Proposed Method for Partial Audio Cryptography Using Haar Wavelet Transform

[1]Prof. Dr. Abdul Moneem S. Rahma [1]Lecturer Maisaa Abid Ali Khodher

[1]Computer Science Department, University of Technology -Baghdad, Iraq.
e-mail:monem.rahma@yahoo.com e-mail:maisaa_ali2007@yahoo.com

Received on: 4/104/2011

Accepted on: 22 /5/2012

Abstract - The rapid developments in communication networks and the Internet have led to propose an encryption algorithm based on a new transfer audio data across networks more quickly. A reduction of data audio files transferred across the network without loss of data led to merging algorithms to compress data at third-level with encryption algorithm, which maintains the confidentiality of information transmitted across the network. This merging is aimed at the transfer of data encryption honestly with rapid retrieving.

The results are obtained by the adoption of the proposed encryption algorithm to generate a random key for possible restoration of the original data files, audio files recorded after the receipt of data without any loss of information.

Keywords- Audio compression, wavelet transform, linear cipher, combining function, partial cryptography.

1- Introduction

Audio compression allows the efficient storage and transmission of audio data. The various audio compression techniques offer different levels of complexity, compressed audio quality, and amount of data compression.

This research uses the partial audio using Haar wavelet transform at the third level of audio data compression and then it uses encoding audio data compressed using low pass and high pass filter depending on the way of least significant bit and after this stage feed back shift register is used so as to generate a randomized key to make more powerful data encryption.

2- Audio Compression Techniques

The idea of audio compression is to encode audio data to take up less storage space and less bandwidth for transmission. To meet this goal different methods for compression have been designed. Just like every other digital data compression, it is possible to classify them into two categories: lossless compression and lossy compression [1],[8].

2.1- Lossless Compression

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. However, lossless compression suffers from two disadvantages. First, it offers small compression ratios, so using it alone does not meet economic needs. Also, it does not guarantee a constant output data rate as the compression ratio is very much dependent on the input data. One advantage of Lossless compression is that it can be applied to any data stream. Lossless techniques are applied in the last stages of audio and video coders to reduce the data rate even further. Two lossless techniques that are in general use are: Run-Length Encoding and Entropy Encoding [5],[6],[10].

2.2- Lossy Compression

In lossy coding, the compressed data is not identical bit-for-bit with 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 that it is missing. In most cases, information that is close to irrelevant is also made redundant, because the quality loss is small compared to the data saving.

The objective of lossy compression is to get maximum benefit, compression ratio or bit rate reduction, at reduced cost, and loss in quality [8],[9].

To pinpoint the portions of the audio signal that is redundant involves using psychoacoustic analysis to determine a masking threshold below which the power of the signal is not strong enough to be heard by the human ear. See Fig. (1). below illustrates this point [5],[6].

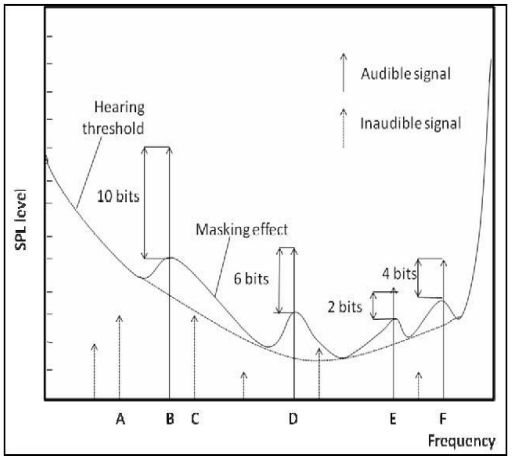


Figure (1) The bit allocation algorithm assigns bits according to audibility of sub band signals. Inaudible tones are not assigned bits, and are not coded [5]

3- Haar Transform and Fast Haar Transform

Haar transform (HT) is one of the simplest and basic transforms from the space domain to a local frequency domain. An HT decomposes each signal into two components, one is called average (approximation) or trend and the other is known as difference (detail) or fluctuation [9],[10].

Haar transform technique is widely used these days in wavelet analysis. Fast Haar [9]. Transform is one of the algorithms which can reduce the tedious work of calculations. One of the earliest versions of FHT is included in HT. FHT involves addition, subtraction and division by 2 [2],[3],[10].

4- WAVE PCM Sound file Format

The WAVE file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. A RIFF file starts out with a file header followed by a sequence of data chunks. A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks—an "fmt" chunk specifying the data format and a "data" chunk containing the actual sample data, this forms the "canonical form"; which knows how it really all works [4][7].

The standard WAVE format is used as created by the SOX program:

The Canonical WAVE file format

Endian	File offset (byte)	Field name	Field size (byte)	The RIFF chunk descriptor
Big	0	Chunk ID	4	The "RIFF" chunk descriptor
little	4	Chunksize	4	The format of concern here is "WAVE ", which requires two sub-chunks: "fmt" and "data"
Big	8	Format	4	
Big	12	SubChunk1ID	4	The "fmt" sub-chunk
Little	16	SubChunk1size	4	
Little	20	Audio Format	4	
Little	22	Num Channels	2	describes the format of the sound information in the data sub-chunk
Little	24	SampleRate	2	
Little	28	ByteRate	4	
Little	32	BlockAlign	2	
Little	34	Bitspersample	2	
Big	36	SubChunk2ID	4	The "data' sub-chunk
Little	40	SubChunk2size	4	
	44	Data	Sub-chunk2size	indicates the size of the sound information and contains the raw sound data

Notes:

- The default byte ordering assumed for WAVE data files is little-endian. Files written using the big-endian byte ordering scheme have the identifier RIFX instead of RIFF.
- The sample data must end on an even byte boundary, whatever that means.
- 8-bit samples are stored as unsigned bytes, ranging from 0 to 255. 16-bit samples are stored as

2's-complement signed integers, ranging from -32768 to 32767.

- There may be additional subchunks in a wave data stream. If so, each will have a char [4].
- subchunkID, unsigned long subchunkSize, and subchunkSize amount of data.
- RIFF stands for Resource Interchange file format [4].

4.1- General Discussion of RIFF Files

Multimedia application requires the storage and management of a wide variety of data including bitmaps, audio data, video data, and peripheral device control information. RIFF provides a way to store all these varied types of data. The type of data a RIFF file contains is indicated by the file extension. Data that may be stored in RIFF files are [4],[7]:

- Audio/visual interleaved data (.AVI)
- Waveform data (.wav)
- Bitmapped data (.RDI)
- MIDI information (.RMI)
- Color palette (.PAL)
- Multimedia movie (.RMN)
- Animated cursor (.ANI)
- A bundle of other RIFF files (.BND)

Notes:

At this point, AVI files are the only type of RIFF files that have been fully implemented using the current RIFF specification. Although WAV files have been implemented, these files are very simple, and their developers typically use an older specification in constructing them [4].

5- Proposed Algorithm of Partial Audio Cryptography

This research uses the partial audio by using the compression method; the type of file is audio .wav.

5-1 Audio Compression

When audio data is compressed in wavelet transform after the process of audio data compression, the process of data encryption begins only without encryption header and has no effect on the encode signal, because the data sometimes carry a signal which is negative. It adds only one byte and every bit of it refers to either a positive or negative signal in this byte.

They are shifting eight bytes, each byte shifting is directed by right step one (one bit) to delete least significant bit.
In this section data is using compressed Haar wavelet transform with level three in audio using part 32 or 2^5.
Audio compression hides features header only but data can be encrypted in two steps: step one uses both high pass and low pass filters, step two uses shift register and combining function.

Step one:
After audio compression, the optimal compression value uses high pass and low pass filters.
Signal = [1 3 2 4 2 1 5 3] …. Audio compression value
Low pass = [1 1], high pass = [1 -1]
Applied low pass filter:

Low = [1 4 5 6 6 3 6 8 3]
Low = [4 6 3 8]
Decomposition filter

Low = $1/\sqrt{2}$ [1 1]

High = 1/ $\sqrt{2}$ [1 -1]
Applied high pass filter:
Signal = [-1 -2 1 -2 2 1 -4 2 3]
High = [-2 -2 1 2]

Merging low pass and high pass filters yields

L							H
4	6	3	8	-2	-2	1	2

8 byte encoding signal

Step two:

Reconstruction of eight byte encoding signal for audio:

Low = [1 1], High = [-1 1]

1- Put zero between low pass to become:
[2 0 6 0 3 0 8]
2- Pass it low to become:
[4 4 6 6 3 3 8 8]
3- Put zero between high pass to become:
[-2 0 -2 0 1 0 2]
4- Pass it high to become:
[-2 2 -2 2 1 -1 2 -2]
5- Add the result of low pass together with the result of high pass to become:

[4 4 6 6 3 3 8 8]
[-2 2 -2 2 1 -1 2 -2]

[2 6 4 8 4 2 10 6]

6- The result of adding low and high is divided by 2:

[2 6 4 8 4 2 10 6] /2
[1 3 2 4 2 1 5 3]

7- After division last operation is first encoded involving signal wave of eight bytes from low pass and high pass filters.

L							H
1	3	2	4	1	2	5	3

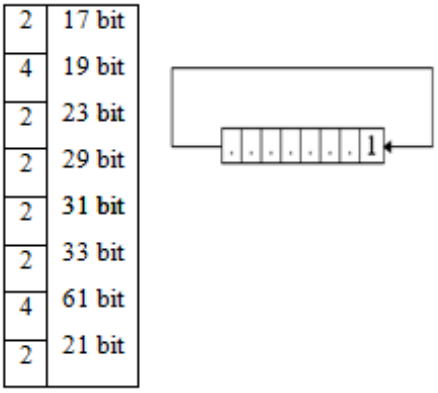
8 Byte

- Each byte carries information about audio.

5-2 Linear Feedback Shift Register and Function Combining

In this section session key is generated randomly for encryption using stream cipher method, whenever output enters eight bytes from low pass to high pass filters the feedback will be linear. Then the output combines with XOR gate to increase encryption. The shift register starts in x0, x1, x2,……x7 and combining function ranges from 0 to 255. Whenever generated key is most powerful, it takes sixteen characters with audio compression XOR, the result of output is audio encryption. When you send this audio data, the receiver can know how to open the audio data which has become more secret as see Fig. (2).

Note: when you encrypt audio data the result of encryption appears as session key. You keep that key. Upon transfer of audio data you can send that session key with each audio transfer depending on basic key and session key.



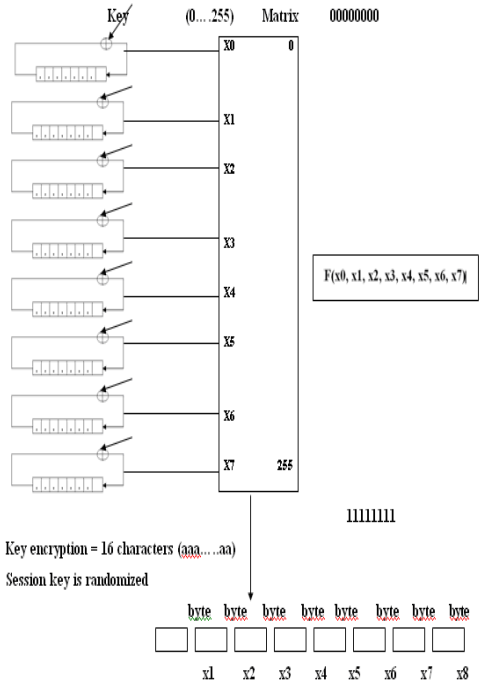


Figure (2) Audio data encryption

6-The Result of implementation System

In figures (3, 4, 5, 6) and examples (1, 2, and 3)

Figure (3) Enter the basic key

Figure (4) compression and encryption
using basic key

Figure (5) Generated Session Key Randomized
Compression / encryption

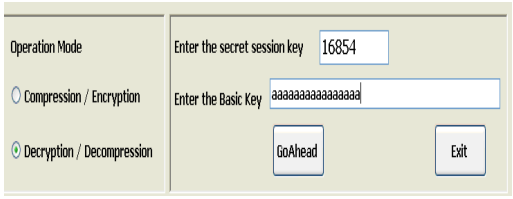
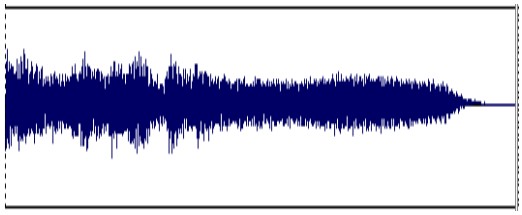
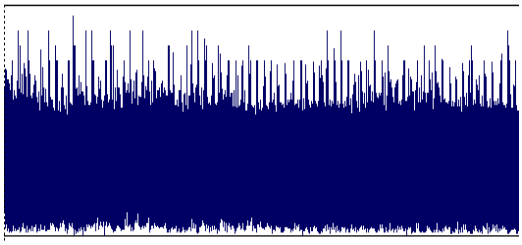


Figure (6) Enter Generated Session Key Randomized
And Enter the basic key
Decryption / Decompression

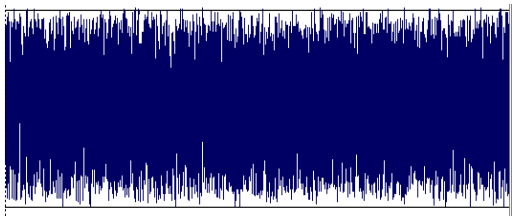
Example: 1



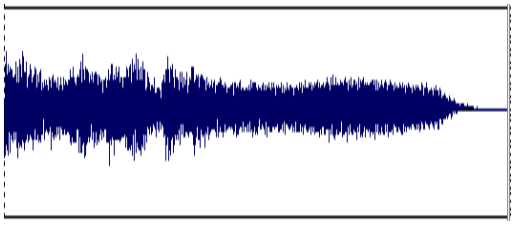
Original Audio / ringin.wav



Compression Audio / ringin.wav



Compression and encryption / ringin.wav



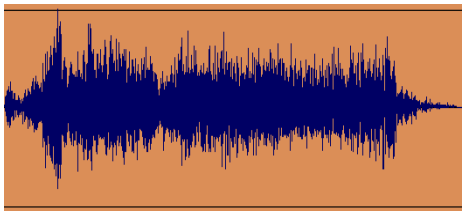
Decompression and decryption / ringin .wav

Chunk ID ==> RIFF
Chunk Size ==> 10018 (Filesize - 8) byte
Format ==> WAVE
SubChunk1 ID ==> fmt
Chunk Size ==> 16
Audio format PCM = 1 ==> Linear Quantization
NumChannels 1 ==> Mono
Sample Rate 11025
Byte Rate 11025 computed Byte Rate 11025
Block Align 1 computed Block Align 1

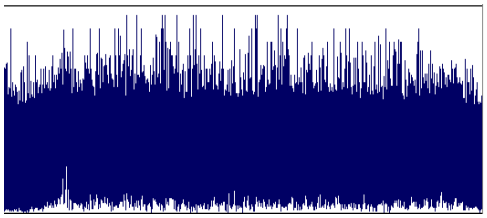
<== Data SubChunk ==>
SubChunk2 ID ==> data
SubChunk2 Size ==> 9981

Header and data to Rangin .wav

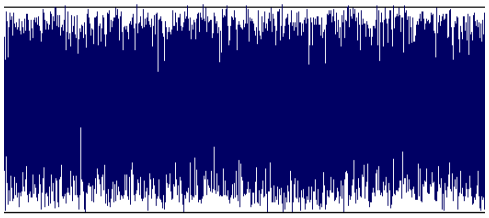
Example: 2



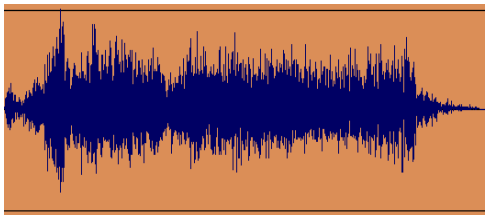
Original Audio / kas .wav



Compression Audio / kas .wav



Compression and encryption / kas .wav



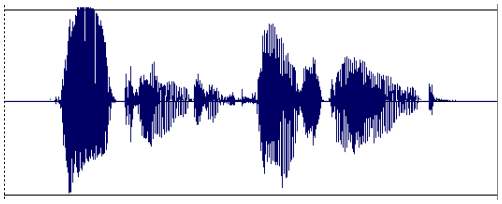
Decompression and decryption / kas .wav

Chunk ID ==> RIFF
Chunk Size ==> 9988 (Filesize - 8) byte
Format ==> WAVE
SubChunk1 ID ==> fmt
Chunk Size ==> 16
Audio format PCM = 1 ==> Linear Quantization
NumChannels 1 ==> Mono
Sample Rate 11025
Byte Rate 11025 computed Byte Rate 11025
Block Align 1 computed Block Align 1

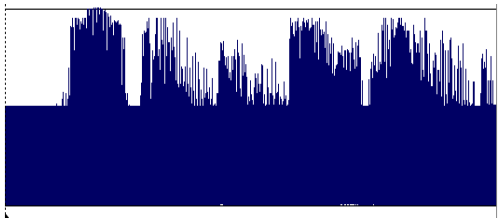
<== Data SubChunk ==>
SubChunk2 ID ==> data
SubChunk2 Size ==> 9771

Header and data to kas .wav

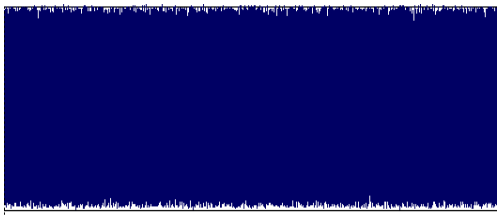
Example: 3



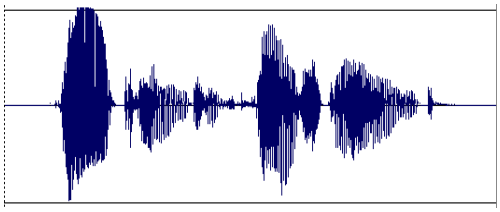
Original Audio / file1.wav



Compression Audio / file1.wav



Compression and encryption / file1.wav



Decompression and decryption / file1 .wav

```
Chunk ID ==> RIFF
Chunk Size ==> 69868  (Filesize - 8) byte
Format ==> WAVE
SubChunk1 ID ==> fmt
Chunk Size ==> 16
Audio format PCM = 1 ==> Linear Quantization
NumChannels 1 ==> Mono
Sample Rate   44100
Byte Rate   44100              computed Byte Rate   44100
Block Align   1                computed Block Align   1

                    <== Data SubChunk ==>
SubChunk2 ID ==> data
SubChunk2 Size ==> 69832
```

Header and data to file1.wav

7- Conclusion

This research provides a good and efficient method for encryption of the audio data and sending it to the destination in a safe manner across the network at high speed.

In this system data compression makes the size of the audio file small even after encoding and also it becomes suitable for any type of audio file format. Encryption and Decryption techniques have been used to make the security system robust across network**.** The header of audio does not change information in compression, it compresses only data, and the header hides the information.

In comparison between continuous wavelet transform and discrete wavelet transform method, the audio.wav files cannot return to original file, whereas in Haar wavelet transform method the (audio.wave) file can return to original exactly.

References

1- llya Pollak, “Audio Compression Using wavelet Techniques”, Electrical and Computer Engineering Purdue Department University, 2005.

2- Scott E. Umbaugh, “Computer vision and Image Processing”, Prentice Hall Inc, 1998.

3- Anuj Bhardwaj and Rashid Ali, “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.

4-“Microsoft WAVE PCM soundfile format”, 2010.

5- Othman O. Khalifa,Sering Habib Harding and Aisha-Hassan A. Hashim “Compression using Wavelet Transform”.

6- Abbas Cheddad, Joan Condell, Kevin Curran and Paul MC Kevitt "Digital Image

7- Steganography Survey and Analysis of Current Methods".

8- Yicog Zhou, Karen Panetta, and Sos Aagaian, "Partial Multimedia Encryption with Different Security Level", 2008.

9- Tomes Sander, Christian F. Tschudin, "Towards Mobile Cryptography", International Computer Science Institute 1947 center Street, Berkeley, CA94704, 2008.

10- Rafael C.Gonzalez, Richard E. Woods, “Digital Image Processing” University of Tennessee, Prentice Hall, 2001.

11- Tinku Acharya, Ajoy K.ray,”Image Processing Principle and Application”, Avisere.Inc. Tucson, Arizona,