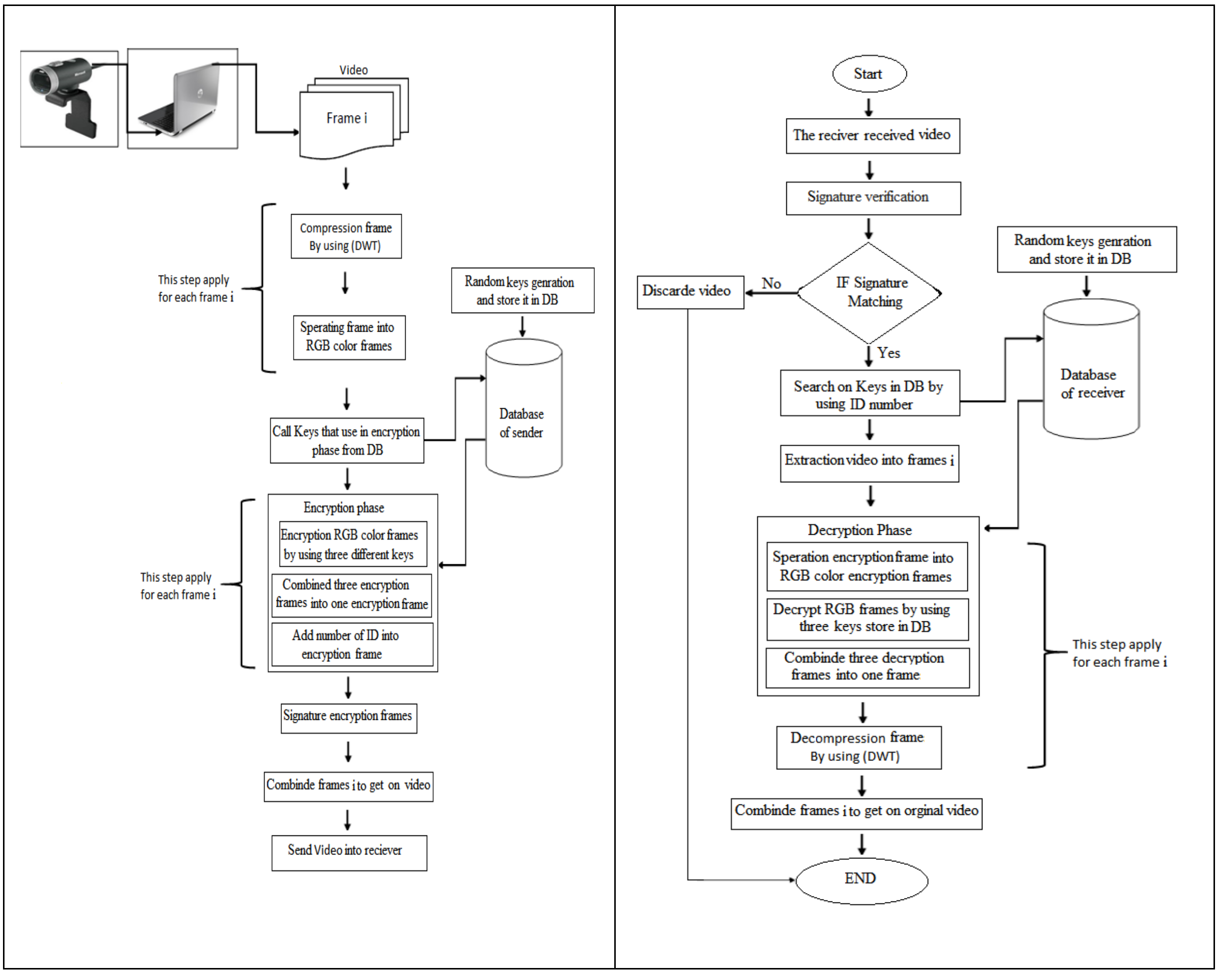

Video
Frame i
Compression frame By using (DWT)
This step apply for each frame i
Sperating frame into RGB color frames
Random keys genration and store it in DB
Database of sender
Call Keys that use in encryption phase from DB
Encryption phase
Encryption RGB color frames by using three different keys
Combined three encryption frames into one encryption frame
Add number of ID into encryption frame
This step apply for each frame i
Signature encryption frames
Combinde frames i to get on video
Send Video into reciever
Start
The reciver received video
Signature verification
IF Signature Matching
No
Yes
Discarde video
Random keys genration and store it in DB
Database of receiver
Search on Keys in DB by using ID number
Extraction video into frames i
Decryption Phase
Speration encryption frame into RGB color encryption frames
Decrypt RGB frames by using three keys store in DB
Combinde three decryption frames into one frame
This step apply for each frame i
Decompression frame By using (DWT)
Combinde frames i to get on orginal video
END

Figure 1-a) Flow chart of sending processes, **b)** Flow chart of receiving processes.

Algorithm1: Extraction video into frames.

Input: Digital video.
Output: Frames of video.

Step1: Start.
Step2: Read video, file name= name of video file and path= location of video file.
Step3: Extract frames from video and store in array of one dimension is [N] // N=number of frame.
Step4: End.

3.2 Compression video phase

The compression phase is very important to reduce the size of video, since the uncompressed videos require a huge space in memory and take huge time to upload or download videos from the internet or during transfer into the receiver. So the video size always effects on the efficiency of video transmission over network, and on storage space on PC hard disk [8].

The video having large areas of redundancies (unnecessary information's).The objective of video compression is to reduce redundancy of the video data in order to be able to store or transmit data in an efficient form. Data compression is achieved when one or more of these redundancies are reduced or eliminated. In lossy compression reduces a file by permanently eliminating certain information, especially redundant information. When the file is uncompressed, only a part of the original information is still there. The pictures and videos from digital cameras are examples of digital files that are commonly compressed using lossy methods. The Haar wavelet transform with one iteration can be used to perform lossy compression with retains the quality of the compressed video.The Haar algorithm used to compressed the video because it popular and simplest transform for data compression and high computation speed [9]. The frame (image) is generated by low pass filter (LPF), while the detailed frame are produced using a high pass filter (HPF), the compression process by using Haar wavelet algorithm divided the frame into a series of the frequency bands the low low (LL) band, the low high(LH) band, the high low(HL) band, and the high high(HH) band that shows in Figure -2, [10]. The results as compressed frame from compression phase give to encryption phase. These processes apply on all frames of video.

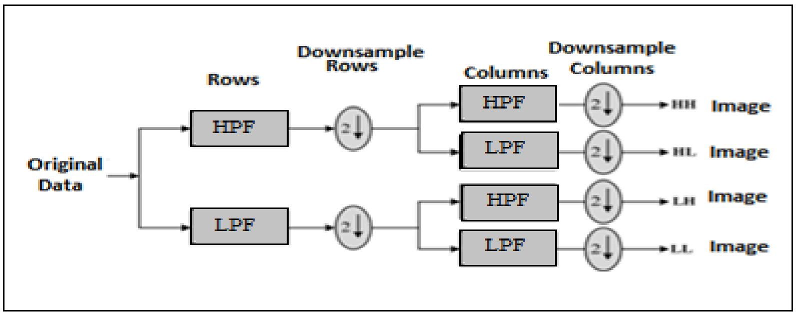


Figure 2- One Stage of Forward DWT [10].

3.3 Keys generation phase

The proposed method generation random keys by using random numbers function and store these keys as a table in the database (DB), only authentication personal that has a username and password can access into this table to insert , delete , update and use its data. Each key different on another one by its content, the length of each key is an array of 16 numbers (1-3 digits) between (0,256). Algorithm (2) explains the generating random keys and store these keys in a Table -1 .The size of the Table-1 is variable and consist of rows and columns, each row of identification (ID) have three keys used in encryption image, and generating new keys to new encryption process image, Table -1 shows some keys.

Table 1-The content of random keys table.

ID	Key1	Key2	Key3
1	249,125,46,210,150,55,42,12, 90,119,135,166,68,108,200,19	146,73,27,123,88,32,24,7,52,69, 79,97,40,63,117,11	185,93,34,156,111,40,31,9,66, 88,100,123,50,80,148,14
2	21,27,72,185,129,67,88,131, 124,221,10,209,125,150,2,239	12,16,42,108,75,39,51,77,72,130, 5,122,73,88,1,140	15,20,53,137,96,50,65,97,92, 164,7,155,92,111,1,178
3	92,226,123,41,219,192,27,124, 138,112,244,10,151,181,181,14	54,132,72,24,128,113,16,73,81,66, 143,6,88,106,106,86	68,168,92,30,163,143,20,92, 103,83,181,8,112,135,134,109
4	223,23,172,192,198,65,35,38, 220,22,19,32,131,11,11,94	130,13,101,112,116,38,21,22,129, 13,11,18,76,6,6,55	165,17,128,143,146,48,26,28, 163,16,14,23,97,8,8,70
5	41,147,222,123,153,234,161,11, 35,181,60,171,255,206,208,156	24,86,130,72,90,137,94,69,21,106, 35,100,149,121,122,91	30,109,165,91,114,173,119,87, 26,134,44,127,189,153,154,115

```
Algorithm2: Generating random keys
Input: Random  numbers between (0,255)
Output: Random keys
Step1: Start.
Step2: Use a function to generate a random key consists from array of 16
          Numbers (3 digits length) between (0,255).
          For i=0 to 16
             K1 [i] = random. Next(0, 255)
Step 3: Next i
Step 4: To generate K2,K3 Repeat steps (2 to 3)
Step 5: End
```

3.4 Encryption frames phase

After generation keys to encrypt frames of video, now to apply encryption phase must extraction frame of video into three frames are Red (R), Green (G), Blue (B), to encrypting compressed frames with three different random keys generated from keys generation phase then used addition operation between all pixels of frames and keys as defined in equation (1).

$$E = (P+K) \text{ Mod } 256 \ldots \qquad \ldots (1)$$

Where + refers to addition operation that means (addition values of pixels (P) from the frame with values of the key (K) mod 256 because the range of values of the pixels of frame and keys between 0 and 256 and store result in E) as shows in algorithm (3), the reason to use addition operation is fast in calculations and uncomplicated and give good result in encryption process.

```
Algorithm3: Encryption RGB frames of video
Input: (Red, Green, Blue) frames with K1, K2, K3.
Output: Encryption (Red, Green, Blue) frames.
Step1: Start.
Step2: Read frames of video by using for loop
        For i=0 to N
        Calculate height(x) and width (y) of Red frame and read each pixel in
        Red frame according to location of pixels(x, y).
          For x=0 to height
              For y=0 to width
                P= read pixel of Red frame [x, y]
Step3: Do addition operation between k1 and pixel [x, y] of Red frame and repeat
         use the k1 with pixels of Red frame and according this condition.
           If (i = = 16) then   i=0
           // addition operation with mod 256 to save value of pixels inside 256
                EncryRed = (P +K1 [i]) % 256;  \\P is pixel
                Draw the pixel of EncryRed as new image
          Next i
Step4: Next y
Step5: Next x
Step6: To encrypt Green and Blue frames by( K2 ,K3) Repeat steps (2 to 5)
Step7: Repeat steps (2 to 6) until encrypt all frames of video.
Step8: End
```

3.5 Hidden ID number phase

In this phase combine the encrypted RGB frames into one frame, the size of each frame is (128×128) to hide ID number that refer into number of row from Table -1 that store keys in it. This key use in decryption phase to enable a receiver to decrypt the encrypted frames. The sender in this phase takes resulting from the algorithm (3) and read first pixel in first frame from frames of video and hide ID number in this pixel. Algorithm (4) shows this.

```
Algorithm4:Hidden ID number inside encryption first frame
Input: Encryption frame and  ID number from Table-1 that have encryption Keys
Output: Hidden ID number inside encryption first frame
Step1: Start.
Step2: Calculate height(x) and width (y) of encryption frame and read each pixel in this frame
         according to location of pixels(x, y).
          For x=0 to height
            For y=0 to width
              P= read pixel of encryption frame [x, y]
             //input ID number if location of pixel in encryption frame is [0, 0]
                If (x < 1 && y < 1) then
                    P= ID number
                      Draw (P) pixel of encryption frame as new frame
                Else
                      Draw another pixels of encryption frame as new frame
Step3: Next y
Step4: Next x
Step5: End
```

3.6 Signature video phase

The attack on video is very popular in the last days, to protect video by encrypting its frames not enough because the active attacker can input frames inside frames of video and by this can change data of video. The sender could add his signature on all frames of video for authentication, algorithm (5) explains added the signature of sender by hiding it inside frames of the video by using steganography LSB technique. It takes the first bit from pixels of frame to hide a bit of signature inside it, this process repeat unlit hide signature inside all frames of video.

```
Algorithm5:Hidden signature inside frames of video
Input: Encryption frames
Output: Signature encryption frames
Step1: Start.
Step2: Read letters(signature) from the textbox
Step3: Find ASCII code of letters and convert into binary number and Store  in array called bin
Step4: Read frames of video by using for loop
       For i=0 to N
       And calculate height(x) and width (y) of encryption frame and read each pixel in this image
       according to location of pixels(x, y).
         For x=0 to height
           For y=0 to width
               P= read pixel of encryption frame [x, y]
                  if(x= = 0&y= = 0)
                   Draw p[x][y] as new image //because content on ID number
                  Else
                    if(n<size of bin[n])
                     Convert  p (pixel from frame)into binary number and store in array called binimage
                     and apply steps from(step5 to step7)
                    Else
                    Go to step8
Step5: Read bits by using for loop from array (bin)  of letters and hidden it inside
       first bit from byte in array of p (binimage)
       for (k = 0; k < 7; k++)
         if(K= =0)
          binimage [k] =bin[n]//k counter of binimage array, n counter of bin array.
          Increase n// to read next binary number store in bin array.
         Else
          binimage[k]= binimage[k]// Remain the same values
Step6:  if(k=7) then  convert values inside binimage that represent one byte into
       decimal number and draw as new frame
       Draw p[x][y] as new frame
Step7: K=0
       binimage="" // To store new values in binimage
       Repeat from step4 to step7 until hidden all letters with pixels of frame
Step8: Draw another values of frame as new frame// Remain the same values of frame23
       if (n >= size of bin[n])
       Draw p[x][y] as new frame //without apply any process on it
Step9: Next y
Step10: Next x
Step11: Repeat steps (4 to 10) until signature all frames of video.
Step12:End
```

3.7 Convert frames into video phase

After getting on compressed encrypted frames with hidden signature of the sender within pixels of frames the frames of video combined in video file to get the encrypted video and send it to the receiver.

3.8 Signature verification phase

When video access into the receiver the first phase is signature verification of each frame from frames of video, if matching with the signature of sender that store in receiver then decrypt data in inverse method to encryption method to get on target frame, if not then discarded data and exit from system, algorithm (6) explains this.

```
Algorithm6:Signature verification
Input: Signed encrypted frames
Output: Signature matching or not
Step1: Start.
Step2: Receiver store signature (letters) of sender in array called sig for verification.
Step3: Read frames of video by using for loop.
        For i=0 to N
        and calculate height(x) and width (y) of encrypted frame and read each
        Pixel in this frame according to location of pixels(x, y).
          For x=0 to height
            For y=0 to width
              P= read pixel of encrypted frame [x, y]
                if(x= = 0&y= = 0)
                  Draw p[x][y] as new frame //because content on ID number
                Else
                  Convert  p (pixel from frame)into binary number and store in array called binimage
Step4: Read first bit from binimage array and store in bin array
        for (k = 0; k < 7; k++)
         if(K= =0)
            bin[n]= binimage[K]
            Increase n// increase n until content on 8 bi
         Else
             binimage[k]= binimage[k]// Remain the same values
         K=0
         binimage="" // To store new values in binimage array
Step5:Repeate steps (step3 and step4) while n not equal into 7
       If(n = 7)
        Convert values(binary numbers) that store in bin array into decimal number that
        represent assci code from it get on letter and store in string such as S
        bin= ""  // To store new values in bin array
        n=0  // n represent counter of bin array
Step6: Signature verification
         if(S= sig[i])
           Textbox=sig[i]
            i++
            S=""   // To store  new letter in S because the algorithm matching one letter in each
            iteration.
              if( i = size of sig[i])
                ("The Signature Matching ") // that mean the sender authentication.
         Else
           ("The Signature not Matching ")
           Textbox= ""  // To display new signature
           Break // Stop the system
Step7: Next y
Step8: Next x
Step9: Repeat steps (3 to 8) until signature verification of all frames of video.
Step10:End
```

3.9 Decryption phase

When the receiver verifying the identification of the sender, the next step is extraction video into frames to decrypt the frames of video by using the number hidden within first pixels of first frame.

This number refer to ID number row that found in table that have three keys used in encryption process and only authenticate person could access to it. The same this key use in decryption process because the keys are symmetric that mean the same keys used in encryption use in decryption process, the decryption stage divided into two phases as follows:

- Find hidden ID number within the first encryption frame as shown in the algorithm (7), then extract each encryption frame into (Red, Green, Blue) frames.
- The decryption (Red, Green, and Blue) frames process is explained in algorithm (8) as shown in equations 2 and 3.

$$M = P + 256 \qquad \text{…………..} \quad (2)$$

$$D = (M - K) \text{ Mod } 256 \qquad \text{…….. ….} \quad (3)$$

Where M is variable to store result of inverse mod, that mean (subtraction values M from values of key (K) mod 256 because the range of values of frame and keys between 0 and 256 and store result in D) and after this combined red frame, green frame, and blue frame together to get color frame as input to algorithm (8).

Alogorithm7: Find hidden ID number within first encrypted frame

Input: First encrypted frame
Output: Find hidden ID number

```
Step 1: Start.
Step 2: Read frames of video if the frame is first frame then read pixel of first encrypted frame in
        location [0, 0] and find ID number that has three keys that use to decryption process.
Step 3: Search on keys and find it in table of receiver because the sender and receiver has the
        same tables.
Step4: End
```

Algorithm8: Decryption(Red, Green, Blue)frames

Input: Encrypted frame and three Keys
Output: Decryption (Red, Green, Blue)frames

```
Step 1: Start.
Step 2: Read Red frames of video by using for loop
        For i=0 to N
        For x=0 to height // to read pixels of encrypted Red frame
            For y=0 to width
              P= read pixel of EncryRed frame [x, y]
Step 3: Do subtraction operation between k1and pixel [x, y] of EncryRed frame and repeat use the k1
        with pixels of EncryRed frame and according this condition.
          If (i = = 16) then i=0
            // Inverse of mod operation
              M = P + 256;   // M is variable to store result of inverse mod
            // subtraction operation is invers of addition operation
              DecryRed   = (Math.Abs (M– K1 [i]) % 256);
              Draw the pixel of DecryRed in new frame.
         Next i
Step 4: Next y
Step 5: Next x
Step 6: Repeat steps (2 to 5) until decryption all Red frames of video.
Step 7: To decrypt Green and Blue frames by( K2 ,K3) Repeat steps (2 to 6)
Step 8: To  get one decryption frame combined (Red,Green,Blue) frames in one frame and to
        decryption this frames repeat decryption  process on all frames of video by using for loop.
Step 9: End
```

3.10 Decompression phase:

Haar Wavelet Transform has feature is symmetric that mean both the forward and the inverse transform has the same complexity [11], that means decompression phase is same of compression phase but in inverse away and the result is decompression frame, Figure-3 shows this. After decompression frames of video, now combined in the same way that mentioned earlier to get on target video.

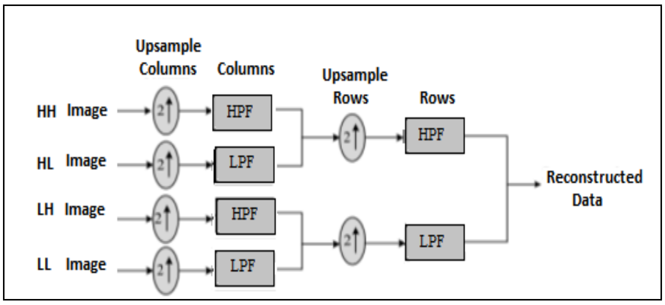


Figure 3- One Stage of Inverse DWT [10].

4. Experimental result

Example of the video to tourist resort applied to discuss the results.

Figure 4- Video to tourist resort.

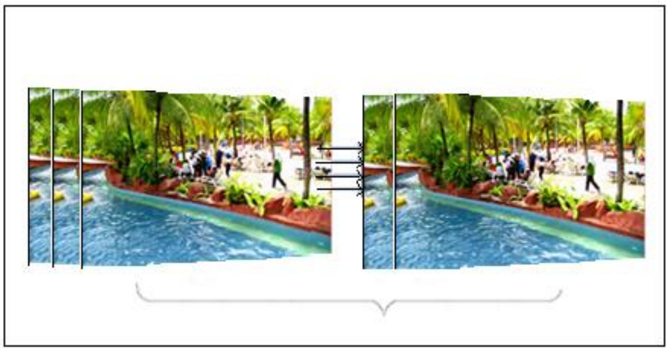


Figure 5- Input video **Figure 6-** Extraction video into frames (25 fames)

Figure 7- Resize frames into smaller size

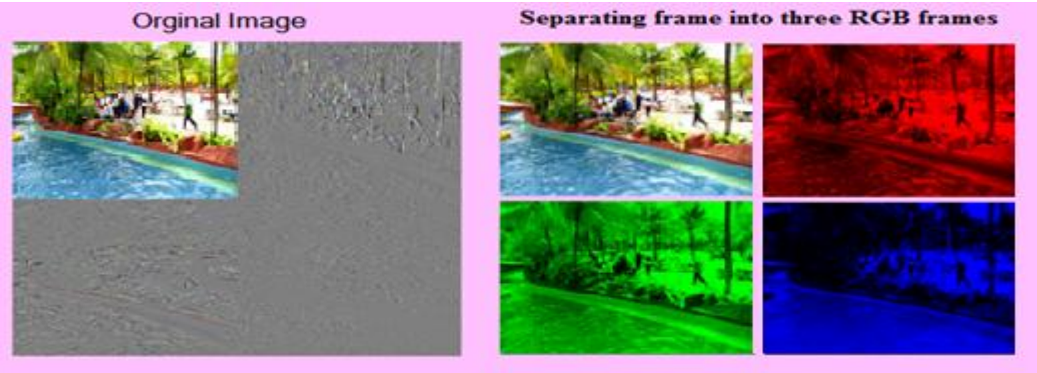
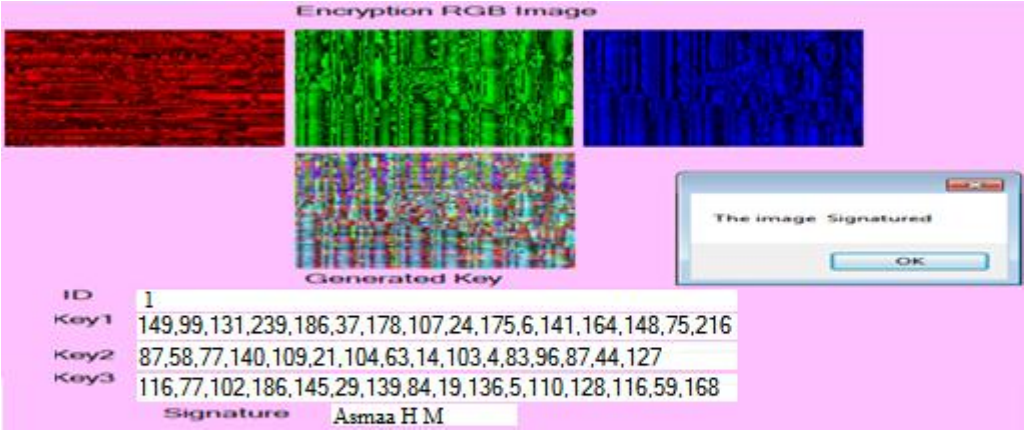


Figure 8- Compression frames and separating each frame into RGB frames



The image Signatured

ID 1

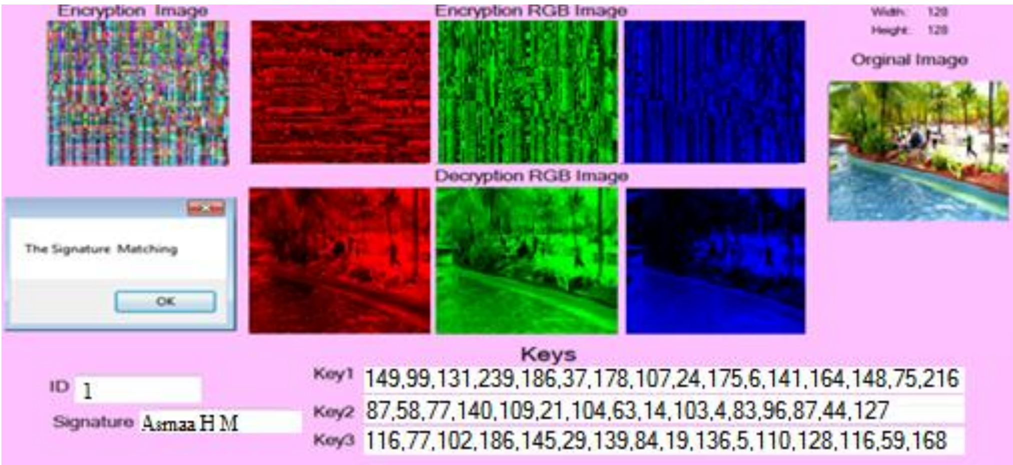
Key1 149,99,131,239,186,37,178,107,24,175,6,141,164,148,75,216

Key2 87,58,77,140,109,21,104,63,14,103,4,83,96,87,44,127

Key3 116,77,102,186,145,29,139,84,19,136,5,110,128,116,59,168

Signature Asmaa H M

Figure 9- Encryption frames and combined into one frame and hide (ID, signature) within it.



Width: 128
Height: 128

The Signature Matching

ID 1

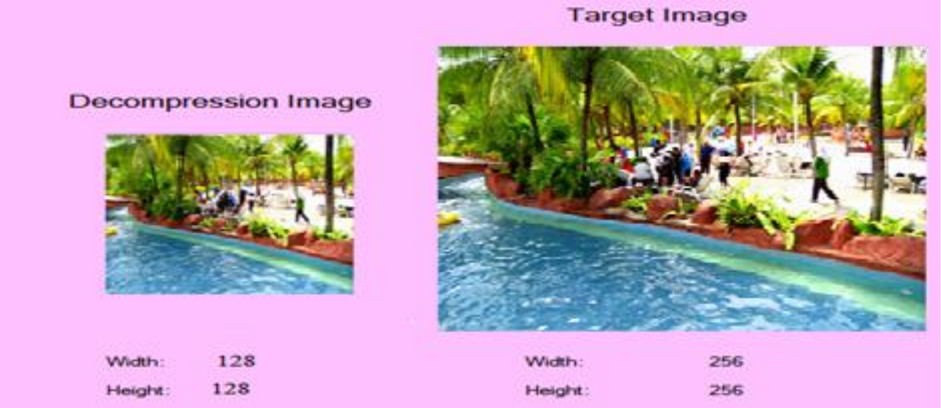
Signature Asmaa H M

Key1 149,99,131,239,186,37,178,107,24,175,6,141,164,148,75,216

Key2 87,58,77,140,109,21,104,63,14,103,4,83,96,87,44,127

Key3 116,77,102,186,145,29,139,84,19,136,5,110,128,116,59,168

Figure 10- Decryption (RGB) frames after signature verification and find ID and combined decryption frames into one color (RGB) frame.



Width: 128
Height: 128

Width: 256
Height: 256

Figure 11- Decompression phase

5. Evaluation of the proposed system

To evolution the quality performance of the proposed system, two quality measurements were used; MSE and PSNR; where PSNR analyzes the quality of compression frame and the encryption frame with original frame[12][13].

$$PSNR = 10\log_{10}\frac{(peakval)^2}{MES} \quad \ldots \quad \ldots (4)$$

Where *peakval* is either specified by the user or taken from the range of the frame data type and this rang is (from 0 to 255) and MES used to measure determination the difference between the original frame and the compression or encryption frame I', If the frame has a size of M * N then

$$MES = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - I'(i,j)]^2 \quad \ldots \quad \ldots (5)$$

The results PSNR, MSE, and the processing time in proposed system applied on two frames from frames of video are the first frame and final frame as examples, because to find final result of the video must calculate results of PSNR, MSE of all frames of video and this take time and effort, Table-2 shows results of one frame and Table-3 shows results of final frame, this results prove success of proposed method compere with another results from another papers [14][15], with take into consideration the time engrossed in to execution various operations is in nanosecond that means operations take few time in the execution. The data during transfer from sender into receiver will be exposed to a lot of attacks, so protect this data by encrypt it with strong key only the administrator person for the system can know these keys, in each encryption process the user generation new keys to prevent the attack from detect these keys, the keys generated outside the proposed system and store in table of DB, so not effect on performance of the proposed system, and also add signature on encryption data for authorization add strength to support the proposed method because this sign enable receiver from known sender identification.

Table 2- Result MES, PSNR, and processing time of first frame from tourist resort video.

		Compression	Encryption
First frame from video	MES	2.12703258	6.3819137396163
	PSNR	37.5912013719810	34.041805914809
	Processing time	2641843500 Nano sec.	248431701 Nano sec.

Table3- Result MES, PSNR, and processing time of final frame from tourist resort video .

		Compression	Encryption
Final frame from video	MES	2.7915281105665	70.1619340129
	PSNR	33.7522106861580	24.0383289728460
	Processing time	607418291 Nano sec.	73946001 Nano sec.

6. Conclusion

The attackers on videos increase now days, in this paper applied many processors on video to protect it from attacker that which can be exposed during transmission video and storage. These processors are (compression, encryption, and steganography). The video encrypt by applying an encryption symmetric algorithm on video after applying compression method for it, since the compression reduce the size of the video into a smaller size and this facilitation work of encryption algorithm. The steganography used Least Significant Bits (LSB) technique that hidden signature inside lower bits of pixels of the image to enable the receiver from sure of the identity of the sender and provide secure communication between two communication parties. To retrieve original video after signature verification applied decryption algorithm and decompression method for video respectively. Simulation results that done by using MSE and PSNR shows this algorithm reduces the time that requires to encryption video with high security and confidentiality and enhanced level of security of video and good encryption algorithm effect.

References

1. Abomhara, M,, Zakaria, O. and Khalifa, O. O. **2010**. An Overview of Video Encryption Techniques. *International Journal of Computer Theory and Engineering*, 2(1), pp: 103-110.
2. Saroya, N. and Kaur, P. **2014**. Analysis of Image Compression Algorithm Using DCT and DWT Transforms. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(2), 897-900.
3. Mistry, D., Desai, R.and Jagad, M. **2015**. Hidden Data Transmission Using Image Steganography. *International Journal of Computer Applications*, 130(14), pp: 8-11.
4. Al-Ani, M. S.and Hammouri, T. A. **2011.** Video Compression Algorithm Based on Frame Difference Approaches. *International Journal on Soft Computing (IJSC)*, 2(4), pp: 67-79.
5. Swathi, A. and Jilani, S. A. K. **2012.** Video Steganography by LSB Substitution Using Different Polynomial Equation. *International Journal of Computational Engineering Research (ijceronline.com)*, 2(5), pp: 1620-1623.
6. Kulkarni, A., Kulkarn, S., Haridas, K. and More, A. **2012**. Proposed Video Encryption Algorithm v/s Other Existing Algorithms : A Comparative Study. *International Journal of Computer Applications*, 65(1), pp: 1-5.

7. Kaur, Jas and Kaur Jag .**2016**. Hiding Text in Video Using Steganographic Technique - A Review. *An International Journal of Engineering Sciences*, 17(1), pp: 578-582.

8. Kaur J., Sharma R. **2012** . A Combined DWT - DCT approach to perform Video compression base of Frame Redundancy. *International Journal of Advanced Research in Computer Science and Software Engineering,* 2(9), pp: 325-332.

9. Gupta D., Choubey S. **2015.** Discrete Wavelet Transform for Image Processing. *International Journal of Emerging Technology and Advanced Engineering*, 4(3), pp: 598-602.

10. Rathee M., Vij A. **2014.** Image compression using discrete Haar wavelet transforms. *International Journal of Engineering and Innovative Technology (IJEIT),* 3(12), pp: 47-51.

11. Gupta R. **2014.** Image compression using Haar wavelet transform and chaos-based encryption. *IJCSI International Journal of Computer Science Issues,* 11(1), *Issue* 2, pp: 174-180.

12. Laskar S. A. and Hemachandran K. **2012.** High capacity data hiding using LSB steganography and encryption. *International Journal of Database Management Systems (IJDMS)*,4(6), pp: 57-68.

13. Reddy V. R, and Reddy T. S. **2014.** Image encryption using fractional random wavelet transform. *International Journal of Advanced Research in Computer and Communication Engineering* ,3(1), pp: 4891-4893.

14. Poobathy D. and Chezian R. M. **2014.** Edge Detection Operators : Peak Signal to Noise Ratio Based Comparison. *I.J. Image, Graphics and Signal Processing*, 7(10), pp: 55-61.

15. Al – Ani S . A . H. **2007.** Steganography Image in Image using Modified method in Least Significant Bit (LSB) substitution. *Um-Salama Science Journal,* 4(1), pp: 133-141.