To Modify the Partial Audio Cryptography for Haar Wavelet Transform by Using AES Algorithm

Dr. Abdul Moneem S. Rahma
Computer Science Department, University of Technology/Baghdad
Email:Monem.rahma@yahoo.com
Maisaa Abid Ali k.
Computer Science Department, University of Technology/Baghdad

Received on: 22 /1/2013 & Accepted on: 3/10/2013

ABSTRACT

The rapid developments that have occurred in data security and confidentiality of information transmitted via the Internet has created the need to preserve the audio information transmitted over the network from intruders who spy on networks and Internet penetration. That led to the proposal of a new encryption algorithm for the transfer of audio data in a rapid, strong, encrypted and confidential way.

The audio data compression algorithm is integrated at the third level to transfer the wave of bilateral and advanced encryption algorithm (AES) and fed so as to obtain strong encryption algorithm.

The results obtained from proposed the algorithms are positive results. Any malicious intruder cannot penetrate the network and open the encryption and see the audio file. It is possible to return the original data of the audio file without losing any information by the recipient.

Key word: Audio File, Format File.Wav, Compression Audio, Haar Wavelet Transform, AES Algorithm.

طريقة بحث بيانات صوتية والتشفير بطريقة متقدمة واستخدام الموجه الثنائية

الخلاصة

ادت التطورات السريعة التي حصلت في امنية البيانات وسرية المعلومات المنقولة عبر شبكة الانترنيت ادت الى ضرورة الحفاظ على المعلومات الصوتية المنقولة عبر الشبكة من المتطفلين الذين يحاولون التجسس واختراق شبكات الانترنيت.

مما دفع الى اقتراح خوارزمية تشفير جديدة لنقل بيانات الصوت وبصورة سريعة ومشفره وبسرية قوية. وهي دمج خوارزمية ضغط البيانات الصوتية في المستوى الثالث بتحويل الموجة الثنائية وخوارزمية التشفير المتقدمة إ ي اس للحصول على خوارزمية تشفير قوية وذات تردد عالي.

ان النتائج التي تم الحصول عليها من الخوارزمية المقترحة نتائج ايجابية لايمكن لا اي متطفل اختراق الشبكة وفتح التشفير ومعرفة الملف الصوتي وممكن اعادة البيانات الاصلية للملفات الصوتية بدون فقدان اي معلومة من قبل المستلم.

https://doi.org/10.30684/etj.32.1B.19
2412-0758/University of Technology-Iraq, Baghdad, Iraq

INTRODUCTION

This research is about a process which reduces the data rate or size of digital audio signal. Storage audio file; type of file is sent ".wave" to the network. Therefore to conserve the quality and quantity of files during the transmission through the network audio file is compressed and encrypted before sent to the network to prevent loss in audio data files. The Haar wavelet transform (HWT) method is used in files compression at third level (32part) to reduce size of audio file and the AES method is also used in cryptography audio (file.wave). In this way data of audio file are hidden during transmission of audio file while unauthorized person can not understand the audio data.

The idea of audio compression involves encoding audio data to take up less storage space and less bandwidth for transmission. To meet this goal different methods for compression have been designed [1]. The lossless compression works by removing the redundant information present in an audio signal. This would be the ideal compression technique as there is no cost to using it other than the cost of the compression and decompression process [1]. Lossless techniques are applied in the last stages of Audio and Video coders to reduce the data rate even further. In Lossy coding, the compressed data is not identical bit-for-bit to the original data. This method is also called Perceptive coding as it utilizes the fact that some information is truly irrelevant in that the intended recipient will not be able to perceive what it is missing. In most cases, information that is close to irrelevant is also made redundant, where the quality loss is small compared to the data savings [2][3].

AUDIO CRYPTOGRAPHY

Audio cryptography is not about audio encryption or encryption in audio data. Instead, audio cryptography shares similar conception with visual cryptography. A plain data is split into two or more shares. Each single share does not convey any meaning, but when shares are combined together they will reveal the original plain data [4].

Cipher algorithms deal with text data. Terms such "plaintext" and "ciphertext" emerge from here. As world is moving to more modern techniques, modern cipher algorithms deal with binary data. These binaries can represent everything in texts, spreadsheets, images, multi-media, programs, etc [4]. At this point, an audio data can be encrypted using any modern cipher.

ADVANCED ENCRYPTION STANDARD (AES)

This standard specifies the Rijndael algorithm, a symmetric clock cipher can process data blocks of 128 bits, using cipher key with lengths of 128, 192, and 256 bits.

In AES, all operations are performed on 8-bit bytes. In particular, the arithmetic operations of addition, multiplication, and division are performed over the finite field GF (2^8). AES operates on 8-bit bytes. Addition of two bytes is defined as the bitwise XOR operation. Multiplication of two bytes is defined as multiplication in the finite field GF (2^8), with the irreducible polynomial $m(x) = x8 + x4 + x3 + x + 1$. The developers of Rijndael give as their motivation for selecting this one of the 30 possible irreducible polynomials of degree 8. State in Figure (1, 2, 3-a, b) [5].

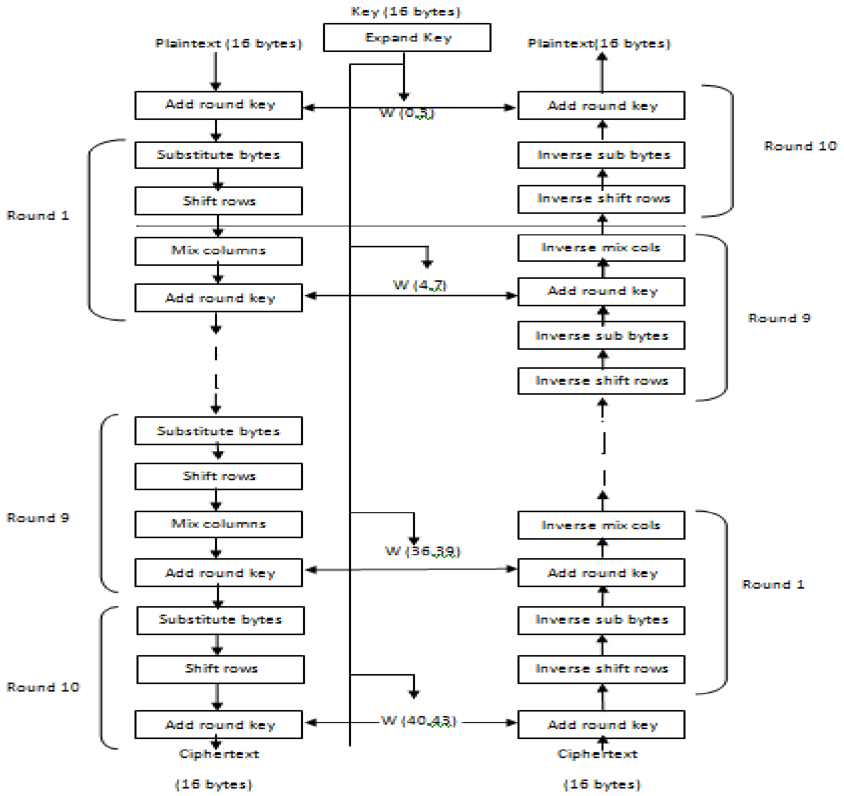


Figure (1) AES Encryption and decryption [5].

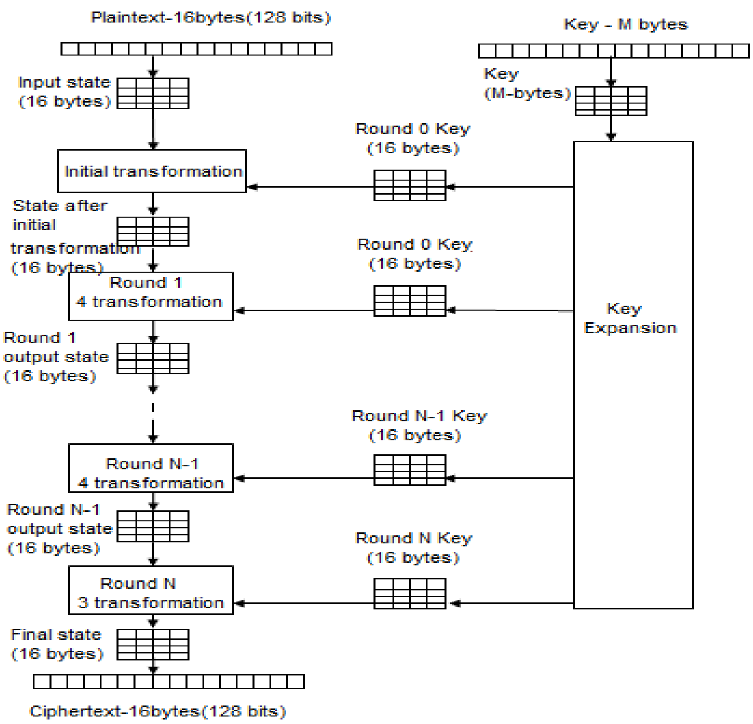
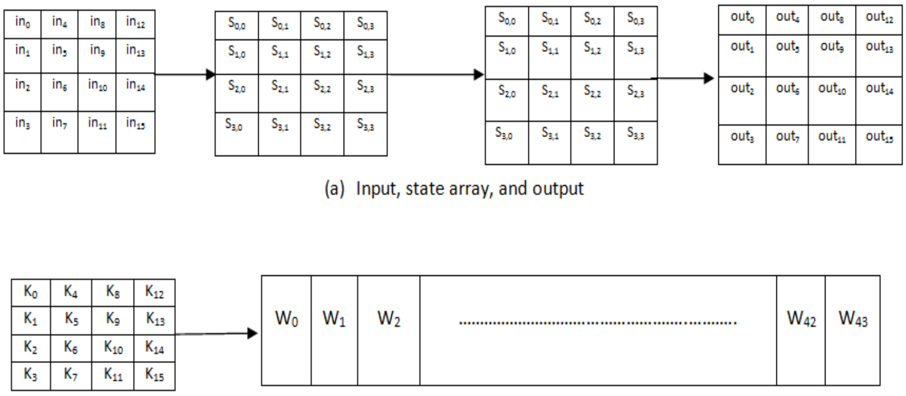


Figure (2) AES Encryption Process [5].



(a) Input, state array, and output

(b) Key and expanded key

Figure (3-a, b) AES Data Structures [5].

The key that is provided as input is expanded into an array of forty-four 32-bit Words, w[*i*]. Four distinct words (128 bits) serve as a round key for each round, in Table (1) these are indicated in Figure (1) [5].

Table (1) Round - Key length.

No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Four different stages are used, one of permutation and three of substitution:

- **Substitute bytes**: Uses an S-box to perform a byte-by-byte substitution of the Block. The forward substitute byte transform, called Sub Bytes, is a simple Table lookup [5].
- **SubBytes ()**: Transformation in the Cipher that processes the State using a non-linear byte substitution table (S-Box) that operates on each of State bytes independently. State in Figure (4) [5] [6].

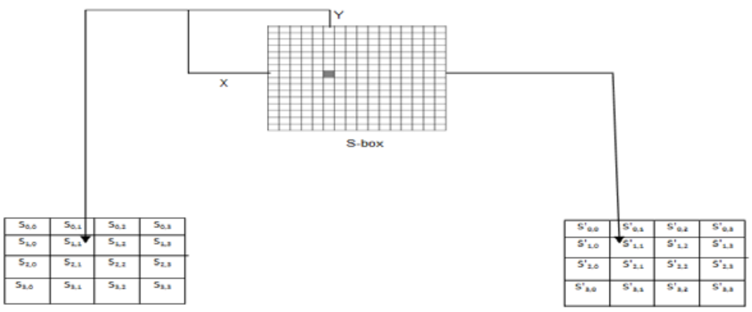


Figure (4) Sub Bytes Transformation [5].

- **Sub Word():** Function used in the Key Expansion routine that takes a four-byte input word and applies an S-Box to each of the four bytes to produce an output word[5].
- **Shift Rows:** Function is the Cipher that processes the state by cyclically Shifting the last three rows of the State by different offsets state in Figure (5) [5][6].

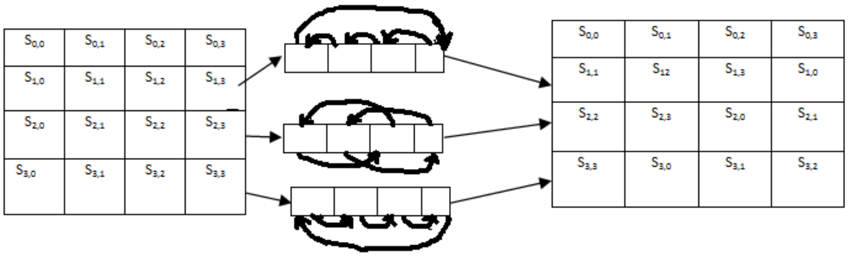


Figure (5) ShiftRows Transformation [5].

- **Mix Columns:** Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one other) to produce new Columns state in figure (6) [5][6].

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

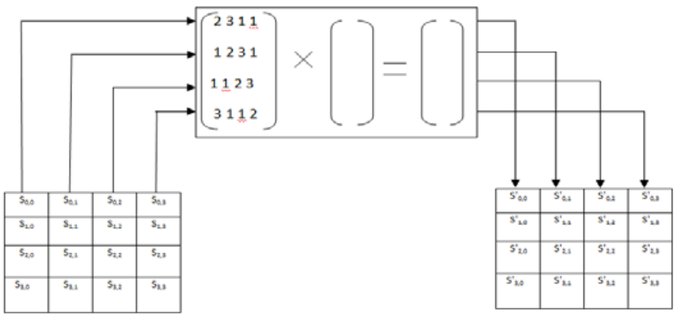


Figure (6) Mix Column Transformation [5].

- **AddRoundKey:** Transformation in the Cipher and Inverse Cipher in which a Round Key is added to the State using an XOR operation. The length of a Round Key equals the size of the State in Figure (7) [5][6].

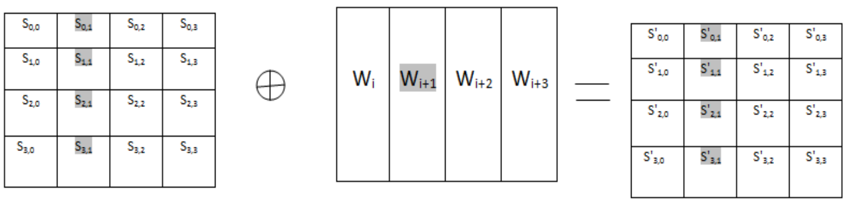


Figure (7) AddRoundKey Transformation [5].

HAAR TRANSFORM AND FAST HAAR TRANSFORM

The Haar transform (HT) is one of the simplest and basic transforms from the space domain to a local frequency domain [1]. A HT decomposes each signal into two components, one is called average (approximation) or trend and the other is known as difference (detail) or fluctuation. Haar transform technique is widely used these days in wavelet analysis. Fast Haar Transform is one of the algorithms which can reduce the tedious work of calculations [2]. One of the earliest versions of FHT is included in HT. FHT involves addition, subtraction and division by 2 [7][8].

THE STANDARD WAVE FILE

Besides the direct access to the audio data from the hardware it is also useful to be able to use recorded files, in fact this might not be necessary for real time systems but at least for the development and testing it is an essential source [9]. WAV is a short form for a Waveform audio format; it is a standard data format for storing audio data. The WAV format is a variant of the RIFF bit stream format method for storing data in "chunks", it is the main format used in Windows systems for raw audio. To use WAV files it is necessary to read or write the WAV file header [10].

The standard WAVE format used is created by the SOX program:
The Canonical WAVE file format in Table (2) [9].

Table (2) Wave File Format.

<table>
<tr><th>Endian</th><th>File offset (byte)</th><th>Field name</th><th>Field size (byte)</th><th>The RIFF chunk descriptor</th></tr>
<tr><td rowspan="13">Big
little
Big
Big
Little
Little
Little
Little
Little
Little
Little
Big
Little

Little</td><td rowspan="13">0
4
8
12
16
20
22
24
28
32
34
36
40
44</td><td>Chunk ID</td><td>4</td><td rowspan="3">The "RIFF" chunk descriptor the format of concern here is "WAVE ", which requires two sub-chunks:"fmt" and "data"</td></tr>
<tr><td>Chunksize</td><td>4</td></tr>
<tr><td>Format</td><td>4</td></tr>
<tr><td>SubChunk1ID</td><td>4</td><td rowspan="8">The "fmt" sub-chunk

describes the format of the sound information in the data sub-chunk</td></tr>
<tr><td>SubChunk1size</td><td>4</td></tr>
<tr><td>Audio Format</td><td>4</td></tr>
<tr><td>Num Channels</td><td>2</td></tr>
<tr><td>SampleRate</td><td>2</td></tr>
<tr><td>ByteRate</td><td>4</td></tr>
<tr><td>BlockAlign</td><td>2</td></tr>
<tr><td>Bitspersample</td><td>2</td></tr>
<tr><td>SubChunk2ID</td><td>4</td><td rowspan="2">The "data' sub-chunk</td></tr>
<tr><td>SubChunk2size</td><td>4</td></tr>
</table>

		Data	Subchunk2size	indicates the size of the sound information and contains the raw sound data

PROPOSED ALGORITHMS

This system uses two steps:

***First Step*:** involves using the compression of original audio data and ignoring header of audio file only ,using compression for audio data from file only for Haar wavelet transform (HWT) in array (1-D) in level two (16 part) and level three (32 part) determine bytes in (1-D), in this system using access to level three.

***Second Step*:** involves using the encryption audio data for file in level three (32 part) and dividing (1-D) audio data into groups each group contains eight bytes. After this step it works to merge two groups of sixteen bytes in audio data to create size matrix of sixteen bytes which is in the same size of matrix of Advanced Encryption Standard algorithm shown in Figure (8).

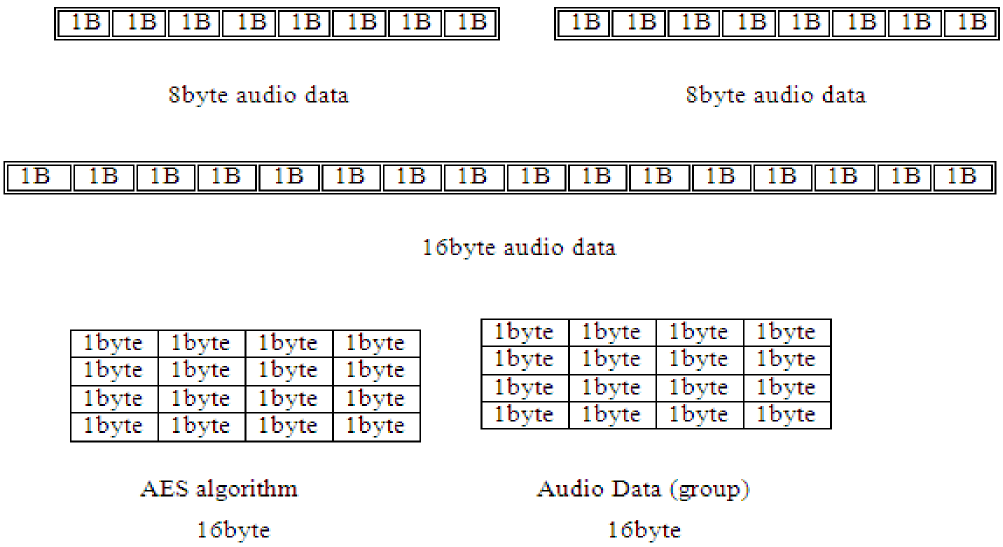


Figure (8) 1-D Audio Data and AES.

Apply aforementioned (AES) algorithm to all audio data groups, and repeat work to complete audio data encryption.

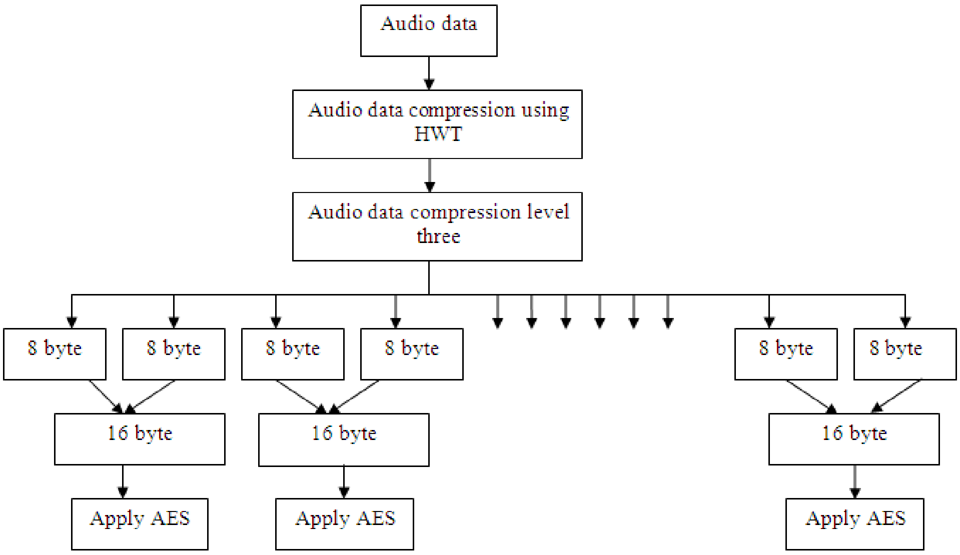
Select key block cipher to perform encryption and decryption on 128 bits input data and private key cipher which uses the same key in data encryption and data encryption operator for 128 bits, 192 bits, and 256 bits key size.

In this research 128 bits use 4 words equal to 10 numbers of a round key (Length of Key=32bits) in Table (3).

Table (3) Key blocks round combination.

Flavour	Key length (Nk word)	Block size (Nb words)	Number of Round (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

The block diagram below explains the main idea of proposed algorithm.



Implementation of audio compression and wavelet transform

The audio data convert to one dimension in array of signal 8byte, and each 8byte division high array and low array. Merge the high and low array in one array and take the positive number only to compression audio data and ignore negative number such as in structure below.

Signal audio [61 24 50 70 67 94 77 59]
Array of signal = Length of 8 byte.
Array of high= Length of 9 byte.
Array of low= Length of 9 byte.
Low= [61 85 74 120 137 161 171 136 59].
High= [-61 37 -26 -20 3 -27 17 18 59].
High &Low = [85 120 161 136 37 -20 -27 18].
High= [37 -20 -27 18].
Low= [85 120 161 136].
Final array [37 20 27 18 85 120 161 136]. (All value is positive)

Apply EAS encryption in system

Proposed Algorithm

The algorithm system work compression audio data and encryption audio data.

Process:
Input: Audio compression, key (AES algorithm). Output: Cipher Audio.
Step1: Initial
A = Load Audio data.
B = Compressed Audio data in sixteen parts (LL2).
C = Compressed Audio data in thirty-two parts (LL3).
D = Audio encryption.
Step2: Encryption Process PlainBlock: (level three C)
Step3: Apply the key to the audio data A: (0 to 9) & (A to Z).
Step4: Add Row Key on the audio data.
Step5: Sub Bytes on the audio data: Substitution.
Step6: Shift Row on the audio data.
'Shift row 2
'shift row 3
'shift row 4
Step7: Multiplication.
Step8: Mixcolumns.
Step9: Addround key.
Step10: Result (Put the result encrypted audio in D).

Using S-box length 16byte depended in Table S-box on index form reference [2].

Shift row

Implementation shift row in system length is 16 byte. It is transposition step where each row of the state is shifted cyclically a certain number of steps. The first row is unchanged. The second row is shifted one byte to the left. The third row is shifted two byte to the left. And the fourth row is shifted three byte to the left, such as structure below.

Before shift row

29	228	75	205
13	219	161	61
229	96	247	184
123	238	105	25

after shift row

29	228	75	205
219	161	61	13
247	184	229	96
25	123	238	105

Multiplication

Implementation multiplication in system: C= Length 16 byte, X= Length 4 byte, Y= Length 4 byte

C array

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

$\oplus$ —

X array

29
219
247
25

Y array

162
171
24
57

Mix Columns: A mix column is length 16 byte. It is a mixing operation which operates on the columns of the state, combining the fourth bytes in each column. The matrix after shift row multiplication(C array) is standard XOR (x array) done using in which the new column (Y array) is XOR product of the old column (x array) first column after shift row and a constant matrix, such as structure below.

29	228	75	205
219	161	61	13
247	184	229	96
25	123	238	105

X

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

$\oplus$

29
219
247
25

$\oplus$

162
171
24
57

=

162	228	75	205
171	161	61	13
247	184	229	96
25	123	238	105

Key expansion: the key expansion length 176, key length 176.

Each array of key from 0 to 43.

Key schedule at 0 to 15. [139 139 139 139 139 139 139 139 139 139 139 139 139 139].

- The key expansion function takes 16 bytes long and utilizes the round constant matrix rcon and substitution table SBox to generate a 176 byte schedule using Encryption and decryption processes.

Key expansion

139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
-	-	-	-
-	-	-	-
-	-	-	-
Sy139	139	139	139

Key schedule

139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
139	139	139	139
-	-	-	-
-	-	-	-
-	-	-	-
139	139	139	139

Decryption in System

Proposed Algorithm

The algorithm system work decryption audio data and decompression audio data.

Process:
Input: Audio Decompression, key inverse (AES algorithm). Output: Audio Decryption.
Step1: Initial
A = Load Audio data Encryption.
B = Decompressed Audio data in sixteen parts (LL2).
C = Decompressed Audio data in thirty-two parts (LL3).
D = Audio Decryption.
Step2: Apply the Inverse key to the audio data A: (0 to 9) & (A to Z).
Step3: Inverse Add Row Key on the audio data.
Step4: Inverse Sub Bytes on the audio data: Substitution.
Step5: Inverse Shift Row on the audio data.
Step6: Inverse Multiplication.
Step7: Inverse Mixcolumns.
Step8: Inverse Addround key.
Step9: reconstraction (Audio).
Step10: Decryption Process PlainBlock: (level three C)
Step11: Result (Put the result decrypted audio in D).

TEST THE RESULTS

In this section three example tests are executed involving the cryptography audio, shown in the table (4) below.

Table (4) Test of Result.

File Name	Audio Type	Frequency HZ	Value dB	View Type	Size
Call.wav	Original	10185 HZ	-96dB	Linear	1024
	Compression	13783 HZ	-99dB	Linear	1024
	Encryption	16771 HZ	-100dB	Linear	1024
	Decryption	10185 HZ	-96dB	Linear	1024
Chimes.wav	Original	9370 HZ	-90dB	Linear	1024
	Compression	17450 HZ	-98dB	Linear	1024
	Encryption	20166 HZ	-100dB	Linear	1024
	Decryption	9370 HZ	-90dB	Linear	1024
Cycle.wav	Original	2716 HZ	-87dB	Linear	1024
	Compression	15074 HZ	-90dB	Linear	1024
	Encryption	23222 HZ	-99dB	Linear	1024
	Decryption	2716 HZ	-87dB	Linear	1024

CONCLUSIONS

In this search AES algorithms are used for efficient encryption of audio data without header usage after compression audio data in wavelet transform method in LL2 and LL3, reducing the data transmitted across the network. This algorithm give can high frequency in compression and encryption of audio data but on the decryption of audio data the frequency is the same frequency in original audio data that is low (before compression and encryption) and without loss in the audio data during transmission across the network or the internet.

In this search using inverse AES algorithms are used for efficient decryption of audio data without header usage also. And decompression audio data from LL3 and LL2 in wavelet transform method without loss any information in audio files.

This state indicates the system is very efficient because it is capable of keeping to audio data without loss of any information, although the header of audio is hidden during transmission.

REFERENCES

[1].Pollak l., “Audio Compression Using Wavelet Techniques”, Electrical and Computer Engineering Purdue University, 2005. {mzanartu@ecn.purdue.edu}

[2].Othman O. Khalifa, Harding S. H. and Hashim A.-H. A. “Compression using Wavelet Transform”. http://thamjoyew.tripod.com/papers/IEEE-R10-full.pdf-cashes-similar

[3].Cheddad A., Condell J., Curran K.and Kevitt P. MC "Digital Image Steganography Survey and Analysis of current Method".{Email:chedded-a@email.ulster}

[4]. Adriansyah Y, "Simple Audio Cryptography", Institute Teknologi Bandung, JI.Ganesha 10 Bandung 40123, Indonesia, 2010.

[5]. Stallings W., "Cryptography and Network Security", Prentice Hall, principle and Practice, Fifth Edition, 2011.

[6]. Vasumathy N. D., Velmathi G., and Sklavos N.,"On the Rijndael Encryption Algorithm Implementation with MATLAB Software Programming", partment of Electronic s and Communication Engineering Sri Siva Subramania Nadar College of Engineering, TamilNadu,India, 2008. http:// www.wseas.us/e-library/conferences/2009/baltimore/.../DNCOCO-31.pdf - Cached - Similar

[7]. Umbaugh S.E.,” Computer Vision and Image Processing”, Prentice Hall Inc,1998.

[8]. Bhardwaj A. and Ali R., “Image Compression Using Modified Fast Haar Wavelet Transform”, Department of Mathematics, Vishveshwarya Institute of Engineering and Technology, Dadri,G,B.Nagar-203207, U.P. India, 2009. http:// idos.org/wasj/wasj7(5)/13.pdf-cashes-similar

[9]. “Microsoft WAVE PCM soundfile format”, http://ccrma.stanford.edu/courses/422/project/waveFormat/,2010.

[10]. Brandau M. A., " Implementation of a real-time voice encryption system", MSC Thesis Universitat Politecnica de catalunya EUETIT,2008. http://upcommons.upc.edu/pfc/bitstream/2099.1/.../MarkusBrandau.pdf-Cashed-similar.

Appendex (1)

S-box: is length 16 byte

	Y																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	D3	38	54	5F	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	D3	8F	5B	CB	6D	A2	11	0F	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Implementation S-box in system

	Y																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	99	124	119	123	242	107	111	197	48	1	103	43	254	215	171	118
	1	202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
	2	183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
	3	4	199	35	195	24	150	5	154	7	18	128	226	235	39	178	117
	4	9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
	5	83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
	6	208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
	7	81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
	8	205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
	9	96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
	A	224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
	B	231	200	55	109	141	213	78	169	108	86	244	243	101	122	174	8
	C	186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
	D	112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
	E	225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
	F	140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

Appendex (2)

Implementation program system

Enter key 16 characters encryption audio

Enter the same key 16 characters decryption audio