

Abstract

There is a modern type of encryption called selective encryption which depends on the choice of some sensitive areas. The encryption of these areas leads to destroy the whole file. This type of encryption is more compatible with compressed files such as Moving Picture Expert Group (MPEG) video files and Joint Picture Expert Group (JPEG). This compatibility produced from a large amount of information focused in a small area more than non-compressed files.

The (MPEG-1) video file contains three types of pictures reference (I), predicted (P) and bidirectional (B) pictures. The most important one among these three types is (I) picture, which is the basis that other types (P and B) were derived from, P pictures are predicted from (I) only in one direction (Forward prediction) and (B) pictures were predicted from (I) and (P) in the two directions (Forward/ Backward prediction), therefore the encryption of (I) pictures only lead to destroy most of visual information (encryption in the first degree of complexity), while the encryption of (I) and (P) pictures together lead to destroy all visual information (encryption in the second degree of complexity) with no need to encrypt (B) pictures. It is important to mention that the total number of (I and P) pictures within the MPEG-1 file is less than the total number of (B) pictures only, leaving a large amount of information without encryption and reduced the time needed for implementation.

In the proposed encryption system, three encryption methods were used Fibonacci, Galois and the Proposed method (Random Seed Values). The proposed method was developed to compete other methods with a smaller probability of breaking encrypted files. The execution time of Galois method is faster than other methods, The encryption system was implemented using Visual Basic 6.0 programming language. The fidelity measures (MSE) and (PSNR) are used to check the result of the whole developed techniques.

ACKNOWLEDGMENTS

Before anything ...

Thanks to Allah for helping me to complete my thesis

I wish to express my deep appreciation and sincere thanks to my supervisor Dr. Ban N. Al-Kallak for her appreciable advices, important comments support and encouragement during the research.

Special and sincere thanks to Dr Loaaay E. George for his important advices in my thesis.

It is of my pleasure to record my independences to the staff members of Computer Science, whom gave me all facilities during my work,

Thanks are extended to the College of Science of Al-Nahrain University for giving me the chance to complete my postgraduate study. Also, I would like to express my deepest thanks to all my friends, and every one who helped me in my project.

Finally, I wish to thank all of my friends Bahaa, Jaber, Dr.Kahlan, Mohammed, Salam, Dhiah, Osama, Akram, Omar, Ahmed, Safaa, Anmar, Hussam, Abo arab, Zaid, Ali, Haider, Hadi, Nahel and Saad for them help.

Muhammed

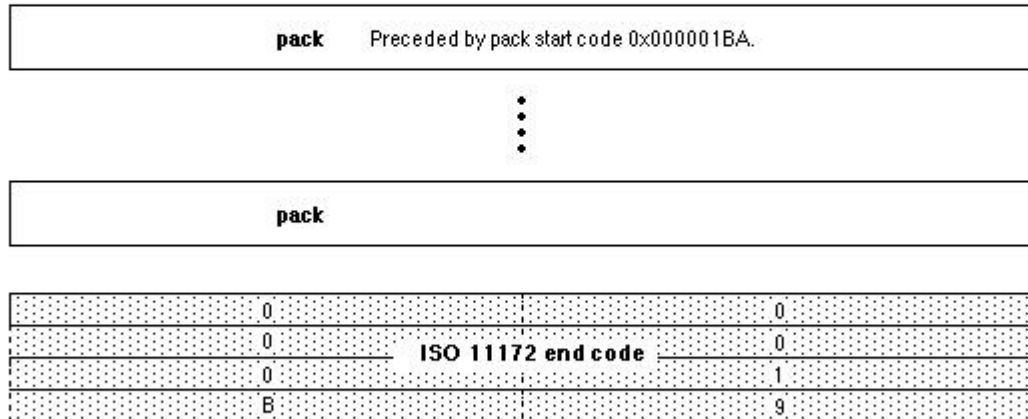
May 2008 

MPEG-1 Data Structures

The ISO/IEC 11172 specification defines the audio, video and multiplexing standards collectively and colloquially referred to as the MPEG-1 (Motion Picture Experts Group) compression standard. More practically, in order to parse an MPEG-1 bitstream, it is necessary to know byte offsets within each structure. To make this information more readily accessible, graphic forms are condensed. A multiplexed MPEG-1 stream is composed of distinct Packs. Each Pack consists of a Pack header and any number of Packets. Within those Packets is either video or audio data. These structures above the video or audio level are called the system layer. Video or audio data is divided into Packets without regard to lower-level structures.

MPEG-1 file will be as follow...

MPEG-1 Multiplexed Stream



MPEG-1 Pack

byte	bit 7	6	5	4	3	2	1	0
0	0			0				
1	0			pack start code			0	
2	0			1				
3	B			A				
4	always set to 0010				system clock reference bits 32..30			marker always set
5	SCR bits 29..15 Intended time, in 90 kHz clock cycles, of arrival of byte 8 of this header.							
6	(cont.)							marker always set
7	SCR bits 14..0							
8	(cont.)							marker always set
9	marker always set	multiplex rate (This #) # 400 bits/sec is the rate at which this stream ...						
10	... is to be delivered to the decoder.							
11	(cont.)							marker always set

(Optional)

system header	Preceded by system header start code 0x000001BB.
----------------------	--

packets	Arbitrary number, each preceded by packet start code prefix 0x000001.
----------------	---

MPEG-1 System Header

byte	bit 7	6	5	4	3	2	1	0	
0	0						0		
1	0		system header start code				0		
2	0						1		
3	B						B		
4	header length				Bytes in header, from byte 6.				
5	(cont.)								
6	marker always set	rate bound			Max value of the multiplex rates of all packs in the stream.				
7	(cont.)								
8	(cont.)							marker always set	
9	audio bound					Max number of audio streams in this ISO stream.		fixed flag Set for fixed bitrate.	CSPS flag Set for constrained.
10	system audio lock flag †	system video lock flag †	marker always set	video bound					Max number of video streams in this ISO stream.
11	reserved								Currently should be set to FF.

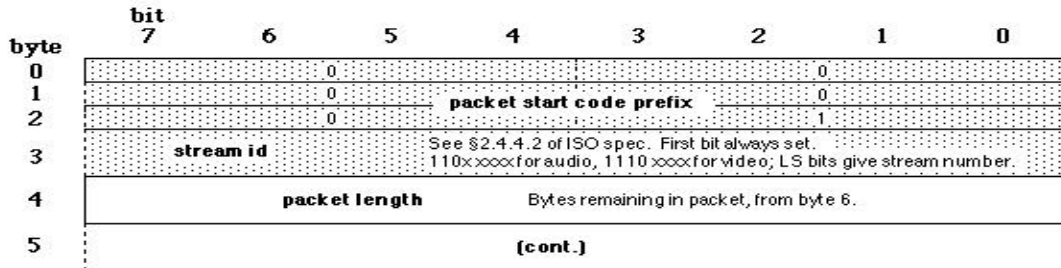
† Set if there is, for all (audio/video) elementary streams in the ISO multiplexed stream, a constant rational relationship between the (audio sampling rate/video picture rate) and the system clock frequency in the decoder. See §2.4.4.2.

The following stream specs are present only if the first bit is a 1. Any number of stream specs may follow.

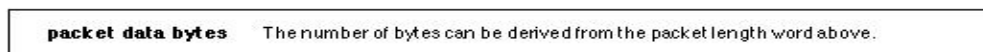
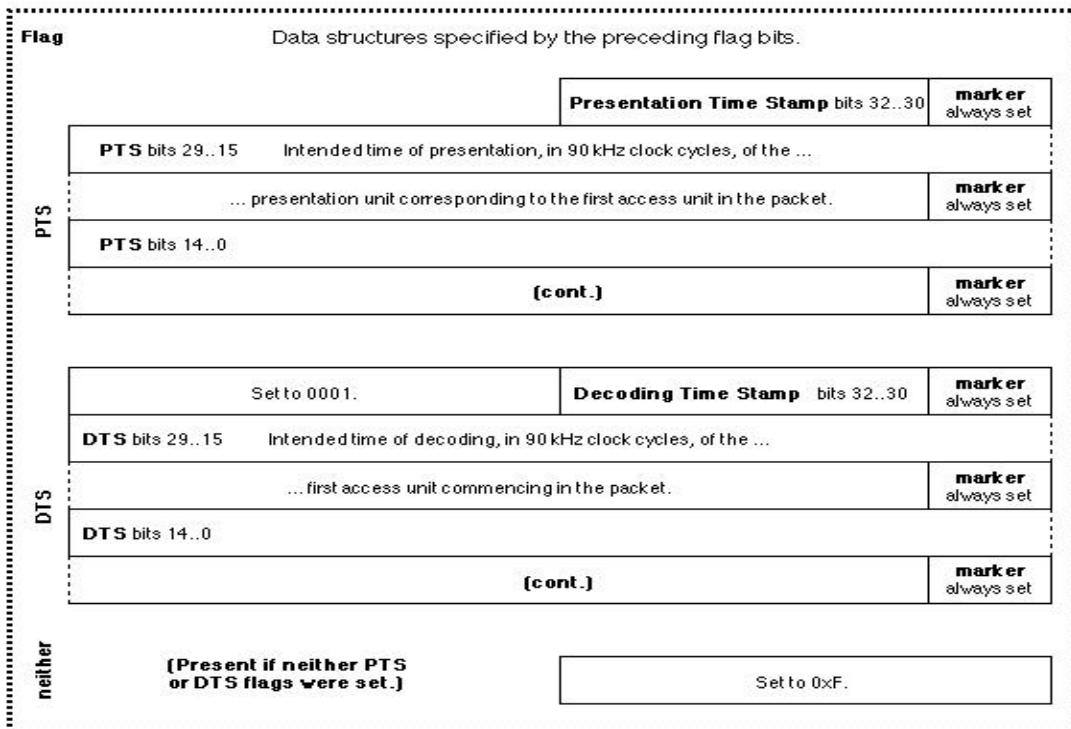
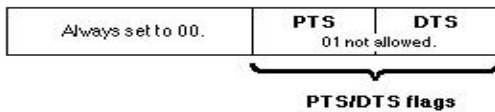
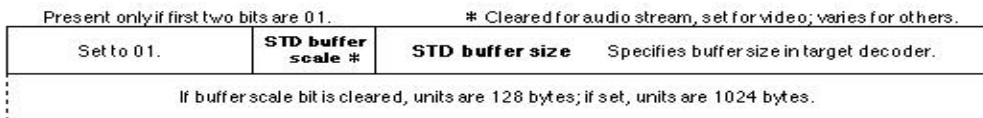
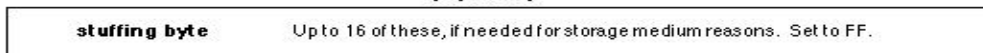
12	stream id		See §2.4.4.2 of ISO spec. First bit always set. 110xxxx for audio, 1110xxxx for video; LS bits give stream number.						
13	Always set to 11.	STD buffer bound scale #	STD buffer size bound						Largest required buffer over all ...
14	... packets in this stream. If bound scale bit is cleared, units are 128 bytes; if set, 1024 bytes.								

Cleared for audio stream, set for video; varies for others.

MPEG-1 Packet

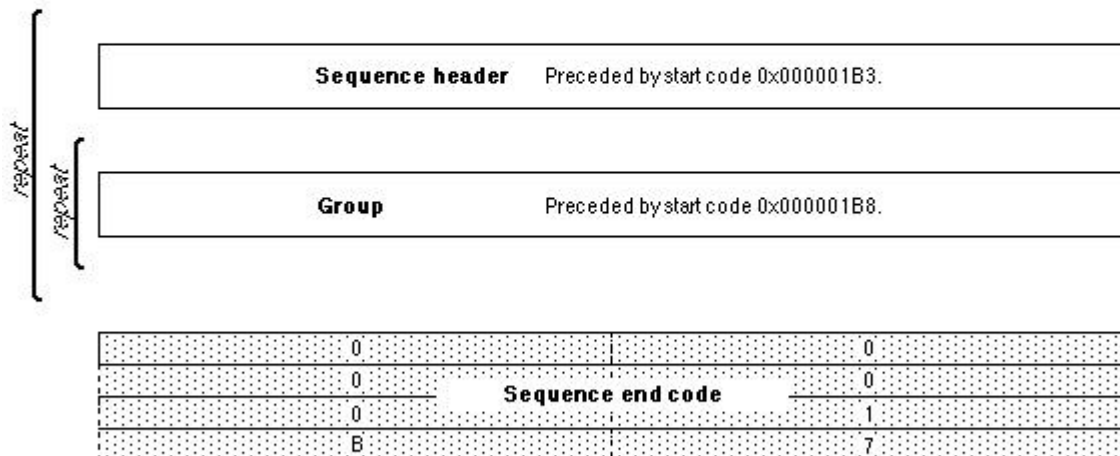


(Optional)



MPEG-1 Video Sequence

(Skip to next start code)



MPEG-1 Sequence Header

byte	bit 7	6	5	4	3	2	1	0
0	0			0			0	
1	0			Sequence header code			0	
2	0			1			1	
3	B			3			3	
4	horizontal size Width in pixels of the displayable luminance picture. The encoded width...							
5	... in macroblocks is (displayable + 15) div 16.				vertical size			
6	(cont.)							
7	pixel aspect ratio Respectively: forbidden, 1.0, 0.6735, 0.7031, 0.7615, 0.8055, 0.8437, 0.8935, 0.9157, 0.9815, 1.0255, 1.0695, 1.095, 1.1575, 1.2051, rsvd.				picture rate Respectively: forbidden, 23.976, 24, 25, 29.97, 30, 50, 59.94, 60, rsvd ... rsvd.			
8	bit rate Stream bitrate in units of 400 bits/s. Zero is forbidden; 3FFFF means variable bitrate.							
9	(cont.)							
10	(cont.)		marker Always set.	YBY buffer size		16 * 1024 * (this #) is minimum YBV...		
11	... size in bits required to decode the sequence.				constrained Set if true.	load intra Q matrix		

(If load intra Q matrix is set)	
12 -	648-bit unsigned integers defining the intra quantizer matrix.
75	⋮

(If load non-intra Q matrix is set)	
76 -	648-bit unsigned integers defining the non-intra quantizer matrix.
139	⋮

load non-intra Q matrix

(Skip to next start code)

Sequence extension data	Reserved for MPEG-2. Preceded by start code 0x000001B5. End is signaled by the presence of 0x000001.
--------------------------------	--

User data (optional)	Preceded by start code 0x000001B2. End is signaled by the presence of 0x000001.
-----------------------------	---

MPEG-1 Group of Pictures

byte	bit 7	6	5	4	3	2	1	0
0	0				0			
1	0				0			
2	0				Group start code			
3	B				8			
4	drop frame	hours (0-23)				minutes (0-59)		
5	[cont.]				marker always set	seconds (0-59)		
6	[cont.]				picture (0-59)			
7	[cont.]	closed gop #	broken link †	† Cleared during encoding. Set if editing has removed info needed to decode B-pictures after first I-picture of Group.				


Set if the Group is encoded without prediction vectors pointing to the previous Group. A closed Group may more easily be edited after encoding.


(skip to next start code)

Group extension data	Reserved for MPEG-2. Preceded by start code 0x000001B5. End is signaled by the presence of 0x000001.
User data (optional)	Preceded by start code 0x000001B2. End is signaled by the presence of 0x000001.
Pictures	Arbitrary number, preceded by Picture start code 0x00000100.

MPEG-1 Picture

	bit	7	6	5	4	3	2	1	0
byte									
0		0				0			
1		0				Picture start code			
2		0				1			
3		0				0			
4		temporal reference Unsigned integer. Set to zero for first displayed Picture of Group, then in-							
5		cremented mod 1024.		coding type Resp.: forbidden, I, P, B, D, rsvd...rsvd			YBY delay Time in 90 kHz ...		
6		... clock cycles needed to fill YBY buffer from empty state at target bitrate to correct level at start of play.							
7		For non-constant bitrate, is set to FFFF.							

 : For P- or B- pictures only.

 : For B-pictures only.

(cont.)	full back-ward vector *	backward f code †	full pel for-ward vector *	forward f code †
----------------	-----------------------------------	--------------------------	--------------------------------------	-------------------------

* Full vector flags are set if motion vector values decoded represent integer pixel offsets rather than half-pixels.

† F codes are unsigned non-zero integers describing decoding motion vectors as per §2.4.4.3, ISO 11172.

(Optional arbitrary number of 9-bit extra information structures.)	extra bit (set)	extra information
(Present only if extra bit is set. Always one byte long.)	•••	extra bit (cleared)

Extra information structures are terminated by a cleared extra bit.

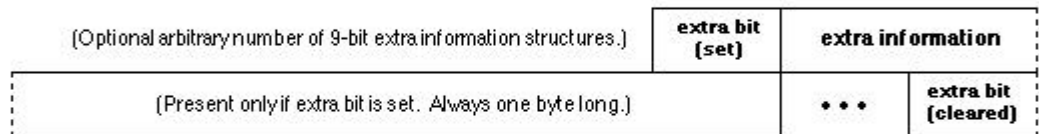
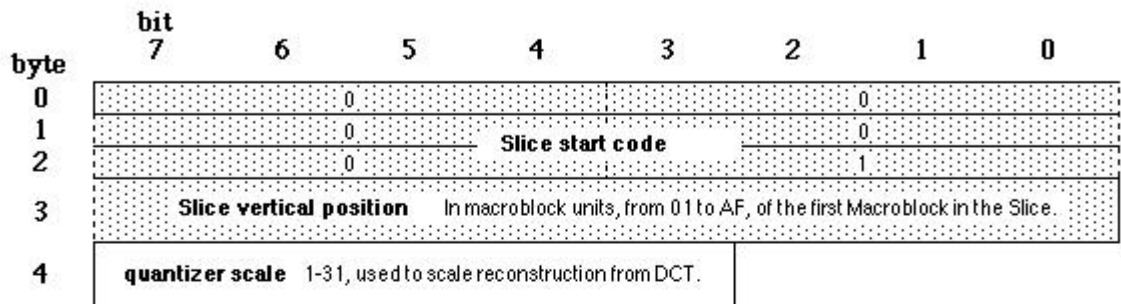
(skip to next start code)

Sequence extension data	Reserved for MPEG-2. Preceded by start code 0x000001B5. End is signaled by the presence of 0x000001.
--------------------------------	--

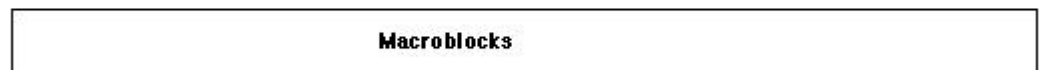
User data (optional)	Preceded by start code 0x000001B2. End is signaled by the presence of 0x000001.
-----------------------------	---

Slices	Preceded by slice start codes 00000101 - 000001AF.
---------------	--

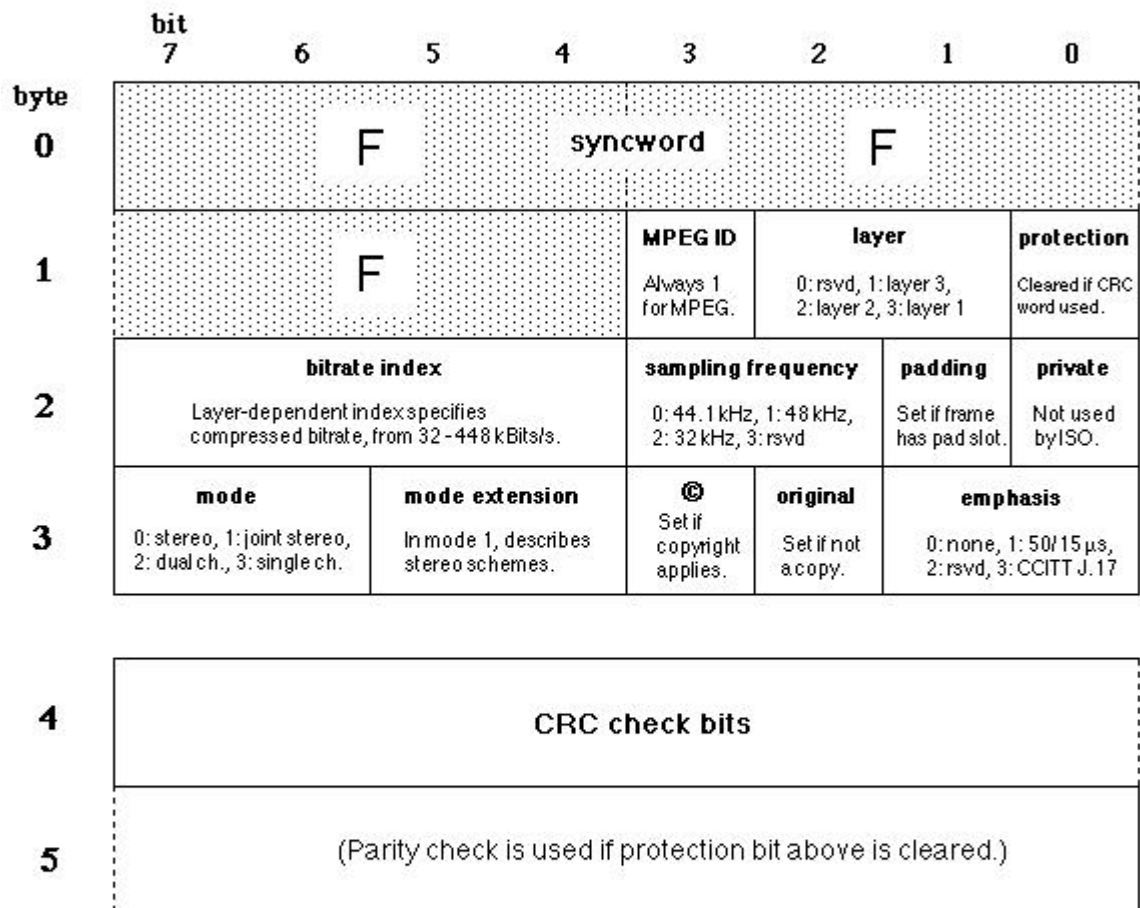
MPEG-1 Slice



Extra information structures are terminated by a cleared extra bit.



MPEG-1 Audio Frame Header



The way that MPEG-1 codes pictures

The human eye has a lower sensibility to color information than to dark-bright contrasts. A conversion from RGB-colour-space into YUV color components help to use this effect for compression. The chrominance components U and V can be reduced (subsampling) to half of the pixels in horizontal direction (4:2:2), or a half of the pixels in both the horizontal and vertical (4:2:0) [Djo07].

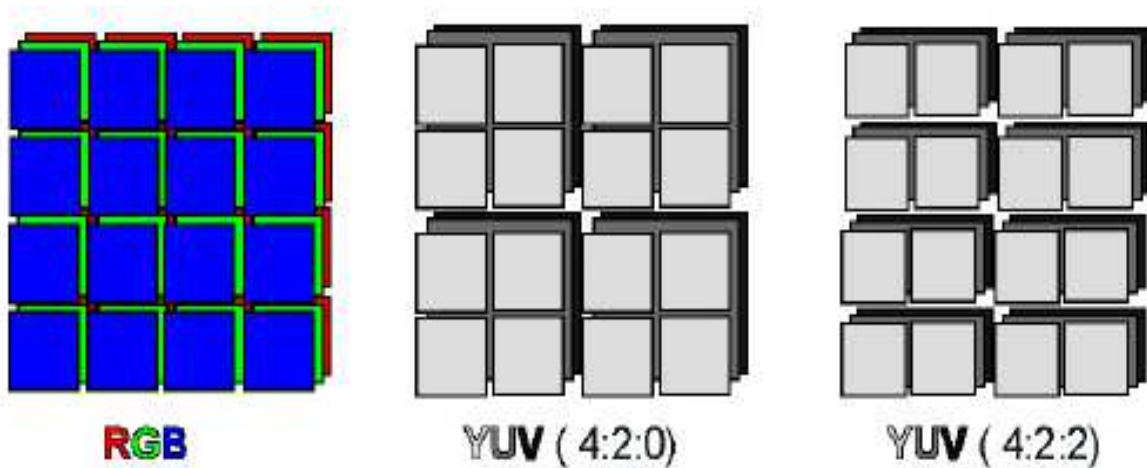


Figure (1) Subsampling

$$|Y|=|U|=|V|$$

$$4:2:0 = \frac{|Y| + \frac{1}{4}|U| + \frac{1}{4}|V|}{|Y| + |U| + |V|} = \frac{1}{2} \dots\dots\dots (1)$$

$$4:2:0 = \frac{|Y| + \frac{1}{2}|U| + \frac{1}{2}|V|}{|Y| + |U| + |V|} = \frac{2}{3} \dots\dots\dots (2)$$

The image to be coded is first partitioned into 8 x 8 blocks. Each 8 x 8 pixel block then subject to an 8 x 8 DCT, resulting in a frequency domain representation of the block as shown in Figure (2) [Yun99].

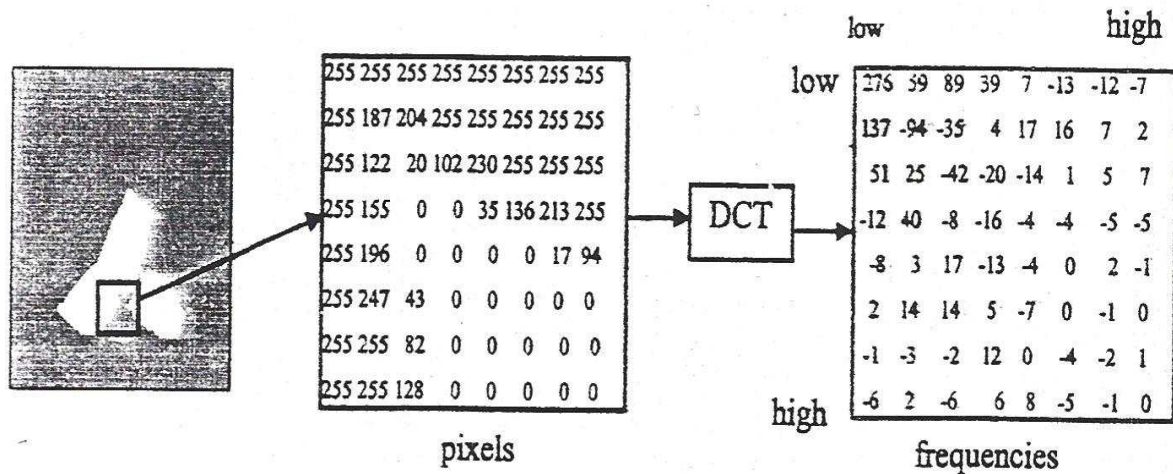


Figure (2) example of 8x8 pixels to DCT

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N} \quad \dots (3)$$

$$C(u), C(v) = 1/\sqrt{2} \text{ for } u, v = 0$$

$$C(u), C(v) = 1, \text{ else}$$

$$N = \text{block size}$$

The goal of the transformation is to decorrelate the block data so that the resulting transform coefficients can be coded more efficiently, the transform coefficients are then quantized, during the process of quantization a weighted quantization matrix is used. The function of quantization matrix is to quantize high frequencies with coarser quantization steps that will

suppress high frequencies with no subjective degradation, thus taking advantage of human visual perception characteristics.

$$F_{QUANTISED} = F(u, v) \text{ DIV } Q(u, v) \quad \dots\dots (4)$$

Where Q is the quantization matrix.

The bits saved for coding high frequencies are used for lower frequencies to obtain better subjective coded images. There are two quantizer weighting matrices in Test Model 5 (TM5) (ISO/IEC, 1993), an intraquantizer weighting matrix and a nonintraquantizer weighting matrix; the latter is flatter since the energy of coefficients in interframe coding is more uniformly distributed than in intraframe coding [Yan99].

In intra macroblocks, the DC value, dc, is an 11-bit value before quantization and it will be quantized to 8, 9, or 10 bits according to the setting of parameter. Thus, the quantized DC value, QDC, is calculated as

$$\text{8-bit: } QDC=dc//8, \text{ 9-bit: } QDC=dc//4, \text{ or 10-bit: } QDC=dc//2 \quad \dots\dots (5)$$

Where symbol // means integer division with rounding to the nearest integer and the half-integer values are rounded away for zero unless otherwise specified. The AC coefficients, ac (i, j), are first quantized by individual quantization factors to the value of ac - (i, j):

$$ac \sim (i,j) = (16 * ac(i,j)) // W1(i,j), \quad \dots\dots\dots(6)$$

Where W1 (i, j) is the element at the (i, j) position in the intraquantizer weighting matrix shown in Figure 3.

The quantized level QAC (i, j) is given by

$$QAC (i,j) = [ac \sim (i,j) + \text{sign}(ac \sim (i,j)) * ((p * mquant) // q)] / (2 * mquant), \quad \dots\dots\dots(7)$$

Where m_{quant} is the quantizer scale or step which is derived for each macroblock by rate control algorithm, and $p = 3$ and $q = 4$ in TM5 (ISO/IEC, 1993). For nonintra macroblocks,

$$ac_{\sim}(i,j) = (16 * ac(i,j)) // WN(i,j), \dots \dots \dots (8)$$

Where $WN(i, j)$ is the nonintraquantizer weighting matrix in Figure (3) and

$$QAC(i, j) = ac_{\sim}(i, j) / (2 * m_{\text{quant}}). \dots \dots \dots (9)$$

An example of encoding an intrablock is shown in figure (4) [Yun99].

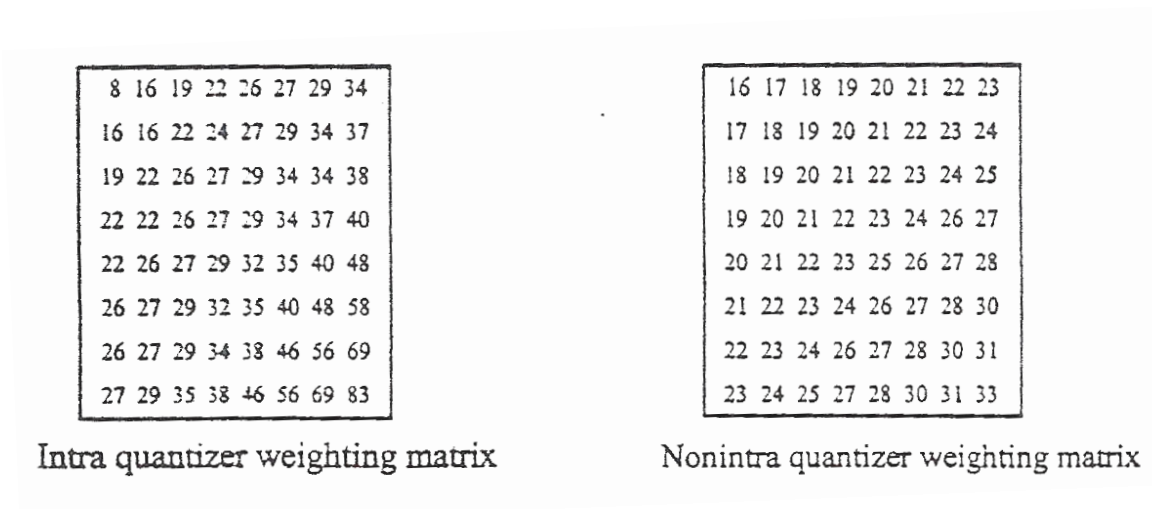


Figure 3 Quantizer matrices for intra and nonintracoding

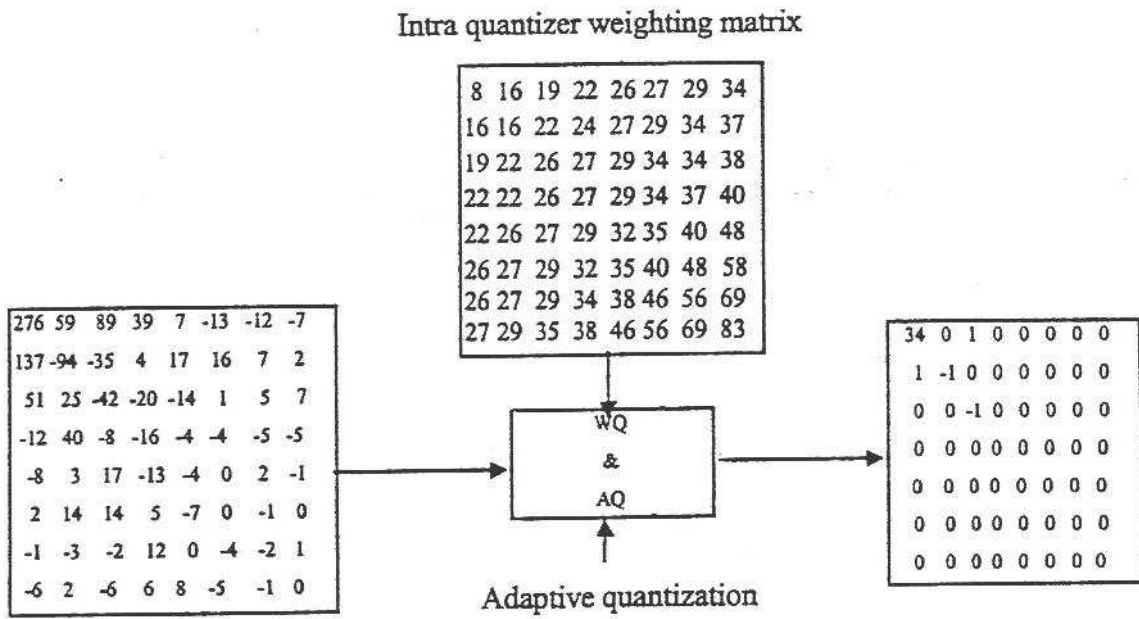


Figure (4) Example of encoding an intrablock

The coefficients are processed in zigzag order since the most energy is usually concentrated in the lower-order coefficients. The zigzag ordering of elements in an 8 x 8 matrix allows for a more efficient run-length coder. This is illustrated in figure (5).

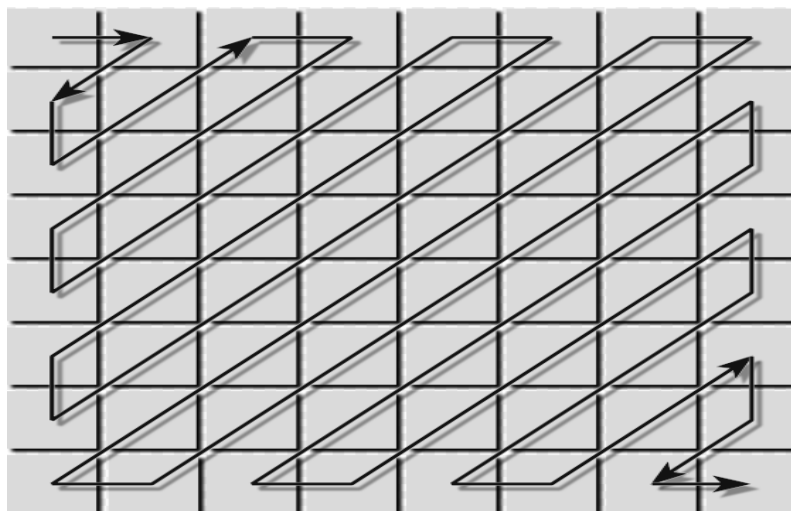


Figure (5) Zigzag

With the zigzag order, the run-length coder converts the quantized frequency coefficients pairs of zero runs and nonzero coefficients:

34 0 1 0 -1 1 0 0 0 0 0 0 -1 0 0 0 0.....

After parsing we obtain the pairs of zero runs and values:

34 | 0 1 | 0 -1 | 1 | 0 0 0 0 0 0 -1 | 0 0 0 0....

These pairs of runs and values are then coded by a Huffman-type entropy coder. For example, the above run/value, pairs are:

Run/Value 34	VLC (Variable Length Code)
1, 1	0110
1, -1	0111
0, 1	110
6, -1	0001011
End of block	10

The VLC tables are obtained by statistically optimizing a large number of training video sequences and are included in the MPEG-2 specification. The same idea is applied to code the DC values, motion vectors, and other information. Therefore, the MPEG video standard contains a number of VLC tables [Joh04].

Chapter Five

Conclusions and Future Work

5.1 Introduction

This chapter is dedicated to present some derived conclusions and a list of proposals for future work related to the research work discussed in the thesis.

5.2 Conclusions

From the result present in the previous chapter, some remarks related to the behavior and performance of the two suggested encryption system. A summary of some important conclusions could be presented as follows:-

1. The encryption under 1st complexity mode is faster than 2nd complexity mode because in the 1st complexity, just I pictures will be encrypted, while the 2nd complexity mode encrypts I and P pictures, this mean more data needs to encrypt due to more execution time.
2. Encryption ratio in the 2nd complexity is greater than 1st complexity, it is a variable value that depends on many parameters such as (GOP length, pictures size)
3. From the calculation of MSE and PSNR, the encryption under 2nd complexity mode destroys the visual data more than encryption under 1st complexity mode.
4. The probability of breaking encrypted file in the proposed method is very small as compared with the two other methods (Fibonacci method and Galois method).
5. The execution time is a variable value for the three implemented method and it depends on the nature of the sample.

5.3 Future work

1. Selective encryption can work on audio and video together instead of sequence pictures that related to video only.
2. The idea of selective encryption could be applied on modern versions of MPEG video files such as MPEG-2, MPEG-4, MPEG-7 and MPEG-21
3. Develop the encryption system to work in real time mode and this idea has benefits in TV satellites broadcasting in the encrypted channels.
4. Develop the encryption method by combining two methods or more to produce a new more complicated method hard to break.

Chapter Four

Tests and Results

4.1 Introduction

This chapter is developed to study the encryption performance of the three encryption methods (Random seed values, Fibonacci method and Galois method). First degree of complexity and second degree of complexity are investigated by performing set of suitable objective fidelity measures (such as Mean square error MSE and peak to signal noise ratio PSNR) on five different videos sequences that are taking as testing samples.

The developed system is implemented using Visual Basic Language (ver 6.0) under windows XP operating system. The system is executed using dell personal computer (processor Core 2 Due 2 GigaHz), 4 Mega cash.

4.2 Fidelity Criteria

Generally, fidelity criteria can be divided in to two classes:

1. **Objective fidelity criteria:** this kind of criteria is borrowed from digital signal processing and information theory, they provide equation that can be used to measure the amount of effected data in the encryption process. Commonly used objective measures are mean-square error (MSE) and the peak to signal noise ratio (PSNR).

Where

$$\text{MSE} = \frac{1}{W * H * N} \sum_{N=1}^N \sum_{r=0}^{H-1} \sum_{c=0}^{W-1} (I(r,c) - \hat{I}(r,c))^2 \dots\dots\dots (4.1)$$

Where I(r,c) is the pixel value of the original picture at the (r,c) location.

$\hat{I}(r,c)$ is the pixel value of the encrypted picture at the same location (r,c) .

The PSNR can be defined as: [scot98]

$$PSNR = 10L \log_{10} \frac{(L-1)^2}{\frac{1}{W * H} \sum_{r=0}^{H-1} \sum_{c=0}^{W-1} [\hat{I}(r,c) - I(r,c)]^2} \dots\dots\dots (4.2)$$

From quality of view, the larger number of MSE mean better destroying on the encrypted video. Alternately, with peak to signal noise ratio, small numbers of PSNR mean better destroying.

2. **subjective Fidelity Criteria**: Two main types of subjective measurements exist. The first is referred to as impairment tests, where the viewers score the pictures in terms of how bad they are.

The second type is referred to as quality tests, where the viewers score the picture in term of how good they are. [scot98]

4.3 Performance Parameters

In this research, the main parameters are the *overall time* required to perform the encryption process for all sensitive parts in the MPEG-1 file and *encryption ratio*. For all methods used in the system, the overall time includes generating of the keys and encryption process.

The minimization of searching time of the sensitive parts is considered as the most cost criteria, we do not need to encrypt the whole file, just the encryption of the sensitive parts of the MPEG-1 video files enough to destroy the video, and that is the one of the important aims of this work.

4.4 Samples Specification

To evaluate the performance of the suggested encryption system, five video sequences were taken. The first picture of each video is shown in figure (4.1). These five video have different number of pictures and sizes.



S1
Size: 818 KB

Figure (4.1.a) Sample 1



S2
Size: 1.714 MB

Figure (4.1.b) Sample 2



S3
Size: 833 KB

Figure (4.1.c) Sample 3



S4
Size: 465 KB

Figure (4.1.d) Sample 4



S5
Size: 1.421 MB

Figure (4.1.e) Sample 5

Table (4.1) Show the number of I, P and B pictures in the 5 samples that used in the evaluations with different GOP lengths. The encryption work only on sensitive parts on I and P pictures with no need to encrypt B pictures i.e. "*selective encryption*".

Table (4.1) Number of I, P and B pictures in five samples.

Samples	I pictures	P pictures	B pictures	GOP's	Total Pictures
S1	9	45	90	9	144
S2	19	38	112	19	169
S3`	11	30	80	11	121
S4	6	24	58	6	88
S5	14	42	110	14	166

4.5 Testing Strategy

The testing operation is implemented on five video samples, each sample was encrypted using three encryption methods (Random seed values, Fibonacci, Galois) twice, one for 1st complexity and the other one for 2nd complexity.

Testing tables illustrate the result of calculating encryption time, MSE, PSNR and encryption ratio for tested samples.

Table (4.2) shows the execution time for the three methods in 1st and 2nd degree of complexity.

Table (4.2) Execution time for the three methods

<i>Samples</i>	Random seed values		Fibonacci Method		Galois Method	
	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.
<i>S1</i>	8.19 sec.	9.67 sec.	10.27 sec.	12.11 sec	8.02 sec.	9.1 sec.
<i>S2</i>	12.09 sec.	13.42 sec.	13.36 sec.	15.64 sec.	11.75 sec.	13.08 sec.
<i>S3</i>	7.98 sec.	8.39 sec.	11.78 sec.	12.95 sec.	6.44 sec.	7.86 sec.
<i>S4</i>	4.33 sec.	4.89 sec.	6.14 sec.	7.92 sec.	4.08 sec.	4.97 sec.
<i>S5</i>	10.16 sec.	12.55 sec.	16.4 sec.	19.38 sec.	11.61 sec.	13.77 sec.

Table (4.3) MSE for five samples using three methods

<i>Samples</i>	Random seed values		Fibonacci Method		Galois Method	
	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.
<i>S1</i>	3292.83	6568.37	2793.63	6442.42	4002.45	4760.57
<i>S2</i>	5240.91	5595.33	3966.84	5110.63	3468.22	3840.21
<i>S3</i>	4963.49	6905.38	5118.35	7643.55	5749.57	7495.73
<i>S4</i>	4880.91	7862.58	2979.42	7013.55	3370.10	6518.99
<i>S5</i>	5917.64	6690.32	4113.64	6855.59	5877.82	7394.16

Table (4.4) PSNR for five samples using three methods

<i>Samples</i>	Random seed values		Fibonacci Method		Galois Method	
	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.	1 st Com.	2 nd Com.
<i>S1</i>	13.79	10.09	14.79	10.2	13.31	11.54
<i>S2</i>	11.14	10.74	12.40	11.25	14.15	12.57
<i>S3</i>	11.39	9.94	11.24	9.39	10.60	9.65
<i>S4</i>	11.55	9.49	13.89	10.01	13.19	10.77
<i>S5</i>	10.56	9.97	12.09	9.85	10.51	9.57

Table (4.5) shows the encryption ratio of sensitive parts in I pictures for 1st degree of complexity and (I&P) pictures for 2nd degree of complexity for 5 samples.

Table (4.5) Encryption Ratio for 5 Samples

Samples	Encrypted Data in 1 st Complexity	1 st Com. Encryption ratio (%)	Encrypted Data in 2 nd Complexity	2 nd Com. Encryption ratio (%)	Size of file
S1	8,590 KB	1.04 %	61,899 KB	7.56 %	817 KB
S2	224,419 KB	19.22 %	441,204 KB	37.79 %	1.14 MB
S3	110,593 KB	13.28 %	309,957 KB	37.24 %	832 KB
S4	51,112 KB	11.01 %	183,375 KB	29.8 %	464 KB
S5	174,073 KB	15.31 %	521,910	45.94 %	1.11 MB

From the testing tables of the three methods (shown in tables 4.2 to 4.5), one can notice the following:

1. The execution time of the 1st complexity in all three methods is usually less than the execution time in 2nd complexity because in 1st complexity, less data will be encrypted than 2nd complexity. The increasing ratio of time will be approximately from (12 to 15%) from the total execution time.

2. The execution time of Galois method (in 1st and 2nd complexity) is less than other method (i.e. the encryption using Galois method is faster than Fibonacci and Random seed values).
3. The MSE of 2nd complexity is higher than 1st complexity in all three methods, that's mean the encryption in 2nd complexity destroy the visual information more than 1st complexity. Also PSNR results show that the encryption in 2nd complexity is better than 1st complexity.
4. The testing results of the Random seed values for (1st and 2nd complexity) techniques are very good results. This idea is obvious through the results in testing tables, where MSE and PSNR of the Random seed values in the tables (4.3) and (4.4) show good destroying of visual information more than the other two methods (Galois and Fibonacci).
5. Encryption ratio is a variable ratio that depends on some properties of the video such as (size of image and GOP length), noticing that
 - If the sizes of (I) pictures are large, then the encryption ratios will increase in (1st and 2nd complexity).
 - If the sizes of (B) pictures are large, then the encryption ratios will decreases in (1st and 2nd complexity).
 - If the sizes of (P) pictures are large, then the encryption ratios will increase in (2nd complexity) only.
 - Encryption ratio increases if the length of the GOP is small and vice versa.

4.6 Brute Force

In cryptanalysis, a brute force attack is a method of defeating a cryptographic scheme by trying a large number of possibilities, for example, exhaustively working through all possible keys in order to decrypt a message. In most schemes, the theoretical possibility of a brute force attack is recognized, but it is set up in such a way that it would be computationally infeasible to carry out.

The selection of an appropriate key length depends on the practical feasibility of performing a brute force attack. By obfuscating the data to be encoded, brute force attacks are made less effective as it is more difficult to determine when one has succeeded in breaking the code. For symmetric-key ciphers, a brute force attack typically means a brute-force search of the key space, by testing all possible keys in order to recover the plaintext used to produce a particular ciphertext. The expected number of trials before the correct key is equal to half the size of the key space. For example, if there are 2^{64} possible keys, a brute force attack would be 2^{63} trials to find the correct key. [Bru96]

• Brute Force For Random seed values

In the proposed method, we use a key of 10 characters to generate 5 integer seed values, each integer consist of 16 bits so

Brute force= 2^{79} trials.

- **Brute Force For Fibonacci Method**

The 10 characters key that used to generate the initial state of 16 bit so

$$\text{Brute force} = 2^{15} \text{ trials.}$$

- **Brute Force For Galois Method**

The 10 characters key that used to generate the initial state of 16 bit so

$$\text{Brute force} = 2^{15} \text{ trials.}$$

4.7 Subjective Samples Testing

To evaluate the suggested encryption methods (for 2 complexity degree) subjectively, 7 study pictures are chosen, the first figure (4.2 a) is the first picture of sample 2, figures (4.2.b), (4.2.c), (4.2.d), (4.2.e), (4.2.f) and (4.2.g) show the same picture encrypted by three encryption methods with 1st and 2nd degree of complexity.



**Figure 4.2(A) Sample 2
Original Picture**



Figure 4.2(B) picture encrypted in First Complexity Random seed values



Figure 4.2(C) picture encrypted in Second Complexity Random seed values

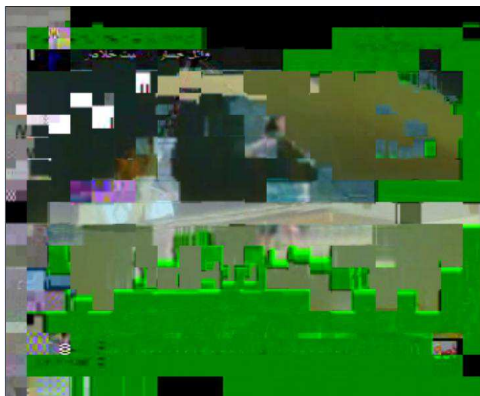


Figure 4.2 D picture encrypted in First Complexity Fibonacci Method

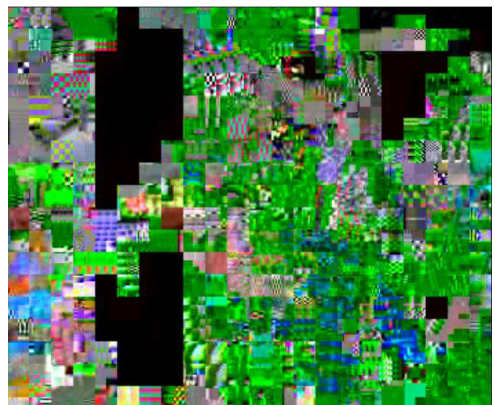


Figure 4.2 E picture encrypted in Second Complexity Fibonacci Method



Figure 4.2 F picture encrypted in First Complexity Galois Method



Figure 4.2 G picture encrypted in Second Complexity Galois Method

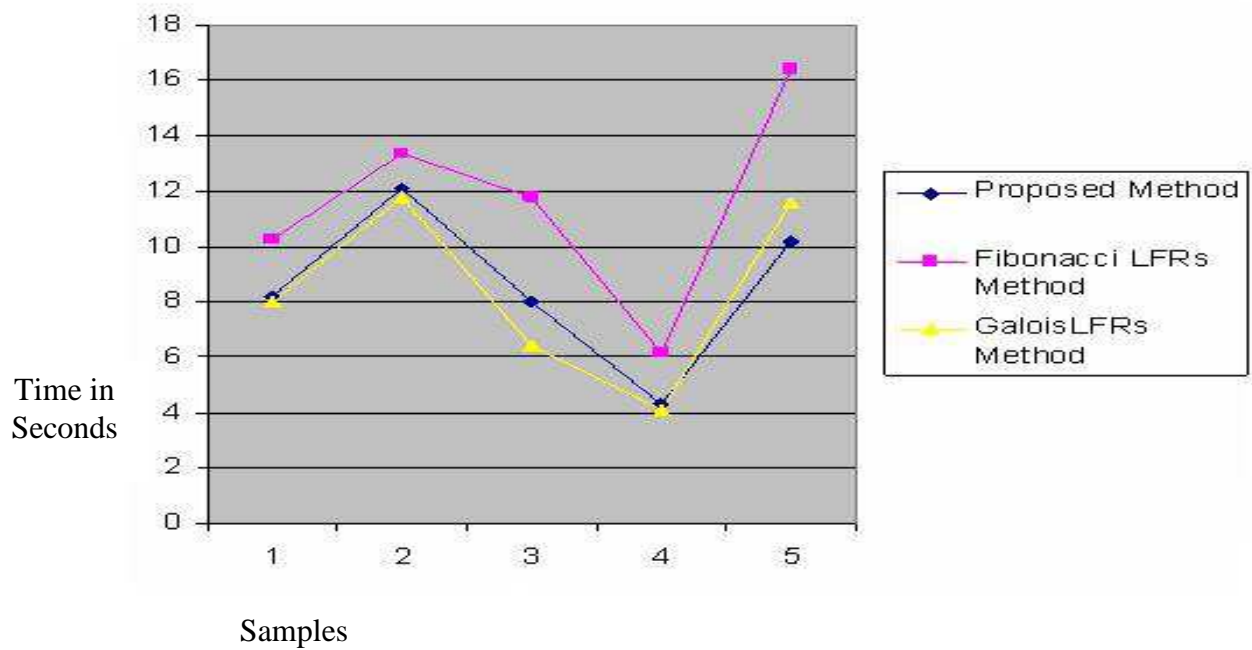


Figure (4.3) Time of the three methods for 1st complexity encryption

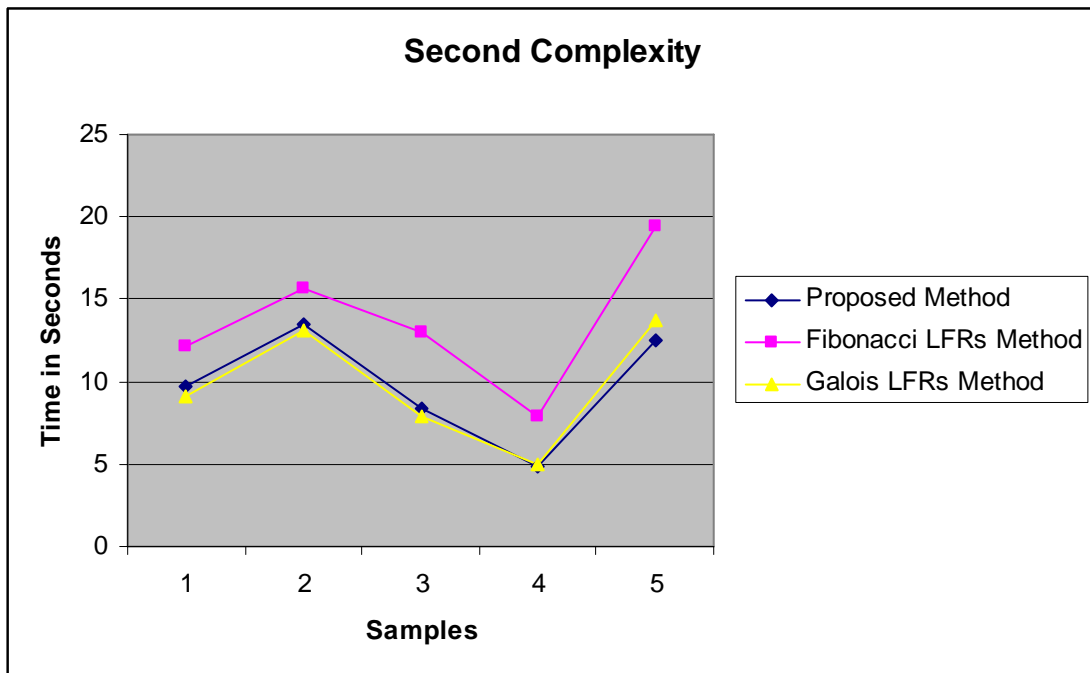


Figure (4.4) Time of the three methods for 2nd complexity encryption

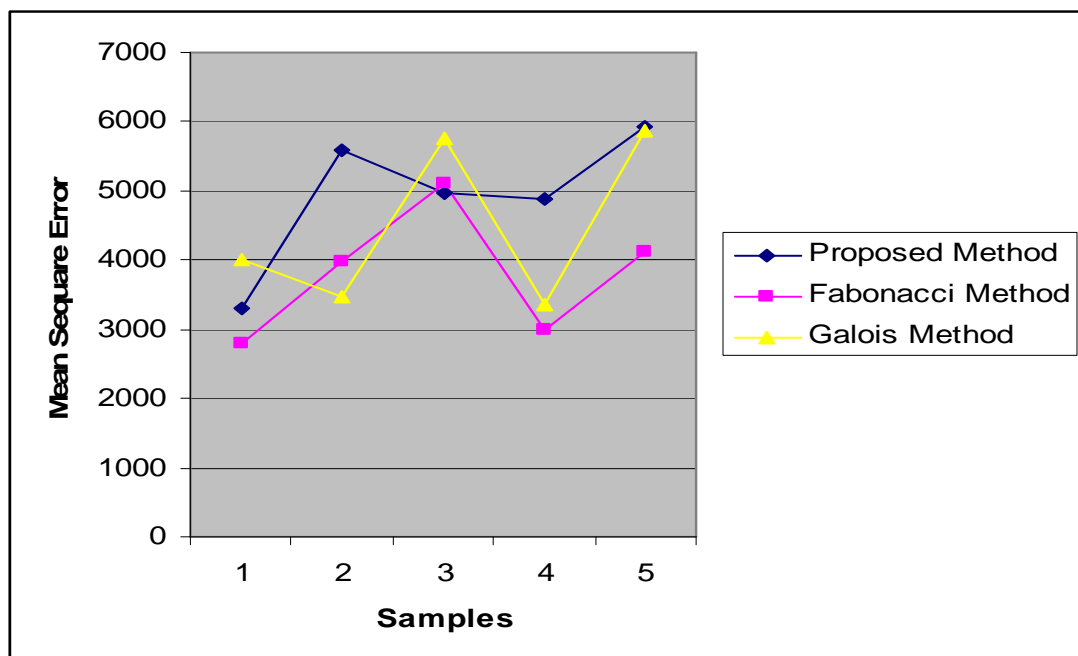


Figure (4.5) MSE of the three methods for 1st complexity encryption

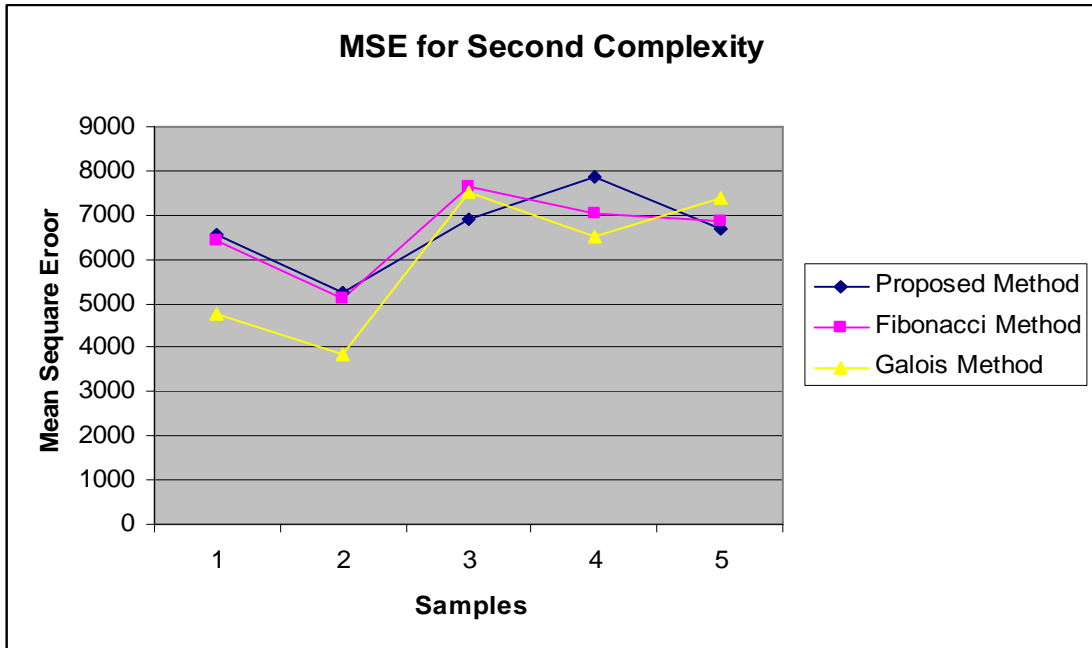


Figure (4.6) MSE of the three methods for 2nd complexity encryption

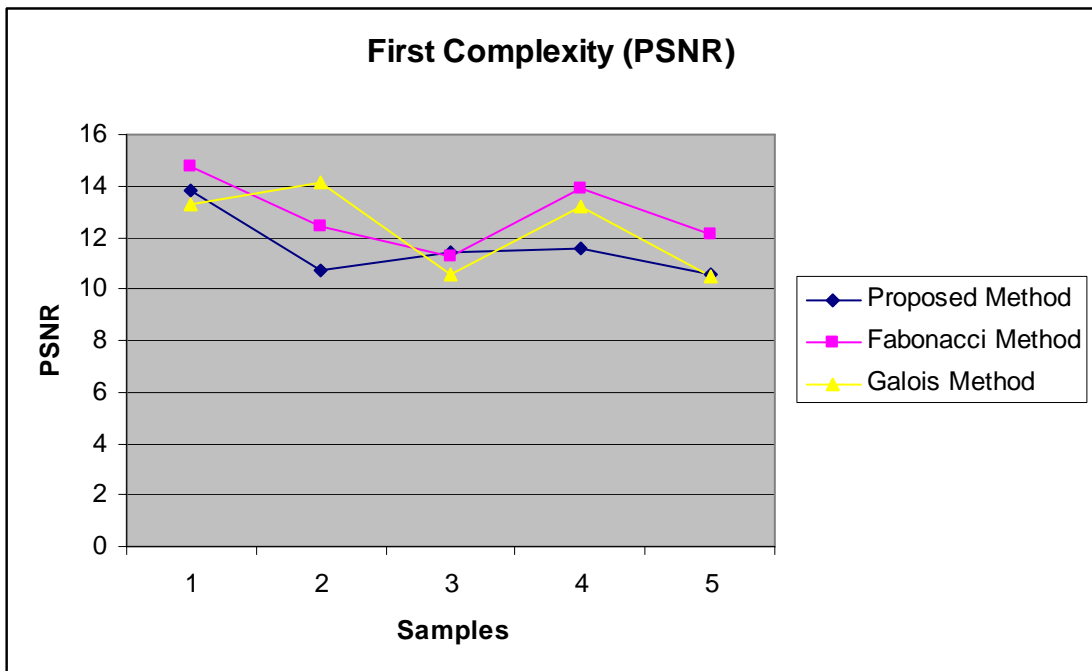


Figure (4.7) PSNR of the three methods for 1st complexity encryption

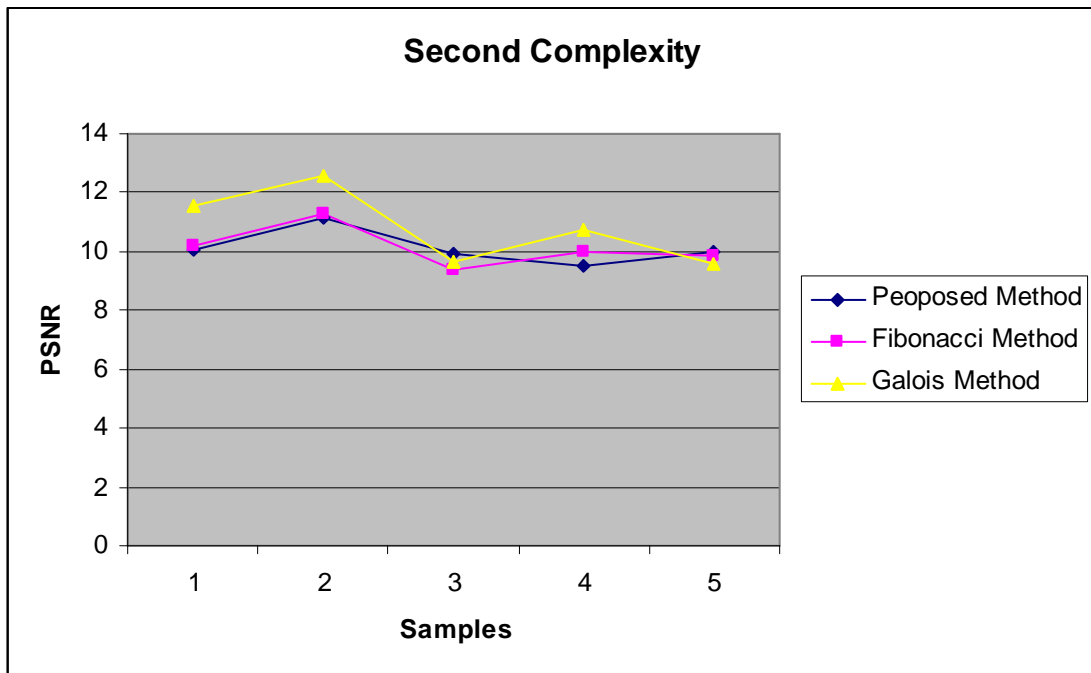


Figure (4.8) PSNR of the three methods for 2nd complexity encryption

Chapter One

Introduction

1.1 Digital Video

Digital television is the sending and receiving of moving images and sound by means of discrete (digital) signals, in contrast to the analog signals used by analog TV. Introduced in the late 1990s, this technology appealed to the television broadcasting business and consumer electronics industries as offering new financial opportunities.

Video recording in digital form, In order to edit video in the computer or to embed video clips into multimedia documents, a video source must originate from a digital camera or be converted to digital. Frames from analog video cameras and (Video Cassette Recorder) VCRs are converted into digital frames (bitmaps) using frame grabbers or similar devices attached to a computer [Ral95].

1.2 Encryption

Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information (in cryptography, referred to as ciphertext). In many contexts, the word encryption also implicitly refers to the reverse process, decryption (e.g. “software for encryption” can typically also perform decryption), to make the encrypted information readable again (i.e. to make it unencrypted) [Men96].

1.3 General Encryption Approaches

In general, there are two encryption approaches as listed below:

- **Stream cipher:** is a symmetric cipher where plaintext bits are combined with a pseudorandom cipher bit stream (keystream), typically by an exclusive-or (XOR) operation. In a stream cipher the plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption [Chr05].
- **Block cipher** is a symmetric key cipher which operates on fixed-length groups of bits, termed blocks, with an unvarying transformation. When encrypting, a block cipher might take a (for example) 128-bit block of plaintext as input, and output a corresponding 128-bit block of ciphertext. The exact transformation is controlled using a second input (secret key). Decryption is similar, the decryption algorithm takes, in this example, a 128-bit block of ciphertext together with the secret key, and yields the original 128-bit block of plaintext [Chr05].

1.4 Video Encryption

Video Encryption is an extremely useful method for the stopping unwanted interception and viewing of any video or other information, for example from a law enforcement video surveillance being relayed back to a central viewing centre. The human eye is very good at, spotting distortions in pictures due to poor video decoding or poor choice of video encryption software. Therefore, it is very important to choose the right software or else the video may be un-secure. Modern image and video compression techniques today offer the possibility to store or transmit the vast amount of data necessary to represent digital images and video in an efficient and robust way [Shi04].

1.5 Moving Picture Experts Group (MPEG):

This types of files was established in 1988 in the framework of the Joint ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) Technical Committee (JTC 1) on Information Technology with the mandate to develop standards for coded representation of moving pictures, associated audio and their combination when used for storage and retrieval on digital storage media. The MPEG committee began life by the hand of Leonardo Chairigloione and Hiroshi Yasuda with the immediate goal of standardizing video and audio for compact disks. MPEG adopts the CCIR601 digital TV format also known as SIF (Source Input Format) [**Web1**].

Many MPEG versions were proposed are listed below:

MPEG-1 supports only non-interlaced video with a bitrate at up to about 1.5 Mbit/s, it's a standard is also referred to as ISO/IEC 11172 and, it has five parts: 11172-1 Systems, 11172-2 Video, 11172-3 Audio, 11172-4 Conformance, and 11172-5 Software [**Shu07**].

Normally, its picture resolution is:

- 352 * 240 for NTSC (National Television System Committee) video at 30 FPS (Frame per second).
- 352 * 288 for PAL (Phase Alternating Line) video at 25 FPS.
- It uses 4:2:0 chroma subsampling

MPEG-2 was designed to provide video quality not lower than NTSC/PAL and up to Committee Consultative International pour la Radio CCIR 601 quality with bitrates targeted between 2 and 10 Mbit/s. emerging applications, such as digital cable TV distribution, satellite broadcasting

distribution and terrestrial digital broadcasting distribution, were seen to benefit from the increased quality expected to result from the emerging MPEG-2 standard. The MPEG-2 standard was released in 1994 [Web2].

MPEG-3 is the designation for a group of audio and video coding standards agreed upon by MPEG .It was designed to handle HDTV (High-definition television) signals in the range of 20 to 40 Mbit/s [Web2].

MPEG-4 is a standard used primarily to compress audio and visual digital data (AV). Introduced in late 1998, it is the designation for a group of audio and video coding standards under the formal standard ISO/IEC 14496. The uses for the MPEG-4 standard are web (streaming media),CD distribution, conversation (videophone), and broadcast television, all of which benefit from compressing the AV stream [Web2].

MPEG-7 Is a multimedia content description standard that allow fast and efficient searching for material that is of interest to the user. It is formally called Multimedia Content Description Interface. Thus, it is not a standard which deals with the actual encoding of moving pictures and audio, like MPEG-1, MPEG-2 and MPEG-4. It uses XML (Extensible Markup Language) to store metadata, and can be attached to time code in order to tag particular events, or synchronize lyrics to a song, for example [Web2].

MPEG-21 aims at defining an open framework for multimedia applications ISO 21000, specifically; MPEG-21 defines a "Rights Expression language" standard as means of sharing digital rights/permissions/restrictions for digital content from content creator to content consumer [Web2].

1.6 Job of MPEG Digital video

The job of MPEG is to take analogue or digital video signals and convert them to packets of digital data that are more efficiently transported over a network. Being digital it has the following. Advantages:

- Signal does not degrade.
- Picture does not get fuzzy.
- Signal-to-Noise ratio goes down.

MPEG is derived from the original work by the Joint Pictures Expert Group (JPEG). The JPEG standard is for still images and is a lossy technique. It takes advantage of the nature of the human eye and removes redundant information that do not see. JPEG was approved in 1994 as ISO 10918-1, The JPEG standard specifies both the *codec*, which defines how an image is compressed into a stream of bytes and decompressed back into an image, and the *file format* used to contain that stream. The compression method is usually lossy compression, although there are variations on the standard baseline JPEG that are lossless [Sha96].

1.7 Literature survey

Many researchers study the field of image encryption; few of them focus on video encryption, some of them are listed below:

1. **Chun Yuan, Bin B. Zhu and et.al, Efficient and Fully Scalable Encryption for MPEG-4, 2001 [Chu01].**

This research proposes a novel and low complexity scheme to encrypt MPEG-4 streams. The encrypted MPEG-4 stream can be

processed by middle stages directly on the ciphertext without decryption. In addition that this research propose scheme that has no degradation on either compression efficiency or error resilient performance, and allows random access.

2. Shujun Lian, Chaotic Encryption Scheme for Real-Time Digital Video, 2002 [Shu02]

This research proposes a novel video encryption scheme based on multiple digital chaotic systems, which is called CVES (Chaotic Video Encryption Scheme). CVES is independent of any video compression algorithms, and can provide high security for real-time digital video with fast encryption speed, and can be simply realized both by hardware and software.

3. Jason But, Limitation of Existing MPEG-1 Ciphers for Streaming Video 2004 [Jas04].

This work presents the suitability of the encryption algorithms in a streaming video context. The conclusion is that none of the existing ciphers are suitable for use in streaming MPEG-1 video and that a new encryption algorithms are required for this purpose.

4. Shiguo Lian, Jinsheng Sun and et.al, A Fast Video Encryption Scheme Based-on Chaos,2004 [Shi04]

This research proposes a cryptosystem for encrypts runlength codes with chaotic run-length encryption algorithm (CREA), encrypts the signs of motion vectors with security-enhanced chaotic stream cipher (SECSC) and distributes keys with chaotic key

distributor (CKD) at the same time. Its security, compression ratio and computational complexity are analyzed in details.

5. Dhiah Eadan Al-Shammary, Interframe Compression Using Distributed Systems, 2005 [Dhi05]

This work aims to develop two different models for video compression. It implements most of the well-known motion search methods with testing and their performances are investigated. This work implements the Fractal Image Compression technique, in addition to the DCT as images transform coding technique.

6. Deniz Taskin, Cem Taskin and et.al, Selective Encryption of Compressed Video Files, 2007 [Den07]

The project addresses the security requirements, and develops conventional methods for compressed video encryption. These approaches are called naive approach and require plenty of system resources. For real time encryption of video files and video stream and offer selective encryption based on RSA asymmetrical encryption method.

7. Shujun Li, Guanrong Chen and et. al, On the Design of Perceptual MPEG-Video Encryption Algorithms 2007 [Shu07].

In this research, some existing perceptual encryption algorithms of MPEG videos are reviewed and some problems, especially security defects of two recently proposed MPEG-video perceptual encryption schemes, are pointed out. The research tries to use

simpler and more effective designs, which selectively encrypts fixed-length codewords in MPEG-video bit streams.

1.8 Aim of Thesis

The proposed work aims to study the internal structure of MPEG-1 video files and apply the idea of selective encryption to MPEG-1 files to minimize the complexity due to less time needed in the process of encryption and decryption, this is a very useful idea for security purpose in transmitting video files over network or storing important video files.

1.9 Thesis Layout

The work in this thesis is organized as follows:

- **Chapter (2):** explains selective encryption, types of video encryption, image and video compression and the internal structure of MPEG-1 video file in details.
- **Chapter (3):** this chapter includes all the details of the designed and implemented video encryption system. All the algorithms used in this work are presented.
- **Chapter (4):** this chapter contains the result of some tests applied on some samples of movies used as test material in this work, the used criteria are fidelity measures (MSE, PSNR) beside the encryption ratios.
- **Chapter (5):** includes the derived conclusions and some suggestions for future work.

Chapter Three

The Proposed Encryption System

3.1 Introduction

The protection of visual property of multimedia content in networks and data storage is a major challenge now, Content protection is provided by cryptography. In public key cryptosystem, there are two keys: a public key which is publicly known and the private key which is kept secret by the owner. The system called asymmetric because different keys are used for encryption and decryption. If data are encrypted with a public key, it can only be decrypted using corresponding private key.

Due to the increase in processor speed on even more to smart cryptanalysis, the key size for public key cryptography grew very large, this created a disadvantage in asymmetric key cryptosystems. Public key cryptography is slower and requires a large memory capacity and large computational power, symmetric key approach solves these problems and meets the real-time constraint for video playback.

The technique of selective encryption encrypts some parts of a compressed data file while leaving others unencrypted. It is a strategy that small fraction of encrypted bits can cause a high ratio of damage to a file. Instead of encrypting the whole file bit by bit, only highly sensitive bits are changed as seen in figure (3.1). Moreover selective encryption reduces required total encryption work and saves system resources.

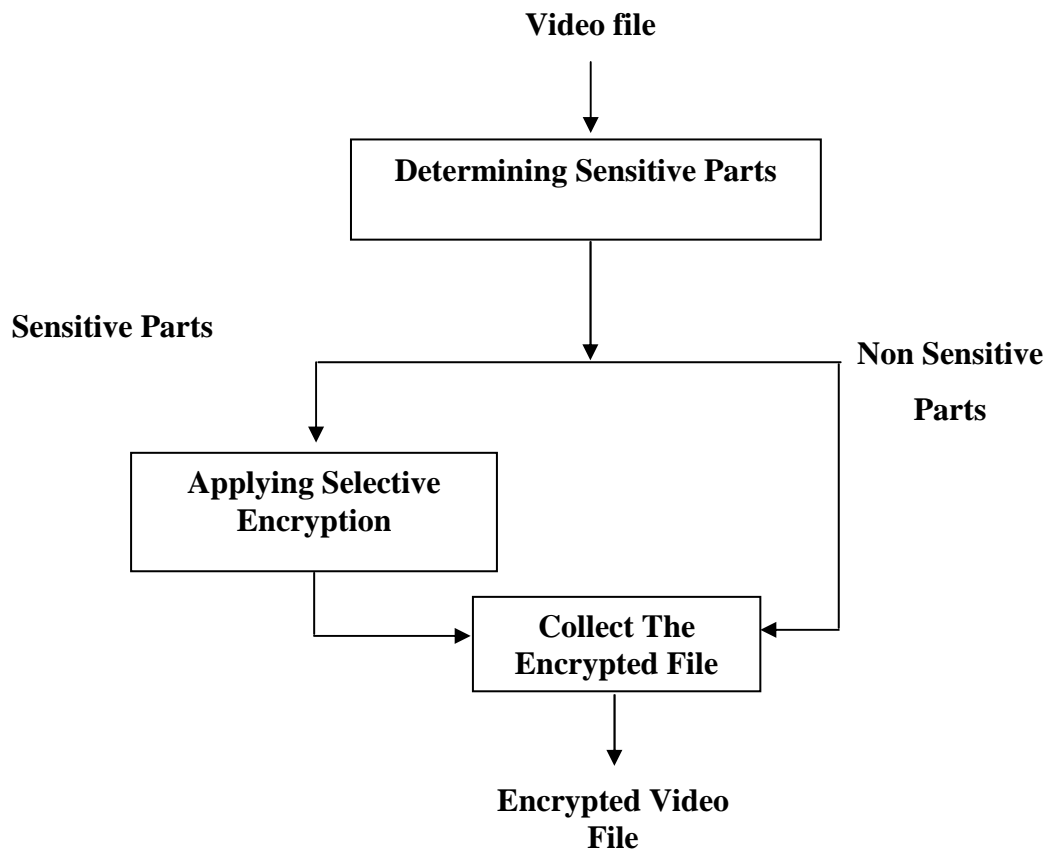


Figure 3.1 General review of selective video encryption

Where this strategy encrypts some parts of compressed video file selectively while guaranteeing the security of the original file. This strategy not saving just the time for encrypting video file, but also system complexity.

Selective encryption may not be effective if the subset is small and it's relatively easy to guess. This makes compression standard very important and have to be studied for making a perfect selective video encryption.

3.2 Media File

MPEG-1 standard is commonly preferred as a solution of compressing video data because of its higher compression ratio. It is still largely employed in video compression and communication industry, the key of higher compression ratio is similarity between pictures in a sequence. A video sequence is simply a series of pictures taken at closely spaced intervals in time. These pictures tend to be quite similar from one to next. MPEG-1 compression system takes advantage of this similarity and it is possible to apply selective encryption on it. Most of video encryption techniques use some important parts of MPEG file and encrypt them to produce a destroyed video file.

3.3 The Proposed Encryption System

In the proposed work, three encryption algorithms (*Galois*, *Fibonacci LFSRs* and *the Random Seed values Method "Proposed Method"*) are implemented, all of these algorithms work in stream cipher mode, while block cipher can not be applied in such situation because there are different sizes (part size) and block cipher needs a fixed size to work properly. The algorithms are applied on different parts of MPEG-1 video file, as listed below:

1. Apply one of these three algorithms to (I) pictures only.
2. Apply one of these three algorithms to (I&P) pictures only.

3.4 Encryption System Structure

Figure (3.2) illustrates the general proposed encryption system. The steps of work are mentioned below:-

1. At the first, the system will open the selected MPEG-1 video file as binary file.

2. Parsing the MPEG-1 file to set main parameters that found in successive headers.

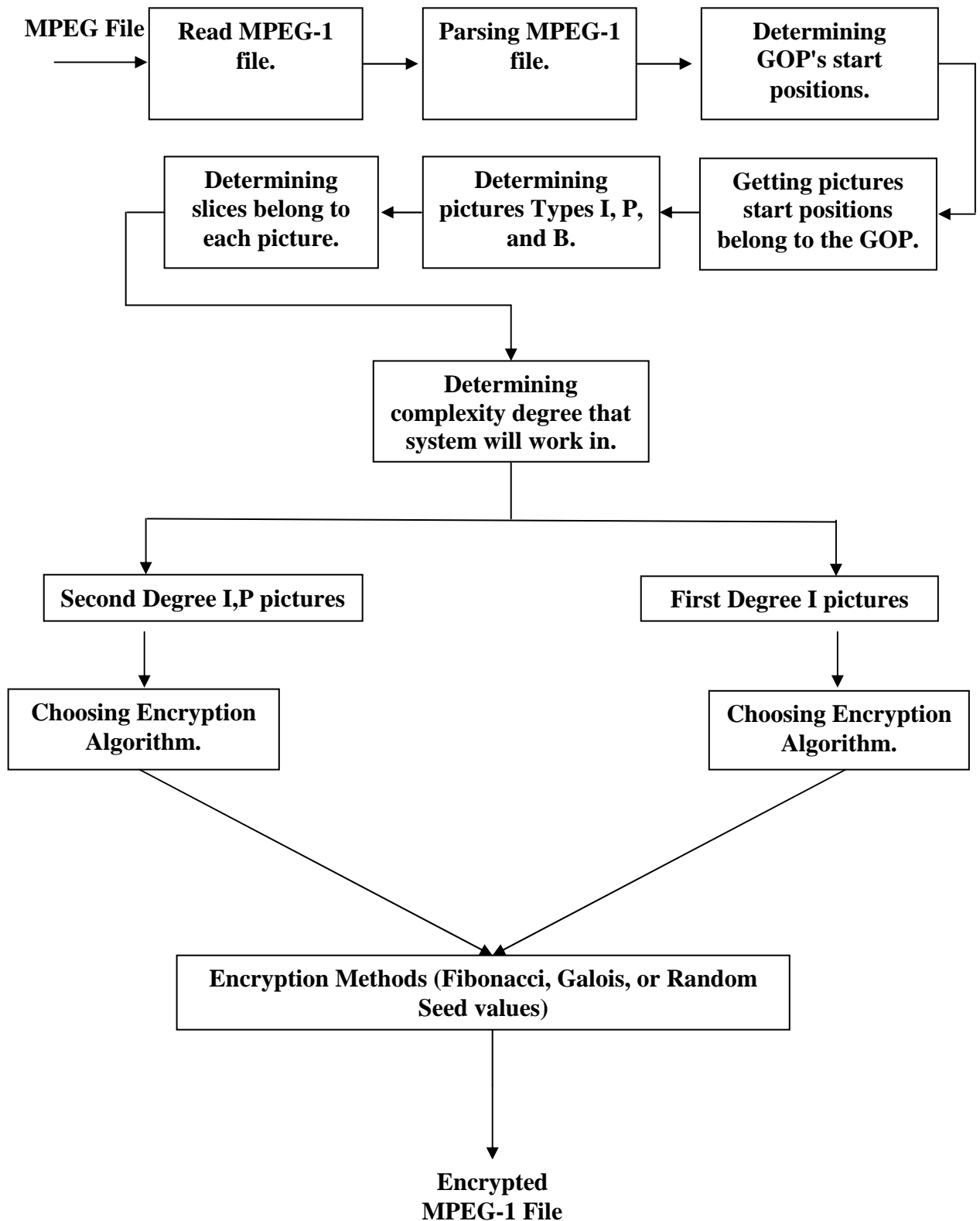


Figure (3.2) The Structure of Encryption System

3. Determining GOP's start positions in the MPEG-1 file, and pictures included in the GOP.
4. Determining pictures type (I, P and B) from picture header.
5. Determining start and end positions of all slices belong to each picture.
6. Complexity Degree: In the proposed work, two complexity degrees, as follow:
 - *First degree*: Apply encryption algorithms to the slices belong to all I frames in the MPEG-1 file.
 - *Second Degree*: Apply encryption algorithms to the slices belong to all I and P frames in the MPEG-1 file.
7. Encryption Algorithms: Three methods in the proposed system are used:
 - *Galois Method*.
 - *Fibonacci LFSRs Method*.
 - *Random Seed Values Method*.

3.5 Parsing of MPEG-1 Video File:

MPEG-1 video file contains important parameters that determine information needed in video players such as:

- Multiplex rate: audio and video multiplexing rate.
- Header length.
- Stream id: determine the next stream if it is audio or video.
- Packet length.
- Horizontal and vertical size in pixel.
- Pixel aspect ratio: defined by a code, this represents the height and width of the video image.

- Picture rate: defined by a code that represents the number of pictures that may be displayed each second. Algorithm (3.1) shows MPEG-1 parsing procedure.

Algorithm (3.1) Parsing MPEG-1 File

Input: MPEG-1 file.

Output: Main Parameters of MPEG-1.

Procedure

While not end of file

Load 4 bytes from the file b (1)... b (4)

Case b (1), b (2), b (3) and b(4)

- 00, 00, 01, BA :MPEG-1 Pack header
Save system clock reference;
Save Multiplex rate;
- 00, 00, 01, BB : MPEG-1 System header;
Save header length.
- 00, 00, 01, (110XXXXX): for audio stream;
Data comes after this header belongs to audio bound.
- 00, 00, 01, (1110XXXX): for video stream;
Data comes after this header belongs to video.
- 00, 00, 01, B3 : MPEG-1 Sequence Header
Save horizontal and vertical size, pixel aspect ratio, picture rate and bit rate.

End of Case

End of While

3.6 Determining GOP's, pictures start positions and relative parameters

For each GOP, there is only one I picture and many P and B pictures. GOP structure is mandatory in MPEG-1. A GOP structure also introduces error resilience due to frequent updates of intra coded pictures.

Furthermore, the GOP structure results in a periodic pattern of the different picture types making a layered approach possible.

GOP header contains parameters about time (hours, minutes and seconds) and number of pictures included in the GOP. All start positions of pictures in the GOP will be determined and saved for encryption processing.

Picture header contains information about temporal reference, picture coding types (I, P or B), buffer delay (time in 90 KHz clock cycles needed to fill buffer from empty state at target bit rate to correct level at start of play) and information about motion compensation (backward or forward) prediction for (P and B) pictures.

Algorithm (3.2) Determine GOP's and pictures positions

Input: MPEG-1 file.

Output: 1-D Array contains GOPs start positions, 1-D array contains start positions of all I pictures, 1-D array contains start positions of all P pictures, 1-D array contains start positions of all B pictures and 2-D array contains all pictures with types.

Procedure

I_pic_count=0: P_pic_count=0: B_pic_count=0

Pos=0: gop_count=0: all_pic_count=0

While not end of file

Load 4 bytes from file in the location Pos as b (1)... b (4)

Case b (1), b (2), b (3), b (4)

- 00, 00, 01, B8: GOP start position.

gop_count = gop_count +1;

gop(gop_count)= Pos;

save time in hour, minute and second and number of pictures.

- 00, 00, 01, 00: Picture start position.

Save temporal reference;

```

If picture = (I) then // coding type
    I_pic_count = I_pic_count +1;
    I_pic(I_pic_count)= Pos;
    all_pic_count=all_pic_count+1
    all_pic(all_pic_count,1)=pos;
    all_pic(all_pic_count,2)=1; "For I pic."
End If
If picture = (P) then // coding type
    P_pic_count = P_pic_count +1;
    P_pic(P_pic_count)= Pos;
    all_pic_count=all_pic_count+1
    all_pic(all_pic_count,1)=pos;
    all_pic(all_pic_count,2)=2; "For P pic."
End If
If picture = (B) then // coding type
    B_pic_count = B_pic_count +1;
    B_pic(B_pic_count)= Pos;
    all_pic_count=all_pic_count+1
    all_pic(all_pic_count,1)=pos;
    all_pic(all_pic_count,2)=3; "For B pic."
End If

```

Determine the prediction type for P and B pictures.

Backward and forward

End of case

Increment Pos by 1;

End of While

3.7 Determining slices for each picture

Each picture consists from number of slices, these slices are differ in size making block cipher imposable to apply in the proposed system.

For more flexibility in the system, it must determine slice (start and end) positions for each picture (the size of slice) and the type of picture that slice belong to, because only more sensitive bytes in the slice will send to encrypt using one of the encryption methods.

Algorithm (3.3) Determine start and end positions of all slices in the file

Input: MPEG-1 video file, 2-D array for all pictures with the types.

Output: 2-D array contain (picture type, start, end) positions of all slices in

all pictures.

Procedure

all_slice_co=0: Flag=0;

//for all pictures in file except the last one.

For i=1 to all_pic_count -1

 j=all_pic(i,1)

 While j< all_pic(i+1,1)

 Load 4 bytes from file in the location j as b(1)...b(4).

// The start codes of all slices will be in the range (from 01 to AF)

 Case b(1), b(2),b(3),b(4)

- From 01 to AF and flag=0;

 all_slice_co= all_slice_co+1;

 type_slice_st_ed(all_slice_co,1)=all_pic(i,2);

// determining the type of picture that slice belong to

 type_slice_st_ed(all_slice_co,2)=j; *// slice start*

 flag=1;

- From 01 to AF and flag=1

//the beginning of the next slice and end of previous slice

 j=j-1;

 type_slice_st_ed(all_slice_co,3)=j; *// slice end*

```
                flag=0;
            End of case
            j=j+1;
        End of While
    Next i

//for the last picture in the file
j=all_pic(all_pic_count,1)
While not end of file
    Load 4 bytes from file in the location j as b (1)... b (4)
    Case b(1), b(2),b(3),b(4)
        • From 01 to AF and flag=0;
          all_slice_co= all_slice_co+1;
          type_slice_st_ed(all_slice_co,1) = all_pic(i,2);
          type_slice_st_ed(all_slice_co,2) = j; // slice start
          flag=1;
        • From 01 to AF and flag=1
          j=j-1;
          type_slice_st_ed(all_slice_co,3) = j; // slice end
          flag=0;
    End of case
    j=j+1
End of While
```

3.8 Galois Encryption Method

Is an alternate structure that can generate the same output sequences as a conventional LFSR.

3.8.1 Generation of state

The first state of Galois encryption method is generated from secret key of the proposed encryption system as shown in figure (3.3)

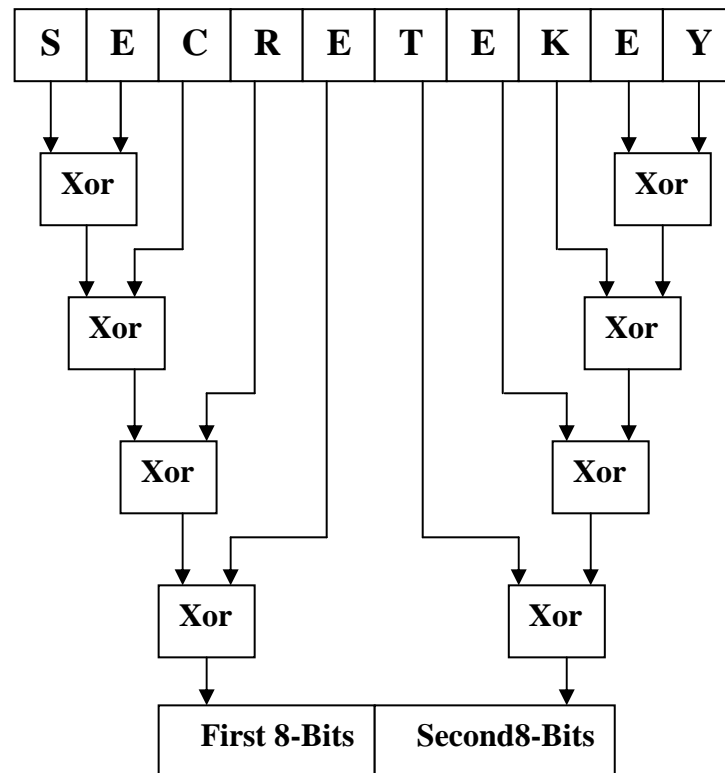


Figure (3.3) The Generation of First state "Galois Method"

Algorithm (3.4) shows the generation of first state of Galois encryption method.

Algorithm (3.4) The generation of first state of Galois encryption method.

Input: 10 characters as secret key.

Output: 16 Bits represents the first state of register.

Procedure

```
co=1;
t1=Asc(sec_key,i,1);
t3=Asc(sec_key,10,1);
For i=1 to 4
    t2=Asc(sec_key,i+1,1);
    t1=t1 Xor t2;
    t4=Asc(sec_key,10-i,1);
    t3=t3 Xor t4;
Next i
B1= t1;    //first byte.
B2=t3;    //Second byte.
```

3.8.2 First Degree of Complexity (Galois Method)

- **Stage1:** Galois encryption method will work on the slices belong to I pictures in the file, so the system must prepare these slices for encryption in this stage. Algorithm (3.5) represents the first degree of complexity applied on stage no.1 of the Galois encryption method.

Algorithm (3.5) Stage no.1 of the Galois encryption method first degree of complexity.

Input: 2-D array type_slice_st_ed and MPEG-1 file.

Output: 1-D array of most sensitive bytes in I pictures in order to send it to encryption procedure.

Procedure

```

i=0;
for i=1 to the number of slices in type_slice_st_ed
    if type_slice_st_ed(i,1)=1;           // I mean slice belong to I picture.
        k= type_slice_st_ed(i,2); // slice start.
        l= type_slice_st_ed(i,3); //slice end.
        co=0;
        For j=k to l
            co=co+1;
            Get 1byte from file in location j as b;
            Put b in enc_bytes(co);
        Next j
        Calling Galois procedure (explained in algorithm
        3.6 stage no. 2) to encrypt enc_bytes()
        For j=k to l
            Put 1 byte in the file in the location j
        Next j
    End if
Next i

```

- **Stage 2:** encryption of sensitive bytes belong to I pictures by using Galois encryption procedure. Algorithm (3.6) represents this stage.

Algorithm (3.6) Stage no.2 of the Galois encryption method first degree of complexity.

Input: 1-D array of sensitive bytes belong to I pictures and 16-bits as first state.

Output: The encryption of sensitive bytes.

Procedure

For i= 1 to the number of bytes in enc_bytes

For j=1 to 8

arr(14)=arr(1) Xor arr(15);

arr(13)=arr(1) Xor arr(14);

arr(11)=arr(1) Xor arr(12);

t1=arr(1);

For i=2 to 16

arr(i-1)=arr(i)

Next i

arr(16)=t1;

b(j)=t1; //where b is 8-bit

Next j

b_value = the integer value from 8-bits

enc_bytes(i)=enc_bytes(i) Xor b_value

Next i

3.8.3 Second Degree of Complexity (Galois Method)

- **Stage1:** Galois encryption method will work on the slices belong to I and P pictures in the file, so the system must prepare these slices for encryption. Algorithm (3.7) illustrates this stage.

Algorithm (3.7) Stage no.1 of the Galois encryption method**Second degree of complexity.****Input:** 2-D array `type_slice_st_ed` and MPEG-1 file.**Output:** 1-D array of most sensitive bytes in I and P pictures in order to send it to encryption procedure.**Procedure**

```

i=0;
For i=1 to the number of slices in type_slice_st_ed
  If type_slice_st_ed(i,1) =1 or 2;      // 1 mean slice belong to I
                                          // picture while 2 mean slice belong
                                          // to P picture.
    k= type_slice_st_ed(i,2); // slice start.
    l= type_slice_st_ed(i,3); //slice end.
    co=0;
    For j=k to l
      co=co+1;
      Get 1byte from file in location j as b;
      Put b in enc_bytes(co);
    Next j
    Calling encryption procedure (explained in algorithm 3.8 stage
      no. 2 ) to encrypt enc_bytes()
    For j=k to l
      Put 1 byte in the file in the location j
    Next j
  End if
Next i

```

- **Stage 2:** encryption of sensitive bytes belong to I pictures by using Galois encryption procedure. Algorithm (3.8) illustrates

stage no.2 of the Galois encryption method for the first degree of complexity.

**Algorithm (3.8) Stage no.2 of the Galois encryption method
second degree of complexity.**

Input: 1-D array of sensitive bytes belong to I and P pictures and
16-bits as first state.

Output: the encryption of sensitive bytes.

Procedure

Apply algorithm (3.6) on I and P pictures.

3.9 Fibonacci LFSRs Encryption Method

A 16-bit Fibonacci LFSR, the register cycles through the maximal number of 65535 states excluding the all-zeroes state.

3.9.1 State Generation

Figure (3.4) shows the generation of the first state (initial state) of Fibonacci LFSR encryption method to produce 16 bits from 10 characters.

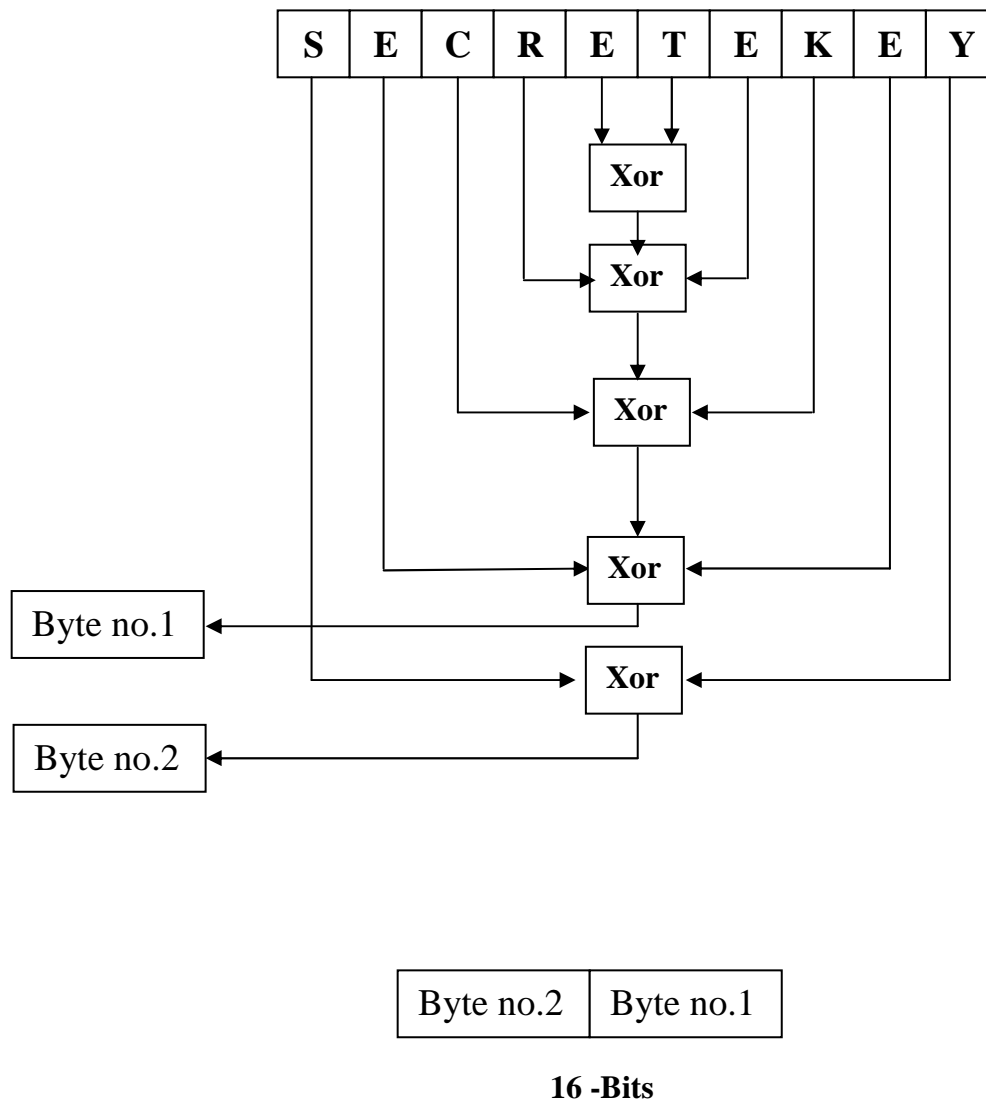


Figure (3.4) State Generation

Algorithm (3.9) shows the first state generation

Algorithm (3.9) First State Generation

Input: 10 characters as secret key.

Output: 16 Bits represents the first state of register.

Procedure

co=0;

For i=2 to 5

```
co=co+1;
t1=Asc(sec_key,i,1);
t2=Asc(sec_key,10-i+1,1);
t(co)=t1 Xor t2;
Next i
t1= Asc(sec_key,1,1);
t2=Asc(sec_key,10,1);
t3=t1 Xor t2; // Byte No. 2
For i=1 to co-1
    t(i)=t(i) Xor t(i+1) // Byte No. 1
Next i
```

3.9.2 First Degree of Complexity (Fibonacci Method)

- **Stage1:** Fibonacci LFSRs encryption method will work on the slices belong to I pictures in the file, so the system must prepare these slices for encryption in this stage. Algorithm (3.10) depicts the stage.

Algorithm (3.10) Stage no.1 of the Fibonacci LFSRs encryption method first degree of complexity.

Input: 2-D array type_slice_st_ed and MPEG-1 file.

Output: 1-D array of most sensitive bytes in I pictures in order to send it to encryption procedure.

Procedure

```

i=0;
for i=1 to the number of slices in type_slice_st_ed
    if type_slice_st_ed(i,1) =1;           // 1 mean slice belong to I picture.
        k= type_slice_st_ed(i,2); // slice start.
        l= type_slice_st_ed(i,3); //slice end.
        co=0;
        For j=k to l
            co=co+1;
            Get 1byte from file in location j as b;
            Put b in enc_bytes(co);
        Next j
        Calling Fibonacci LFSRs procedure (explained in algorithm
        3.11 stage no. 2 ) to encrypt enc_bytes()
        for j=k to l
            put 1 byte in the file in the location j
        Next j
    End if
Next i

```

- **Stage 2:** encryption of sensitive bytes belong to I pictures by using Fibonacci LFSRs encryption procedure. See algorithm (3.11).

Algorithm (3.11) Stage no.2 of the Fibonacci LFSRs encryption method first degree of complexity.

Input: 1-D array of sensitive bytes belong to I pictures and 16-bits as first state.

Output: the encryption of sensitive bytes.

Procedure

```

For i= 1 to the number of bytes in enc_bytes
  for j=1 to 8
    t1=arr(16) Xor arr(14);
    t1=t1 Xor arr(13);
    t1=t1 Xor arr(11);
    For k=15 down to 1
      arr(k+1)=arr(k)
    Next k
    arr(1)=t1;
    b(j)=t1; //where b is 8-bit
  Next j
  b_value = the byte value from 8-bit array
  enc_bytes(i)=enc_bytes(i) Xor b_value
Next i

```

3.9.3 Second Degree of Complexity (Fibonacci Method)

- **Stage1:** Fibonacci LFSRs encryption method will work on the slices belong to I and P pictures in the file as shown in algorithm (3.12), so the system must prepare these slices for encryption.

Algorithm (3.12) Stage no.1 of the Fibonacci LFSRs encryption method Second degree of complexity.

Input: 2-D array type_slice_st_ed and MPEG-1 file.

Output: 1-D array of most sensitive bytes in I and P pictures in order to send it to encryption procedure.

Procedure

```

i=0;
For i=1 to the number of slices in type_slice_st_ed
    If type_slice_st_ed(i,1) =1 or 2;    // 1 mean slice belong to I
                                        picture while 2 mean slice belong
                                        to P picture.
        k= type_slice_st_ed(i,2); // slice start.
        l= type_slice_st_ed(i,3); //slice end.
        co=0;
        For j=k to l
            co=co+1;
            Get 1byte from file in location j as b;
            Put b in enc_bytes(co);
        Next j
        Calling encryption procedure (explained in algorithm 3.13 stage
        no. 2 ) to encrypt enc_bytes()
        For j=k to l
            Put 1 byte in the file in the location j
        Next j
    End if
Next i

```

- **Stage 2:** encryption of sensitive bytes belong to I and P pictures by using Fibonacci LFSRs encryption procedure

Algorithm (3.13) Stage no.2 of the Fibonacci LFSRs encryption method Second degree of complexity.

Input: 1-D array of sensitive bytes belong to I and P pictures and 16-bits as first state.

Output: the encryption of sensitive bytes.

Procedure

Apply algorithm (3.11) on I and P pictures.

3.10 The Random Seed Values Method

After determining start and end positions for each slice with the type of picture that slice belong to, the proposed encryption system will have a 2-D array (type_slice_st_ed), an example is shown in table (3.1)

Table (3.1) an array with Picture type, start, and

Picture Type	Start position	End position
I	2650	18322
P	19106	33491

All encryption algorithms in the system will use this array before encryption, and it is very useful in the determining first degree and second degree of encryption.

- For first degree of complexity: all slices in the array (type_slice_st_ed) that belong to I pictures will be encrypted.

- For second degree of complexity: all slices in the array (type_slice_st_ed) that belong to I and P pictures will be encrypted.

In the Random Seed Values method, a secret key of 10 characters were used to generate 5 seeds values as shown in figure (3.6). Algorithm (3.14) illustrates the generating of seed values.

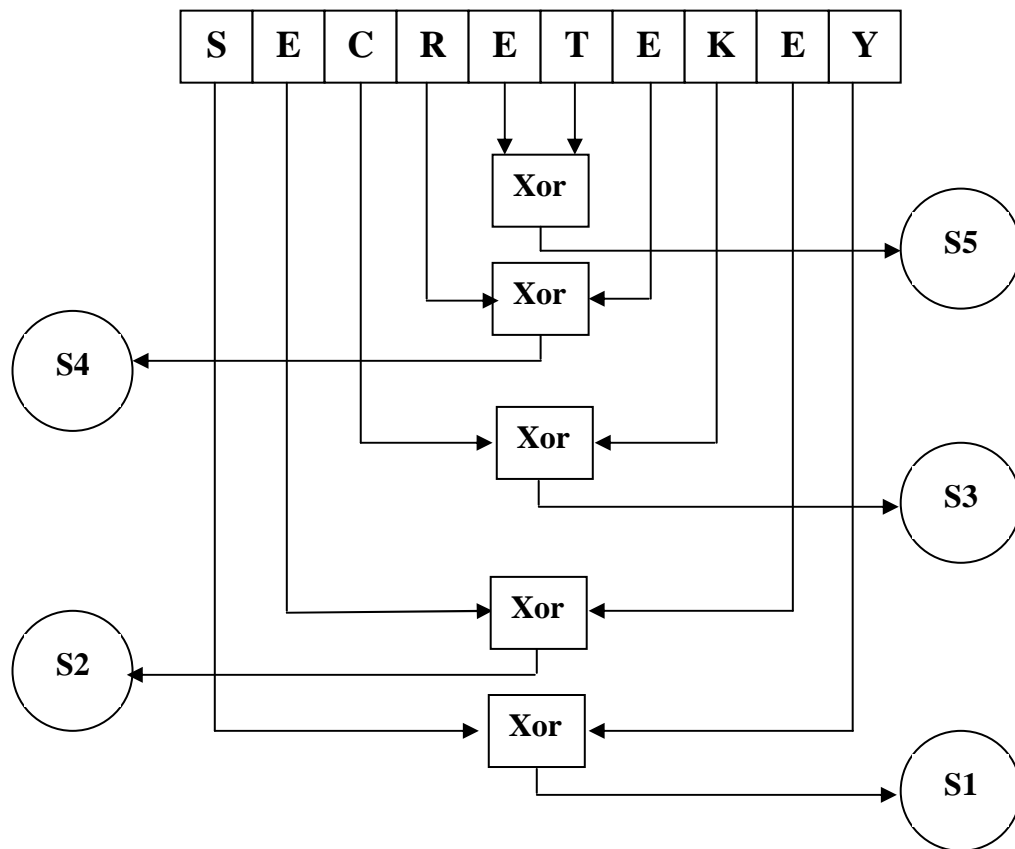


Figure (3.6) Generating seed values

These seeds values are used to generate a random numbers, Visual Basic Ver.6 random generator were used in the system, see algorithm (3.15). From each seed value, there are N random numbers where N is a prime number as:

- R1 (241) for seed no. 1.
- R2 (151) for seed no. 2.

- R3 (253) for seed no. 3.
- R4 (157) for seed no. 4.
- R5 (251) for seed no. 5.

The prime number will reduce the probability of repeating same numbers in the iterations.

Algorithm (3.14) Generating Seed Values

Input: 10 characters as secret key.

Output: 5 Seed values.

Procedure

```

For i=1 to 5
    T1 = asc(sec_key, i, 1);
    T2= asc(sec_key, 10-i+1, 1);
    T3= T1 Xor T2;
    Seed(i)=T3;
Next i

```

Algorithm (3.15) Generating random numbers from seed values

Input: 1-D array of seeds values.

Output: 5 arrays represent random numbers for each seed value.

Procedure

```

Randomize (seed(1))
For i =1 to 241:  r1(i)=rnd *255 : Next i
Randomize (seed(2))

For i =1 to 151:  r2(i)=rnd *255 : Next i

```

```

Randomize (seed(3))
For i =1 to 253:  r3(i)=rnd *255: Next i
Randomize (seed(4))
For i =1 to 157:  r4(i)=rnd *255: Next i
Randomize (seed(5))
For i =1 to 251:  r5(i)=rnd *255: Next i

```

3.10.1 First Degree of Complexity (Proposed Method)

- **Stage1:** All slices belong to I pictures will prepare for encryption.

Algorithm (3.16) illustrates stage no.1 1st complexity of the proposed method.

Algorithm (3.16) Stage no.1 of the proposed encryption method first degree of complexity

Input: 2-D array type_slice_st_ed and MPEG-1 file.

Output: 1-D array of most sensitive bytes in I pictures in order to send it to encryption procedure.

Procedure

```

i=0;
For i=1 to the number of slices in type_slice_st_ed
    If type_slice_st_ed(i,1) =1;    // 1 mean slice belong to I picture.
        k= type_slice_st_ed(i,2); // slice start.
        l= type_slice_st_ed(i,3); //slice end.
        co=0;
        For j=k to l
            co=co+1;
            Get 1byte from file in location j as b;

            Put b in enc_bytes(co);

```

```

        Next j
    Calling encryption procedure (algorithm 3.17) applied on I
    pictures to encrypt enc_bytes()
    For j=k to l
        Put 1 byte in the file in the location j
    Next j
    End if
Next i

```

- **Stage 2:** Encryption of sensitive bytes.

**Algorithm (3.17) Stage no.2 of the proposed encryption method
first degree of complexity**

Input: 1-D array of sensitive bytes and 5 arrays of 5 Seed values
come from I pictures only.

Output: the encryption of 1-D array.

Procedure

```

i1=0: i2=0: i3=0: i4=0: i5=0;
For i= 1 to the number of bytes in enc_bytes
    enc_bytes(i)= r1(i1) Xor r2(i2) Xor r3(i3) Xor r4(i4) Xor r5(i5)
    Xor enc_bytes(i);
    i1= (i1+1) mod 241;
    i2= (i2+1) mod 151;
    i3= (i3+1) mod 253;
    i4= (i4+1) mod 157;
    i5=i5+1 mod 251;
Next i

```

3.10.2 Second Degree of Complexity (Random Seed Values Method)

- *Stage1*: All slices belong to I and P pictures will prepare for encryption. Algorithm (3.18) depicts the 2nd degree of complexity.

**Algorithm (3.18) Stage no.1 of the Random Seed Values method
second degree of complexity.**

Input: 2-D array type_slice_st_ed and MPEG-1 file.

Output : 1-D array of most sensitive bytes in I and P pictures in order to send it to encryption procedure.

Procedure

```

i=0;
for i=1 to the number of slices in type_slice_st_ed
  If type_slice_st_ed(i,1) =1 or 2;           // 1 mean slice belong to I
                                              picture while 2 mean slice belong
                                              to P picture.

  k= type_slice_st_ed(i,2); // slice start.
  l= type_slice_st_ed(i,3); //slice end.
  co=0;
  For j=k to l
    co=co+1;
    Get 1byte from file in location j as b;
    Put b in enc_bytes(co);
  Next j
  Call encryption procedure (algorithm 3.19) applied on I and P pictures
  to encrypt enc_bytes()
  For j=k to l
    put 1 byte in the file in the location j
  Next j
Next i

```


- **Stage 2:** Encryption of sensitive bytes. A second degree of complexity will be implemented by applying algorithm (3.19) on I and P pictures.

**Algorithm (3.19) Stage no.2 of the proposed encryption method
second degree of complexity**

Input: 1-D array of sensitive bytes and 5 arrays of 5 Seed values
come
from I and P pictures.

Output: the encryption of 1-D array.

Procedure

Apply algorithm (3.17) on I and P pictures.

Chapter 2

Theoretical Background

2.1 Introduction

With the development of computer technology and Internet technology, multimedia data especially video data are used more and more widely. Some sensitive videos about business, military or politics often require to be protected before transmission, which can be realized by data encryption algorithms. Various data encryption algorithms have been proposed and widely used, such as Data Encryption Standard (DES), Ron Rivest, Adi Shamir, and Leonard Adleman (RSA), International Data Encryption Algorithm (IDEA) and Advanced Encryption Standard (AES) etc. But most of them are used to encrypt text or binary data, which are often of high computational complexity. They are difficult to be used into video encryption directly, for video data are often of large-volume and require real-time operation, such as picture displaying, time seeking, frame cutting or bit-rate control and so on [Shi04].

2.2 Selective Encryption

Selective encryption is a technique that is used to minimize computational complexity or enable system functionality by only encrypting a portion of a compressed bitstream while still achieving reasonable security. Selective encryption needs to rely not only on the beneficial effects of redundancy reduction, but also on the characteristics of the compression algorithm to concentrate important data representing the source in a relatively small fraction of the compressed bitstream [Liu04].

These important elements of the compressed data become candidates for selective encryption.

Selective encryption is at odds with the typical layered concept of encryption in which source processing of any type (such as compression) is applied independently of successive processing by an encryptor (with the operations reversed at a decoder). As such, it bears a special burden of showing sufficient performance improvement in target applications to warrant foregoing the substantial system design and complexity management advantages of layering. It has been proposed in a number of specific applications, especially where it can

- Create an opportunity to efficiently add multiple different encryption systems to the same bitstream by only encrypting a fraction of the data under each encryption system, while sending the remaining information in the clear (this is the focus of an important commercial application of selective encryption to MPEG-2).
- Permit different ways to organize data; for example caching large amounts of in the clear data close to users while providing the remaining necessary portions from a distant but more secure site at the time of use.
- Provide a method for efficiently making a low quality version of a bitstream that can be viewed by all while the full quality version is reserved for those who pay (a variation on the selective encryption strategy in which the encrypted data is not that required to view even a poor reproduction, but rather the additional data required to view a good reproduction) [Tom04].

2.3 Types of Video Encryption Algorithms

Many video encryption algorithms have been proposed, which can be classified into three types as shown below [Shi04].

1. **Complete-encryption algorithms:** It encrypts raw data or compressed data directly with traditional cryptosystems. Among them, some ones encrypt raw data directly. Some ones encrypt compressed data directly. These complete encryption algorithms are often of high security that benefits from the traditional cryptographies. But they are also of high computational complexity and change file format. So they are more suitable for secure video storing than for transmission.
2. **Selective encryption algorithms:** It encrypts video data partially or selectively. Among them, some ones encrypt signs of discrete cosine transform (DCT) coefficients or motion vectors. In these schemes, only signs of DCT coefficients or motion vectors are encrypted. Some algorithms propose to confuse DCT coefficients completely or partially, but they are not secure against known-plaintext attacks. So the algorithms combining coefficient confusion with sign encryption are preferred. These algorithms often satisfy real-time requirement and keep file format unchanged. However, they often change compression ratio greatly, They change the statistical characteristics of DCT coefficients. Thus, they are more suitable for real-time applications such as video transmission or video access, than for video storing.
3. **Compression-encryption combined algorithms:** That realizes compression process and encryption process at the same time. It done by controlling the parameters of entropy encoding.

2.4 Lossless and Lossy data Compression

Lossless compression techniques provide the guarantee that no pixel difference between the original and the decompressed image, i.e. lossless schemes result in the reconstructed data that exactly matches the original.

Lossy compression is techniques to remove as much information from a given set of coded data where the impact of the quality of the decoded representation is the least. Or, in simpler words: Lossy compression removes things we can't see or hear (for video and audio compression). Example of lossy compression is the (JPEG) image format (*.jpg or *.jpeg) and the audio compression scheme ATRAC Adaptive Transform Acoustic Coding (ATRAC) and MPEG video files. [Dhi05].

2.5 Image Compression

Is the application of data compression on digital images, in effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. Figure (2.1) shows the most popular lossy and lossless methods [add00]

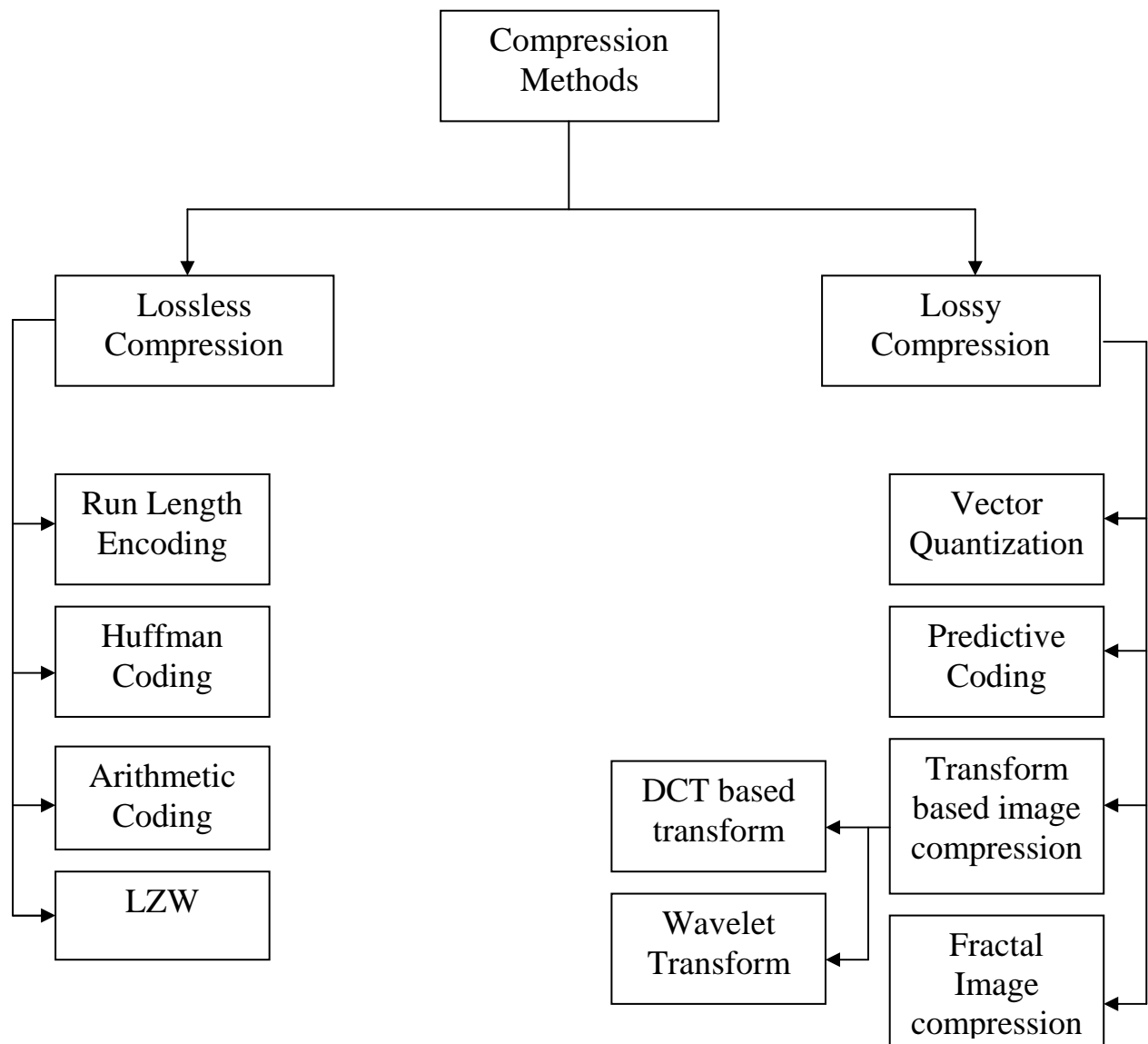


Figure (2.1): The Most popular Image Compression Methods

2.6 Video Compression

Even with powerful computer systems (storage, processor power, network bandwidth), such data amount cause extreme high computational demands for managing the data. Fortunately, digital video contains a great deal of redundancy. Thus it is suitable for compression, which can reduce these problems significantly. Especially lossy compression techniques deliver high compression ratios for video data. However, one

must keep in mind that there is always a trade-off between data size (therefore computational time) and quality. The higher the compression ratio, the lower the size and the lower the quality. The encoding and decoding process itself also needs computational resources, which have to be taken into consideration. It makes no sense, for example for a real-time application with low bandwidth requirements, to compress the video with a computational expensive algorithm which takes too long to encode and decode the data.

Compressed video may help to make the path ahead look much clearer. If digital video signals could be processed in such a way as to enable them to be economically recorded on computer hard disks without any apparent loss of quality, then the possibilities for editing, painting, and animating would seem endless. Also, if digital video could be squeezed into the same bandwidth as that occupied by conventional analog signals, viewers could receive guaranteed studio quality pictures at home. All of this and more is made possible using compression techniques. [Dhi05]

2.7 Image and Video Compression Techniques

The following compression standards are the most known nowadays. Each of them is suited for specific applications. In table (2.1) the top entry is the lowest and last row is the most recent standard. The MPEG standards are the most widely used ones, which will be explained in more details in the following sections [Nic03].

Table (2.1) Popular Image and Video Standards

Standards	Applications	Bit Rate
JPEG	Still image compression.	Variable.
H.261	Video conferencing over ISDN.	P x 64 kb/s.
MPEG-1	Video on digital storage media (CD-ROM.)	1.5Mb/s.
MPEG-2	Digital Television.	2-20 Mb/s.
H.263	Video telephony over PSTN.	33.6-? kb/s.
MPEG-4	Object-based coding, synthetic content, Interactivity	Variable.
JPEG-2000	Improved still image compression	Variable.
H.264/ MPEG-4 AVC	Improved video compression	10's to 100's kb/s.

2.8 MPEG-1 Standard

The video compression technique developed by MPEG-1 covers many applications from interactive systems on CD-ROM to the delivery of video over telecommunications networks at 1.5 Mb/sec. The MPEG-1 video coding standard is thought to be generic. To support the wide range of applications profiles a diversity of input parameters including flexible picture size and frame rate can be specified by the user. The standard video input consists of a non-interlaced video picture format. It should be noted that by no means the application of MPEG-1 is limited to this constrained parameter set [Tho02].

The algorithms employed by MPEG-1 do not provide a lossless coding scheme, the main purpose of MPEG-1 video is to code moving image sequences or video signals. To achieve a high compression ratio, both intraframe redundancy and interframe redundancy should be exploited. This implies that it would not be efficient to code the video signal with an intraframe coding scheme, such as JPEG. On the other hand, to satisfy the requirement of random access, we have to use intraframe coding from time to time. Therefore, the MPEG-1 video algorithm is mainly based on DCT coding and interframe motion compensation.

The DCT coding is used to remove the intraframe redundancy and motion compensation is used to remove interframe redundancy. With regard to input picture format, MPEG-1 allows progressive pictures only, but offers great flexibility in the size, up to 4095 x 4095 pixels. However, the coder itself is optimized to the extensively used video SIF picture format. The SIF is a simple derivative of the CCIR601 video format for digital television applications. According to CCIR601, a color video source has three components, a luminance component (Y) and two chrominance components (C_b and C_r) which are in the 4:2:0 subsampling format.

The MPEG coding algorithm is a full-motion-compensated DCT and Differential Pulse Code Modulation (DPCM) hybrid coding algorithm [**Jas04**].

2.8.1 The Internal Structure of MPEG-1

Figure (2.2) illustrate the internal structure of MPEG-1 [Dre03].

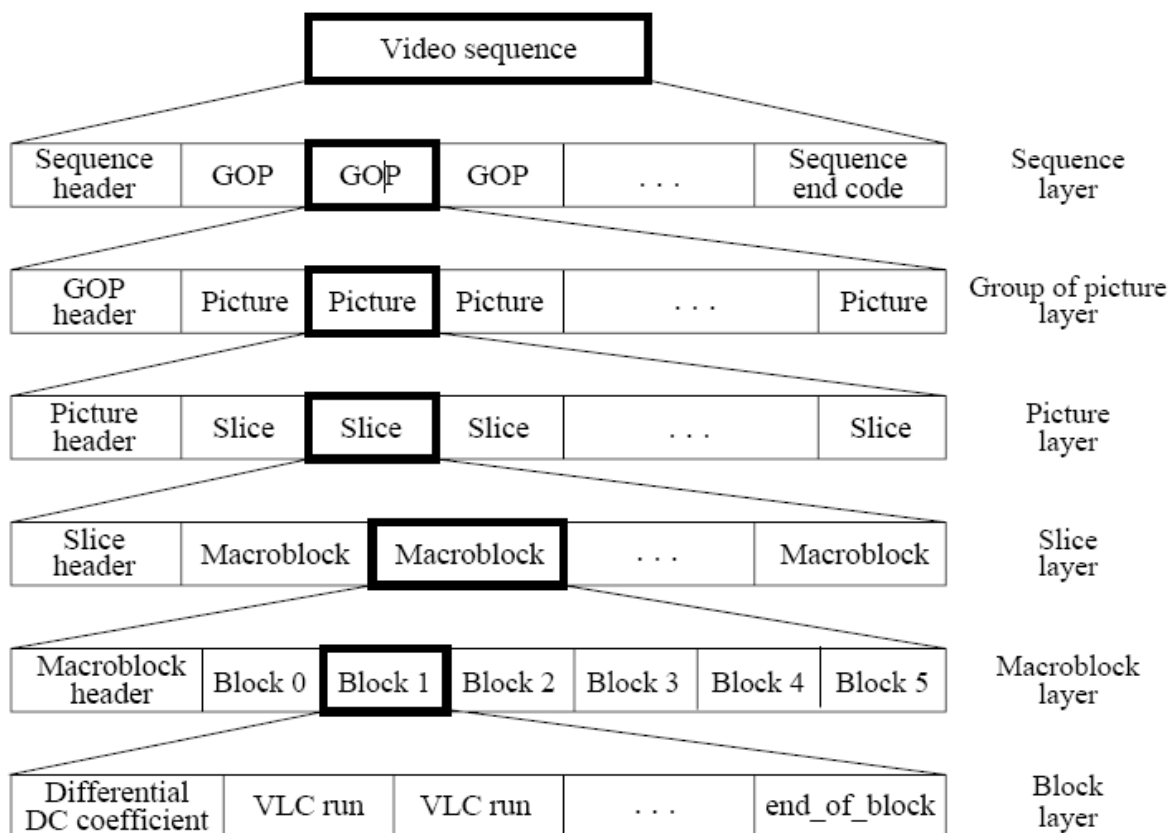


Figure (2.2) MPEG-1 structures

2.8.2 Group of Pictures (GOP)

Each GOP may include three types of pictures or frames: intracoded (I) picture or frame, predictive-coded (P) picture or frame, and bidirectionally predictive-coded (B) picture or frame.

I-pictures are coded by intraframe techniques only, with no need for previous information, in other words, I-pictures are self-sufficient. they are used as anchors for forward and/or backward prediction.

P-pictures are coded using one-directional motion-compensated prediction from a previous anchor frame, which could be either an I or a P-picture. The distance between two nearest I-frames is the size of GOP and denoted by N and the distance between two P frame denoted by M , A larger number of N and M will increase the coding performance but cause error propagation or drift.

(Figure 2.3) shows the GOP and forward motion compensation and backward motion compensations [Joh04].

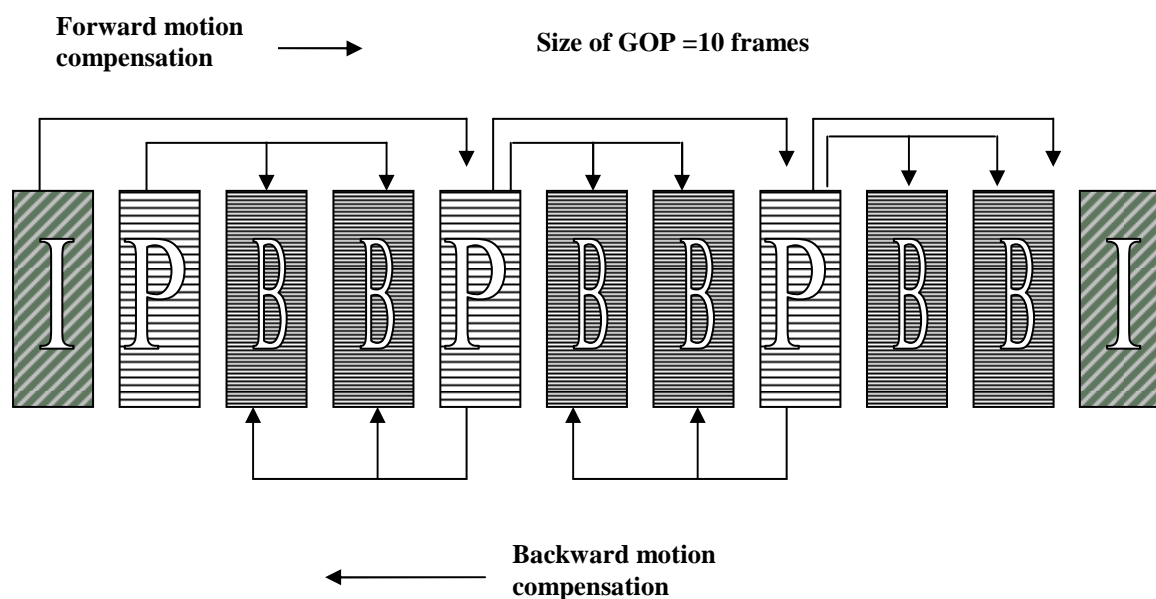


Figure (2.3) GOP

Usually, N is chosen from 12 to 15 and M from 1 to 3.

2.8.3 Slices

Regardless of the type of picture, each one may be divided into slices; each slice consists of several macroblocks (MBs). There is no rule to decide the slice size. A slice could contain all macroblocks in a row of a frame or all macroblocks of a frame. Smaller slice size is favorable for the purpose of error resilience, but will decrease coding performance due

to higher overhead, Figure (2.4) shows a number of slices in the picture [Dre03].

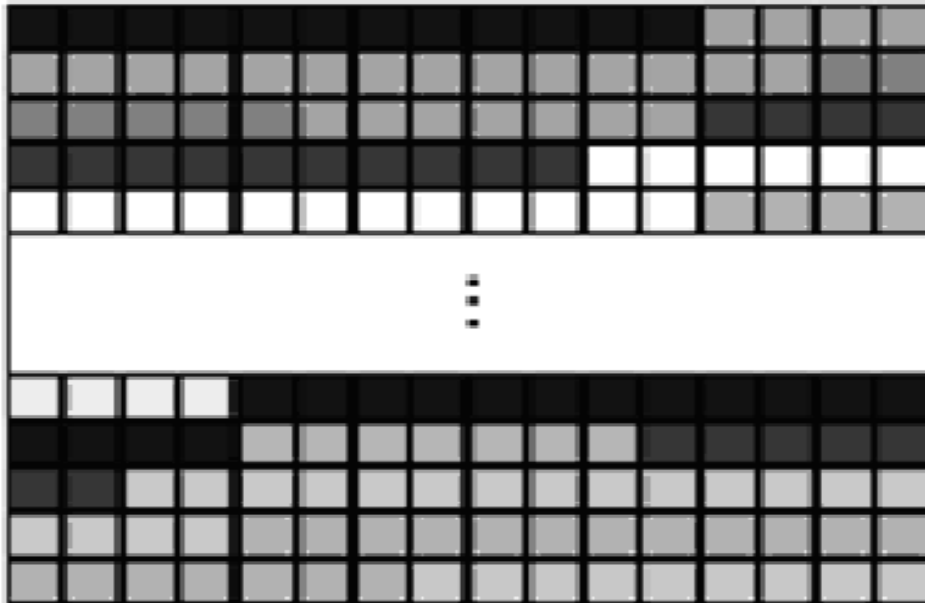


Figure 2.4 Slices in the picture

2.8.4 Macroblock

A macroblock contains a 16×16 Y component and spatially corresponding 8×8 C_b and C_r components. A macroblock has four luminance blocks and two chrominance blocks (for 4:2:0 sampling) and the macroblock is also the basic unit of adaptive quantization and motion compensation. Each block contains 8×8 pixels over which the DCT operation is performed, to exploit the temporal redundancy in the video sequence, the motion vector for each macroblock is estimated from two original luminance pictures using a block-matching algorithm. The criterion for the best match between the current macroblock and a macroblock in the anchor frame is the minimum mean absolute error. Once the motion vector for each macroblock is estimated, pixel values for

the target macroblock can be predicted from the previously decoded frame. All macroblocks in the I-frame are coded in intramode with no motion compensation. Macroblocks in P- and B-frames can be coded in several modes. Among the modes are intracoded and intercoded with motion compensation. This decision is made by mode selection. Most encoders depend on values of predicted differences to make this decision. Within each slice, the values of motion vectors and DCT values of each macroblock are coded using DPCM [Tho02].

2.8.5 Reordering Pictures in MPEG-1:

The encoding order is different from the display order in MPEG-1 pictures, the input sequence has to be reordered for encoding. For example, the GOP size =10, and the distance between two nearest P frames =3, the display order and encoding order are as shown in Figure (2.5)

It should be noted that in the encoding order or in the bitstream the first frame in a GOP is always an I-picture. In the display order the first frame can be either an I-picture or the first B-picture of the consecutive series of B-pictures [Yun99].

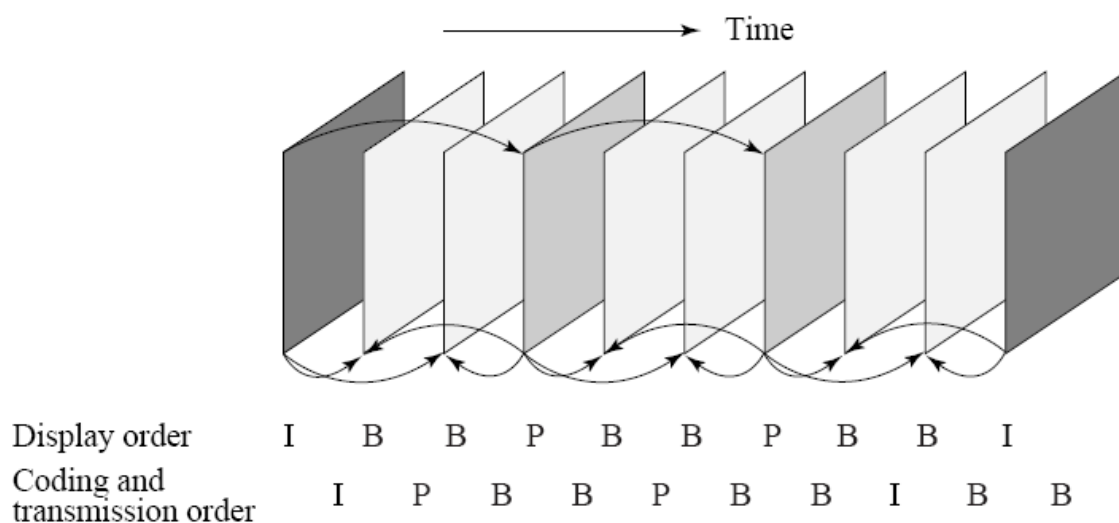


Figure (2.5) Frame Display order

2.8.6 Motion Compensation of MPEG-1

The MPEG-1 video compression technique uses motion compensation to remove the interframe redundancy. The concept of motion compensation is based on the estimation of motion between video frames. The fundamental model that is used assumes that a translational motion can approximate the motion of a block. If all elements in a video scene are approximately spatially displaced, the motion between frames' can be described by a limited number of motion parameters. In other words, the motion can be described by motion vectors for translatory motion of pixels. Since the spatial correlation between adjacent pixels is usually very high, it is not necessary to transmit motion information for each coded image pixel. This would be too expensive and the coder would never be able to reach a high compression ratio. The MPEG video uses the macroblock structure for motion compensation; i.e., for each 16

x 16 macroblock only one or sometimes two motion vectors are transmitted. The motion vectors for any block are found within a search window that can be up to 512 pixels in each direction. Also, the matching can be done at half-pixel accuracy where the half-pixel values are computed by averaging the full-pixel values. Figure (2.6) shows half and full pixel locations [Yun99].

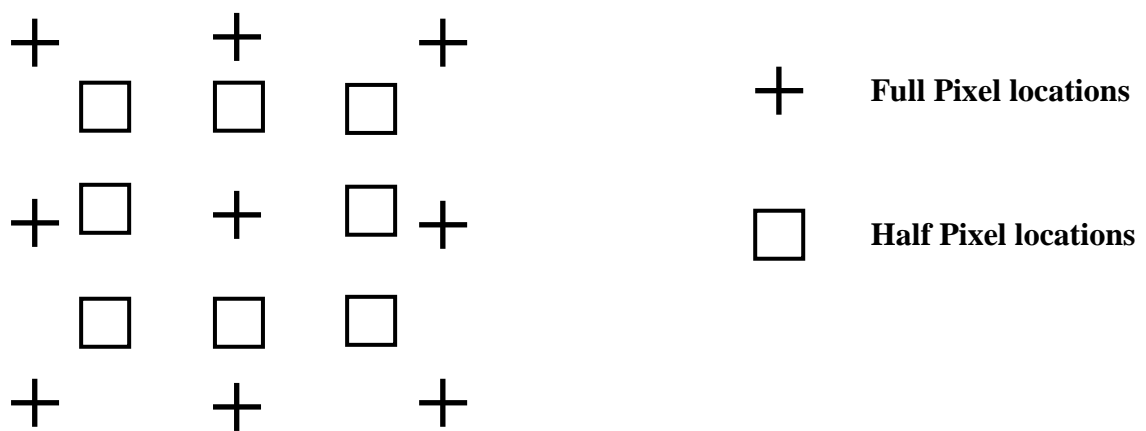


Figure 2.6 Half pixel locations

For interframe coding, the prediction differences or error images are coded and transmitted with motion information. A 2-D DCT is used for coding both the intraframe pixels and the predictive error pixels. For P frame, each macroblock (MB) of the target P-frame is assigned a best matching MB from the previously coded I or P frames (figure 2.7).

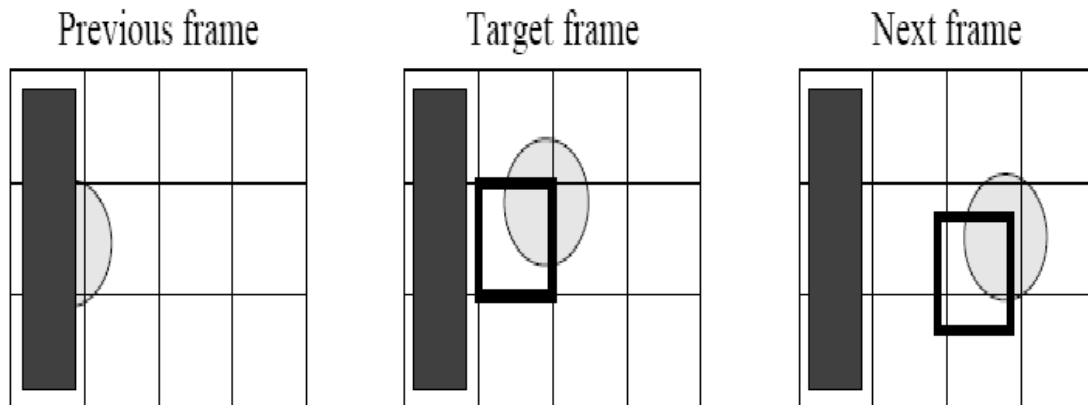


Figure (2.7) Prediction belong to P frame

For B frames, each MB from a B-frame will have up to two motion vectors (MVs) (one from the forward and one from the backward prediction), If matching in both directions is successful, then two MVs will be sent and the two corresponding matching MBs are averaged (indicated by %) before comparing to the target MB for generating the prediction error, If an acceptable match can be found in only one of the reference frames, then only one MV and its corresponding MB will be used from either the forward or backward prediction as shown in figure (2.8) [Dre03].

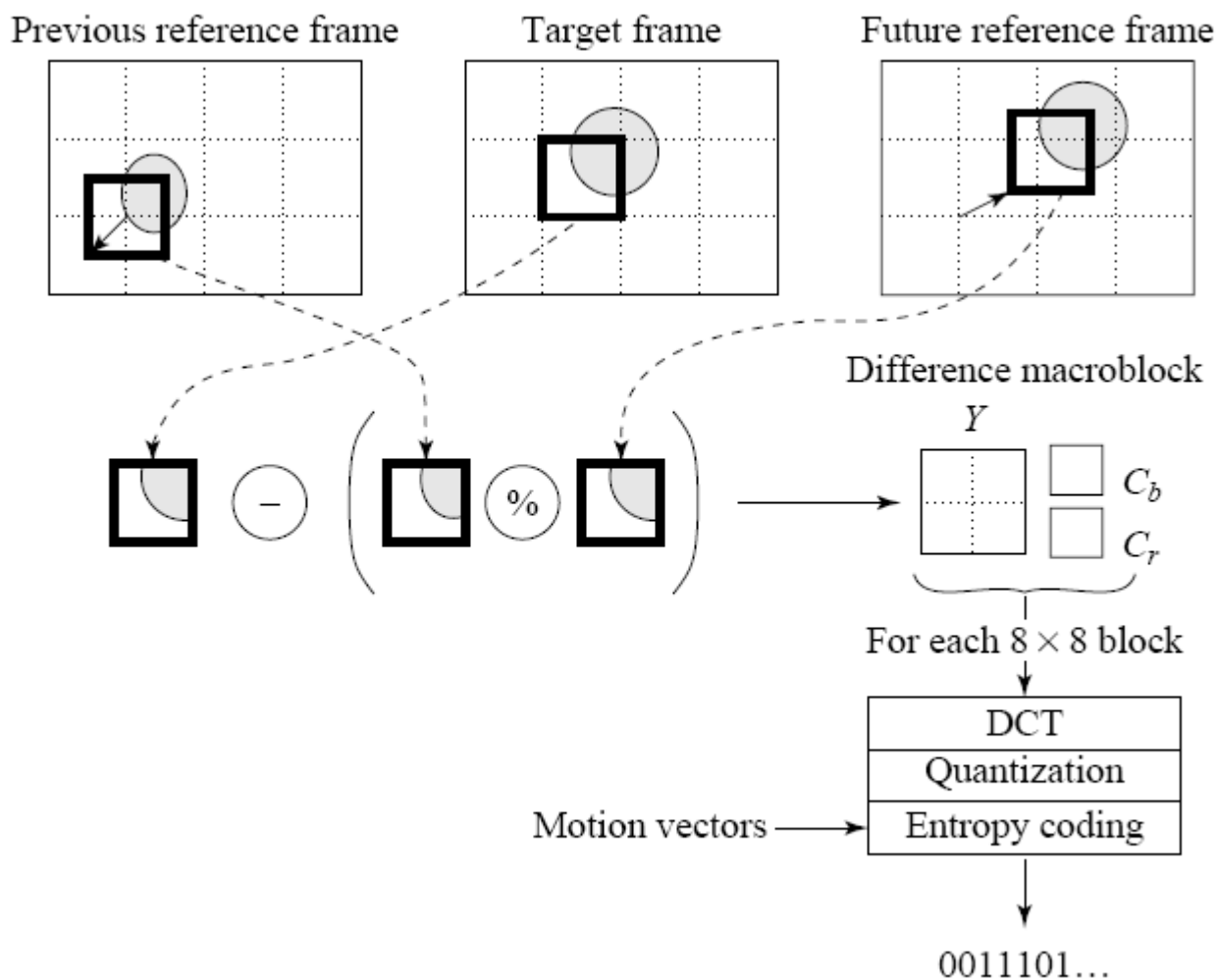


Figure (2.8) prediction belong to B frame

Three steps in motion compensation are listed below in more details:

- **Frame Segmentation:** The actual frame is divided into nonoverlapping blocks (macro blocks), the smaller the block sizes are chosen, the more vectors need to be calculated; the block size therefore is a critical factor in terms of time performance, but also in terms of quality: if the blocks are too large, the motion matching is most likely less correlated. If the blocks are too small, it is probably, that the algorithm will try to match noise. MPEG uses usually block sizes of 16×16 pixels.

- **Search Threshold:** In order to minimize the number of expensive motion estimation calculations, they are only calculated if the difference between two blocks at the same position is higher than a threshold, otherwise the whole block is transmitted.
- **Block Matching:** In general block matching tries, to “stitch together” an actual predicted frame by using blocks from previous frames, the process of block matching is the most time consuming one during encoding. In order to find a matching block, each block of the current frame is compared with a past frame within a search area, only the luminance information is used to compare the blocks, but obviously the color information will be included in the encoding. The search area is a critical factor for the quality of the matching. It is more likely that the algorithm finds a matching block, if it searches a larger area. Obviously the number of search operations increases quadratically, when extending the search area. Therefore too large search areas slow down the encoding process dramatically. To reduce these problems often rectangular search areas are used, which take into account, that horizontal movements are more likely than vertical ones [Mic06].

2.8.7 The Goal Factors in Motion Compensation

Most of the research works have been concentrated on optimizing the block-based motion compensation technique. As the demand for real time video applications (like video recording, video conferencing, video phone, etc) the needs for video coding had been grown. Fast video encoding with good compression ratio as well as high signal to noise ratio is highly essential. Good compression ratio means reducing the size of the

coded video with little degradation of quality. Motion compensation is exactly a technique designed to achieve good compression ratio in video compression. However, speed and quality are often two contradicting goals. Nowadays, researchers are still actively investigating for an optimum trade-off between these two factors. Most of the motion proposed estimation algorithms tends to bias toward speed by sacrificing visual quality [Moh99].

2.9 Fibonacci LFSRs Encryption Algorithm

A 16-bit Fibonacci Left Shift Register (LFSR), the register cycles through the maximal number of 65535 states excluding the all-zeroes state.

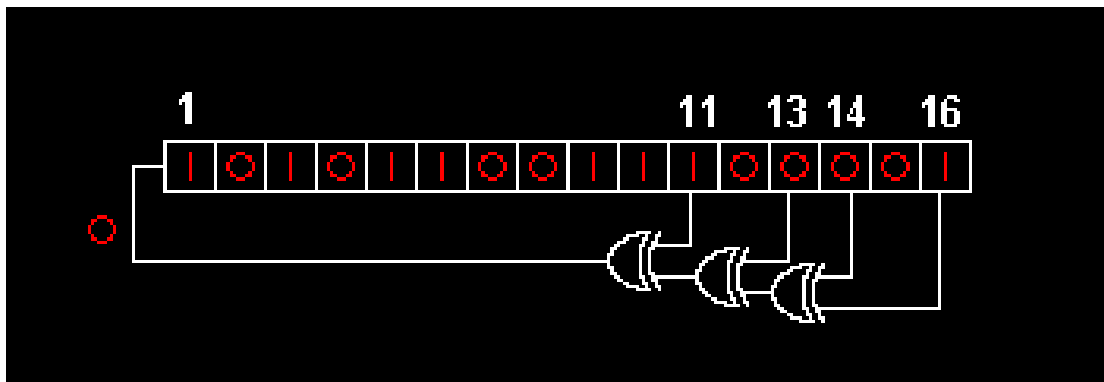


Figure (2.14) Fibonacci LFSRs

The list of the bits' positions that affect the next state is called the tap sequence, In the figure (2.15) the sequence is [16, 14, 13, 11, and 0], The taps are XOR'd sequentially with the output and then feed back into the leftmost bit, so the sequence of numbers generated by an LFSR can be considered a binary numeral system just as valid as Gray code or the

natural binary code. A maximal LFSR produces an n -sequence (i.e. cycles through all possible $2^n - 1$ states within the shift register except the state where all bits are zero), unless it contains all zeros, in which case it will never change

The tap sequence of an LFSR can be represented as a polynomial mod 2. This means that the coefficients of the polynomial must be 1's or 0's. This is called the feedback polynomial or characteristic polynomial. For example, if the taps are at the 16th, 14th, 13th and 11th bits (as shown), the feedback polynomial is:

$$X^{16} + X^{14} + X^{13} + X^{11} + 1$$

The 'one' in the polynomial does not correspond to a tap - it corresponds to the input to the first bit (i.e. x^0 , which is equivalent to 1). The powers of the terms represent the tapped bits, counting from the left. The first and last bits are always connected as an input and tap respectively. [Bru94]

Table (2.2) Feedback polynomial

Bits	Feedback polynomial	Period
n		$2^n - 1$
4	$x^4 + x^3 + 1$	15
5	$x^5 + x^3 + 1$	31
6	$x^6 + x^5 + 1$	63
7	$x^7 + x^6 + 1$	127
8	$x^8 + x^6 + x^5 + x^4 + 1$	255
9	$x^9 + x^5 + 1$	511
10	$x^{10} + x^7 + 1$	1023
11	$x^{11} + x^9 + 1$	2047
12	$x^{12} + x^{11} + x^{10} + x^4 + 1$	4095
13	$x^{13} + x^{12} + x^{11} + x^8 + 1$	8191
14	$x^{14} + x^{13} + x^{12} + x^2 + 1$	16383
15	$x^{15} + x^{14} + 1$	32767
16	$x^{16} + x^{14} + x^{13} + x^{11} + 1$	65535

It should be noted that LFSRs can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence, such as direct-sequence spread spectrum radio. It can generate an extremely long sequence which can be used to encrypt valuable information, for example by combining bits from the LFSR with information bits in an XOR gate. The resulting bit stream can only be decrypted by a receiver equipped with the same LFSR, starting at the same state. Although an attacker might know or discover the LFSR construction, as long as the starting state is kept as a secret key the attacker confronts a monstrous search problem.

2.10 Galois Encryption Algorithm

Is an alternate structure that can generate the same output sequences as a conventional LFSR. In the Galois configuration, when the system is clocked, bits that are not taps are shifted as normal to the next flip-flop. The taps, on the other hand, are XOR'd with the new output, which also becomes the new input. These won't be shifted in until the next clock cycle.

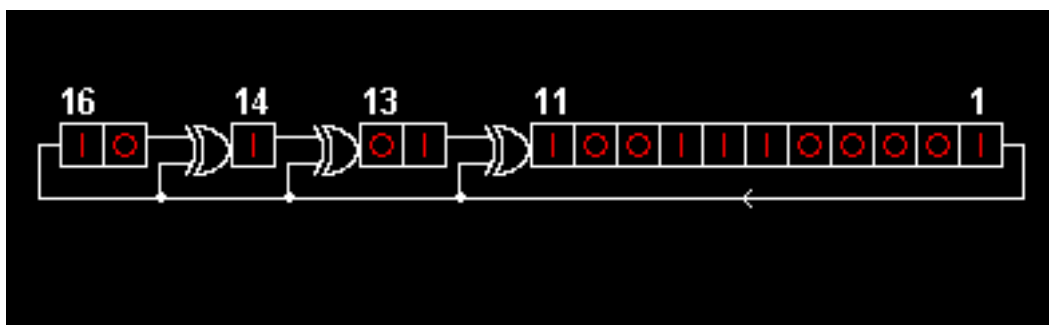


Figure (2.15) Galois Method

Galois LFSRs do not concatenate every tap to produce the new input (the XOR'ing is done within the LFSR and no XOR gates are run in serial, therefore the propagation times are reduced to that of one XOR rather than a whole chain), thus it is possible for each tap to be computed in parallel, increasing the speed of execution, Also In a software implementation of an LFSR, the Galois form is more efficient as the XOR operations can be implemented a word at a time: only the output bit must be examined individually.

The repeating sequence of states of an LFSR allows it to be used as a divider, or as a counter when a non-binary sequence is acceptable as is often the case where computer index or framing locations need to be machine-readable. LFSR counters have simpler feedback logic than natural binary counters or Gray code counters, and therefore can operate at higher clock rates. However it is necessary to ensure that the LFSR never enters an all-zeros state, for example by presetting it at start-up to any other state in the sequence. The table of primitive polynomials shows how LFSR's can be arranged in Fibonacci or Galois form to give maximal periods. One can obtain any other period by adding to an LFSR that has a longer period some logic that shortens the sequence by skipping some state(s), e.g. as tabulated in. [Bru94]

List of Abbreviations

<i>Abbreviations</i>	<i>Meaning</i>
AES	Advanced Encryption Standard
ATRAC	Adaptive Transform Acoustic Coding
AV	Audio and Visual data
CREA	Chaotic Run-length Encryption Algorithm
CVES	Chaotic Video Encryption Scheme
DCT	Discrete Cosine Transforms
DES	Data Encryption Standard
DPCM	Differential Pulse Code Modulation
FPS	Frame Per Second
GOP	Group Of Pictures
HDTV	High-Definition TeleVision
IDEA	International Data Encryption Algorithm
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
JPEG	Joint Pictures Expert Group
JTC	Joint Technical Committee
LFSR	Left Shift Register
MB	MacroBlocks
MPEG	Moving Picture Expert Group
MSE	Mean Square Error
MV	Motion Vector
NTSC	National Television System Committee
PAL	Phase Alternating Line

PSNR	Peak to Signal Noise Ratio
RGB	Read Green Blue
SECSC	Security-Enhanced Chaotic Stream Cipher
SIF	Source Input Format
VCR	Video Cassette Recorder
VLC	Variable Length Code
XML	Extensible Markup Language

- [Add00] Addel j., "*lossy and lossless Image compression*", Prentice Hall PTR, 2000.
- [Bru94] Bruce Schneier and John Wiley "*Applied Cryptography*", New York, 2nd edition, 1994.
- [Chr05] Christof Paar, "*Applied Cryptography and Data Security*", Department of Electrical Engineering and Information Sciences, University of Ruhr, Germany, 2005.
- [Chu01] Chun Yuan, Bin B. Zhu and et.al, "*Efficient and Fully Scalable Encryption for MPEG-4 FGS*", Dept. of Computer Science, University of Beijing, China, 2001.
- [Dhi05] Dhiah E. Al-Shammary, "*Interframe Compression Using Distributed Systems*", MSc. thesis, Nahrian University, 2005.
- [Den07] Denz Taskin, Cem Taskin and et.al, "*Selective Encryption of Compressed Video Files*", International Scientific Conference, Computer Engineering Department, Trakya University, 2007.
- [Djo07] Djordje Mitrovic, "*Video Compression*", University of Edinburgh, 2007.
- [Dre03] Drew and Li. "*Fundamentals of Multimedia*", Prentice Hall PTR, 2003.
- [Jas04] Jason But, "*Limitation of Existing MPEG-1 Ciphers for Streaming Video*", Technical Report 040429A, Swinburne University of Technology Australia, 2004.
- [Joh04] John G. Apostolopoulos, "*Video Compression*", Streaming Media Systems Group, Hewlett-Packard Laboratories, 2004.

- [Liu04] Liu Jun, Zou LingLing and et.al, "*A Two-Way Selective Encryption Algorithm for MPEG Video*", Huazhong University of Science and Technology, 2004.
- [Men96] A. Menezes, P.Van Oorschot and et.al, "*Handbook of Applied Cryptography*", CRC Press, 1996.
- [Mic06] Michael Roitzsch and Martin Pohlack, "*Principles for the Prediction of Video Decoding Times Applied to MPEG-1/2 and MPEG-4 Part 2 Video*", Operating Systems Group, Department of Computer Science, University of Dresden, 2006.
- [Moh99] Mohamed Alkanhal, "*Correlation Based Search Algorithms for Motion Estimation*", Carnegie Mellon University, 1999.
- [Nic03] Nicorsin F., "*Video Compression Technologies*", NIC INC, 2003.
- [Ral95] Ralf Schafer and Thomas Shikora, "*Digital Video Coding Standards and Their Role*", IEEE, Vol. 83, No.6, p.907 1995.
- [Sco98] Scot E. Umbaugh, "*Computer Vision and Image Processing*", PhD., Penetric Hall PTR, 1998.
- [Sha96] Shanawaz Basith, "*Mpeg Standards Technology and Applications*", Information Systems Engineering, Department of Computing and Department of Electrical and Electronic Engineering, university of Soehtia, 1996.
- [Shi04] Shiguo Lian, Jinsheng Sun and et.al, "*A Fast Video Encryption Scheme Based-on Chaos*", Department of

- Automation, University of Nanjing, P.R China 210094, IEEE, 0-7803-8653-1, P.126, 2004.
- [Shu07] Shujun Li, Guanrong Chen and et.al, "*On the Design of Perceptual MPEG Video Encryption Algorithms*", IEEE, Vol.17, No.2, P.214 2007.
 - [Shu02] Shujun Li, Xuan Zheng and et.al, "*Chaotic Encryption Scheme for Real-Time Digital Video*", SPIE, vol. 4666, P. 149-160, Real-Time Imaging VI, China, 2002.
 - [Tho02] Thomas Sikora, "*MPEG Digital Video Coding Standard*", First Edition, McGRAW-Hill Book Company, 2002.
 - [Tom04] Tom Lookabaugh, "*Selective Encryption, Information Theory and Compression*", Computer Science Department, University of Colorado, 2004.
 - [Yan99] Yan Shu. B., "*Coding of Motion Picture Without Spatial Transforms*", MSc. thesis, University of Newfoundland, 1999.
 - [Yun99] Yun Q. Shi and Huifang Sun, "*Image and Video Compression for Multimedia Engineering*", New Jersey Institute of Technology, 1999.

Web:

- [Web1] The Official Site of MPEG <http://mpeg.org>.
- [Web2] Wikipedia encyclopedia, <http://en.wikipedia.org/wiki/Mpeg>.

Table of Contents

Acknowledgment	I
Abstract	II
Table of content	III
List of Abbreviations	VI
<i>Chapter One " Introduction "</i>	1
1.1 Digital Video	1
1.2 Encryption	1
1.3 General Encryption Approaches	2
1.4 Video Encryption	2
1.5 Moving Picture Experts Group	3
1.6 Job of MPEG Digital video	5
1.7 Literature survey	5
1.8 Aim of Thesis	8
1.9 Thesis Layout	8
<i>Chapter Two "Theoretical Background"</i>	9
2.1 Introduction	9
2.2 Selective Encryption	9
2.3 Types of Video Encryption Algorithms	11
2.4 Lossless and Lossy Data Compression	12
2.5 Image Compression	12
2.6 Video Compression	13
2.7 Image and Video Compression Techniques	14
2.8 Mpeg-1 Standard	15
2.8.1 The Internal Structure of MPEG-1	17
2.8.2 Group of Pictures (GOP)	17
2.8.3 Slices	18

2.8.4 Macroblock	19
2.8.5 Reordering Pictures in MPEG-1	20
2.8.6 Motion Compensation of MPEG-1	21
2.8.7 The Goal Factors in Motion Compensation	25
2.9 Fibonacci LFSRs Encryption Algorithm	26
2.10 Galois Encryption Algorithm	۲۸
<i>Chapter Three "Encryption System"</i>	۳۰
3.1 Introduction	۳۰
3.2 Media File	۳۲
3.3 The Proposed Encryption System	۳۲
3.4 Encryption System structure	۳۲
3.5 Parsing of MPEG-1 Video File	۳۴
3.6 Determining GOP's, pictures start positions and relative parameters	۳۵
3.7 Determining slices for each picture	۳۷
3.8 Galois Encryption Method	۳۹
3.8.1 Generation of state	40
3.8.2 First Degree of Complexity (Galois Method)	41
3.8.3 Second Degree of Complexity (Galois Method)	43
3.9 Fibonacci LFSRs Encryption Method	45
3.9.1 State Generation	45
3.9.2 First Degree of Complexity (Fibonacci Method)	۴۷
3.9.3 Second Degree of Complexity (Fibonacci Method)	۴۹
3.10 Random Seed Values Method	51
3.10.1 First Degree of Complexity (Random Seed Values Method)	۵۴
3.10.2 Second Degree of Complexity (Random Seed Values Method)	۵۶

<i>Chapter Four "Tests and Results"</i>	๕๗
4.1 Introduction	๕๗
4.2 Fidelity Criteria	๕๗
4.3 Performance Parameters	๕๙
4.4 Samples Specification	๖๐
4.5 Testing Strategy	๖๒
4.6 Brute Force	๖๗
4.7 Subjective samples testing	๖๗
<i>Chapter Five "Conclusions and Future Work"</i>	๗๔
5.1 Introduction	๗๔
5.2 Conclusions	๗๔
5.3 Future Work	๗๕
<i>References</i>	๗๖
<i>Appendices</i>	

الخلاصة

هنالك نوع حديث من التشفير يدعى بالتشفير الانتقائي الذي يعتمد على اختيار بعض المناطق ذات الحساسية العالية وتشفيرها بصورة تؤدي إلى تدمير الملف بصورة كاملة، هذا النوع من التشفير يتلائم أكثر مع الملفات المضغوطة مثل الملفات الفيديوية من نوع (MPEG) والصور المضغوطة من نوع (JPEG) الخ من أنواع الملفات المضغوطة. هذا التلائم ناتج بسبب أن الملفات المضغوطة تتركز فيها كمية كبيرة من المعلومات في مساحات صغيرة أكثر من الملفات غير المضغوطة.

إن الملفات الفيديوية من نوع (MPEG) تحتوي على ثلاث أنواع من الصور (المرجع I، التنبؤ الامامي P، التنبؤ الامامي/الخلفي B) داخل الملف، إن الصور الأكثر أهمية من بين هذه الأنواع الثلاثة هي الصور المرجع التي تعد كأساس تشتق منه الصور الأخرى، هذه الأنواع من الصور توجد بشكل متسلسل داخل الملف الفيديوي وعلى شكل مجاميع، ففي كل مجموعة نجد صورة واحدة من النوع المرجع وعدد مختلف من الصور ذات التنبؤ الامامي والتنبؤ الامامي/الخلفي يتراوح من (٩ - ١٥) حسب طبيعة الملف وكمية البيانات المضغوطة، لذلك فإن تشفير الصور المرجع فقط يؤدي إلى تدمير نسبة لا بأس بها من المعلومات المرئية (تشفير ذو الدرجة الاولى من التعقيد) أما تشفير الصور المرجع مع تشفير الصور ذات التنبؤ الامامي سيؤدي إلى تدمير كامل للمعلومات المرئية (تشفير ذو الدرجة الثانية من التعقيد) مع اهمال الصور ذات التنبؤ الامامي/الخلفي بدون تشفير. من المهم ذكره أن مجموع الصور المرجع والصور ذات التنبؤ الامامي داخل الملف الفيديوي هو أقل بكثير من مجموع صور التنبؤ الامامي/الخلفي فقط، وبذلك تركنا كمية كبيرة من المعلومات من دون تشفير أدى إلى تقليل الوقت المستغرق للتنفيذ.

في النظام المعتمد تم تشفير المناطق الأكثر حساسية باستخدام ثلاثة أنواع من طرق التشفير وبدرجاتي التعقيد الاولى والثانية، هذه الطرق هي طريقة فابوناشي، طريقة جاليوس واخيرا الطريقة المقترحة. فقد تم تطوير الطريقة الاخيرة لتضاهي الطرق الاخرى المعروفة لتعمل بكفاءة عالية ذات وقت تنفيذي مناسب وبدرجة تشفير في اغلب الحالات تكون هي الأعلى بالاضافة الى ان احتمالية فك الشفرة تكون قليلة جدا مقارنة مع الطريقتين الاخرتين. لقد تم تنفيذ النظام المعتمد باستخدام فيجوال بيسك ٦,٠ كلغة برمجية. تم استخدام معدل الخطأ التربيعي (MSE)، ونسبة الاشارة الى التشويش (PSNR) كمعاملات حساب دقة النتائج المستخلصة من التقنيات الكلية المطورة في النظام المعتمد.

الأسم: محمد صبحي صادق الاوسي.

التحصيل الدراسي: ماجستير علوم حاسبات.

سنة التخرج: بكالوريوس ٢٠٠٥

الجامعة: جامعة النهرين

المواليد: ٢٢-٦-١٩٨٤

العنوان: واسط - كوت - حي ١٧ - ٣٠ تموز.

عنوان السكن: محله : ١٩ ، الزقاق: ١١ ، الدار: ٤

تاريخ المناقشه : ٢٣-٦-١٩٨٤

رقم الهاتف: ٠٧٨٠١٧٨٨٣٦٦

الايمل: alaisy84@yahoo.com

اسم المشرف: المدرس الدكتورة بان نديم الكلاك

عنوان الاطروحه: تشفير الملفات الفيديوية امبيج-١

*Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science
Department of Computer*



Encryption of MPEG-1 Video Files

A Thesis

*Submitted to the College of Science Al-Nahrain University as a Partial
Fulfillment of the Requirements for the Degree of Master of Science in
Computer*

By

*Muhammed Subhi Sadiq Alausy
(B.Sc., Al-Nahrain University, 2005)*

Supervised by

Dr. Ban N. Al-Kallak

May

2008

Jamada Alawal

1429



جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النجف
كلية العلوم
قسم الحاسبات

تشفير الملفات الفيديوية امبيج-1

رسالة

مقدمة إلى كلية العلوم - جامعة النجف وهي جزء من متطلبات نيل درجة الماجستير في

علوم الحاسبات

مِن قِبَلِ

محمد صبيح صادق الأوسي

(بكالوريوس علوم, جامعة النجف, ٢٠٠٥)

بإشراف

د. بان نديم الكلك

جمادى الأول

١٤٢٩

أيار

٢٠٠٨ م

هـ