Data Hiding in Sound Using Time Modulation of Unvoiced Blocks

Dr. Hilal H. Saleh*, Dr. Loay A. Jorj* & Dr. Nidaa F.Hassan*

Received on:15/10/2008
Accepted on:31/12/2008

Abstract

This paper is concerned with hiding secret data in audio media file (.WAV). When performing data hiding in audio, one must exploit the weaknesses of Human Auditory System (HAS), while at the same time being a ware of the extreme sensitivity of this system. This hiding method is oriented to embed the secret data such that it is capable of surviving against modifications produced by MP3 compression standard. Statistical and analytical investigations are performed to assess the variations which may occur in the WAVE audio when it is subjected to MP3 compression. Features of speech signal are exploited (Voiced-Unvoiced segments) for embedding the secret data. Hiding is attempt by shortening or elongating the unvoiced blocks of audio file (cover) data. To support the immunity of the proposed hiding system, an encryption method is added to the proposed hiding system.

Key-words: Audio Hiding, Speech Analysis methods, Audio compression.

أخفاءالبيانات في الصوت بأستخدام تعديل الوقت للمقاطع اللاصوتية

الخلاصه

يهتم هذا البحث بأخفاء البيانات السرية في ملفات صوتية ذات الأستطالة (.Wav) . أن الأخفاء في الصوت يتم عن طريق استغلال مناطق الضعف في النظام السمعي للانسان وفي نفس الوقت يجب الحذرمن التحسس الشديد لهذا النظام. ان طريقه الاخفاء جرى تصميمها لأخفاء بيانات سرية لها القدرة على الصمود أمام التغيرات التي يمكن ان يتعرض له الصوت بواسطة البرنامج القياسي لضغط الصوت (MP3) . فبعد أجراء تحليل أحصائي وعددي لأيجاد أنماط الأختلافات التي قد تحدث على بيانات الملف الصوتي (WAVE) عندما يتعرض الى ضغط بواسطة (MP3) . تم أستغلال خصائص أشارة الكلام (المقاطع الصوتية واللاصوتية) لغرض أخفاء البيانات السرية. حيث تم أخفاء البيانات السرية بأستخدام تقنية تقصير أو أستطالة المقاطع اللاصوتية في الملف الصوتي (الغطاء) .ولزيادة درجة الامنية للنظام المقترح , تم أضافة طريقة تشفير الى نظام الأخفاء المقترح.

1. Introduction

Data hiding embeds data into digital media for the purpose of identification, annotation, and copyright. Several constraints affect this process: the quantity of data to be hidden, the need for invariance of these data under conditions
where a "host" signal is subjected to distortions, e.g., lossy compression, and the degree to which the data must be immune to interception, modification, or removal by a third party [1].

The central theme of this work is concentrated on the fact there must be an appropriate compromise between data hiding in audio and perceptual coding. Perceptual coding refers to the lossy compression of multimedia signals using human perceptual models; the

https://doi.org/10.30684/etj.27.6.16
2412-0758/University of Technology-Iraq, Baghdad, Iraq

compression mechanism is based on the premise that minor modifications of the signal representation will not be noticeable in the displayed signal content. These modifications are imposed on the signal in such a way as to reduce the number of information bits required for storage of the content. Human perceptual models are often theoretically and experimentally derived to determine the changes on a signal, which remains imperceptible. So, one of the main obstacles within the data hiding in audio is to develop a scheme which is robust to perceptual coding standard (MP3). MP3 is one of many methods to compress audio in digital form trying to consume as little space as possible but keep audio quality as good as possible. MP3 is one of the best achievements in this area [2].

Hiding methods are designed to embed the secret data by time modulating (shortening or elongating) the unvoiced blocks of WAVE file to be resisting against MP3 compression. The data are encrypted before hiding to give more security immunity to the covered data, so if the attacker extracts the correct encrypted data from the stego object, he will discover that the extracted data have no meaning.

2. Properties of Data Hiding

The most important properties of data hiding schemes are robustness, undetectability, invisibility, security, complexity, and capacity [3]. These properties are mutually competitive and cannot be clearly optimized at the same time. This observation is schematically depicted in Figure (1).

3. Data Compression and Information Hiding

Data hiding within multimedia has received growing interest in recent years due to its potential for signal captioning, maintaining audit trials in media commerce, and copy protection through the development of digital watermarking technology. MP3 is one of many methods to compress audio in digital form trying to consume as little space as possible but keep audio quality as good as possible. MP3 is one of the best achievements in this area [2].

Compression is one of the most common operations on digital files; therefore, we must take into account the effect of compression when designing information hiding. Traditionally, data hiding and compression have had contradictory goals. The former adds perceptually irrelevant information in order to embed data, while the latter removes this irrelevancy and redundancy to reduce storage requirements [4].

This hiding method exploits the some features of audio signal to be able for perceptual and robust hiding.

4. Audio Signal Classification

Audio signal classification (ASC) consists of extracting physical and perceptual features from a sound, and of using these features to identify into which set of classes the sound is most likely to fit. The feature extraction and classification algorithms used can be quite diverse depending on the classification domain of the application .The first step in any classification problem is to identify the features that will be used to classify the data [5].

4.1 Physical and Perceptual Features

The features typically used in ASC can be divided into physical and perceptual categories. Physical features are properties that correspond to physical quantities, such as fundamental frequency (F0), Energy (EN), Zero-crossing rate (ZCR) and Modulation rate. In this research Energy (EN) and Zero-Crossing rate (ZCR) were utilized, these two features are used to make a simply discriminate between Voiced / Unvoiced (V/UV) audio segments [5].

A. Energy Measurement

One of the simplest representations of a signal is its energy. In the case of a real discrete-time signal x (n), the energy is defined in general in equation (1):

$$EN(n) = \sum_{n=-\infty}^{\infty} x(n)^2 \quad \ldots\ldots\ldots (1)$$

For nonstationary signals such as speech, it is often more appropriate to consider a time-varying energy calculation such as the following equation:

$$EN(n) = \sum_{n=0}^{N-1} x(n)^2 \; W(n) \ldots\ldots (2)$$

Where N is the number of samples in the nth sample in the frame.

The major significance of Energy Measure (EN) is that it provides a good measurement for separating voiced speech segments from unvoiced speech segments. EN for unvoiced segments is much smaller than for voiced segments [6].

B. Zero-Crossing Measurement

Zero crossing is very simple time- domain analysis method. In the context of a digital implementation, a zero crossing can be said to occur between sampling instants n and n-1 as equation (3):

$$\text{Sign}[x(n)] \neq \text{Sign}[x(n-1)] \quad \ldots (3)$$

Zero crossing rate measures how often the sound signal crosses from positive to negative or vice-versa. Zero crossing measurements (along with energy information) are often used in making a decision about whether a particular segment of speech is voiced or unvoiced. If the zero crossing rate (ZCR) is high, the implication is unvoiced; if the zero crossing rate is low, the segment is most likely to be voiced. Speech signals are broadband signals and the interpretation of average zero-crossing rate is therefore much less precise [6].

5. Hiding Data in Audio

Hiding data in audio, exploits how the human auditory system (HAS) interprets sounds. This method becomes especially challenging, since the HAS is extremely sensitive. The HAS drowns quiet sounds and emphasizes larger sounds. The goal of steganography in audio is to exploit this weakness. Bits that encode sound outside the range of human hearing can be encoded with covert data [7].

While HAS has a large dynamic range, it has a small differential range, and as result, loud sound tends to mask quiet ones. Additionally, the HAS does not perceive absolute phase, but only relative phase. Finally, there are some environmental distortions so common as to be ignored by the listener in most cases. These “holes” can be exploited by data hiding techniques [8].

6. Data Hiding within Unvoiced Blocks

The technical challenges of data hiding are formidable. Any “holes" to fill with data (either statistically or perceptually) in the host signal you may utilize are likely be targets for removal by lossy signal compression. The key to successful data hiding is the finding of holes that are not suitable for exploitation by compression algorithms.

In this section, a method for hiding secret data in audio files by detection of unvoiced / voiced blocks is introduced. The bit is embedded by shortening or elongating the period (i.e., time length of unvoiced blocks). As the first steps in designing stage of this method, statistical and analytical investigation are performed to assess the difference which may occur in the WAVE audio when it is subjected to MP3 compression, and these investigations include:

A. Comparison of WAVE audio file size before and after MP3 compression
B. Comparison between lengths and locations of unvoiced segments.

A. Comparison of WAVE files size before and after MP3 compression

In this section, test samples (WAVE audio files) are subjected to MP3 compression and decompression to demonstrate the differences in size that occurs when WAVE file is subjected to this (MP3) attack at different levels of compression. The test samples are different in size and type (song, music, speech). Table (1) shows some test audio files (**F1**, **F2,** and **F3**) of WAVE type, PCM format, one channel (mono), and 8-bit sample value, they were compressed by different levels of MP3 compression and then the MP3 files were decoded to reconstructed the new WAVE files format (all these coding steps were performed by using Sound Forge Ver. 6.0).

From the results listed in the table (1), it is noticed that the reconstructed file is different in size with original file even at 128 kbps compression level

B. Comparison of unvoiced bytes, unvoiced blocks and their locations

This stage provides primarily analyses to detect:

1. The number of unvoiced byte.
2. The number of unvoiced blocks and their position.

These analyses are applied to the original file (WAVE) before compression and the reconstructed file (WAVE) after decompression. The detection of whether a sample is unvoiced or voiced is based on whether the sample value is near to 128 (offset value) or not. Table (2) shows the number of unvoiced segments in the original and reconstructed file.

Now, the unvoiced blocks and their positions in the file are to be allocated. The detection of whether a block is unvoiced or not depends on searching for any sequence of samples whose values are near to 128. In tables (3), the detection of the first (10) unvoiced blocks and their positions are listed for test files (**F1**, **F2,** and **F3**), before MP3 compression and after MP3 decompression at Compression level 128 kbps

According to the results, the following remarks must be taken into consideration:

1. Compression levels 128 kbps and 96 kbps lead to positive results since some similarities in length and number of unvoiced segments (extracted from the original and decompressed audio data) exist, while the dissimilarity could be categorized into three types of dissimilarities, and they are:
   a) Sizes of unvoiced blocks are not identical.
   b) Positions of unvoiced blocks are shifted by (1201).
   c) Total number of unvoiced blocks is not identical.
2. Compression at level 64 kbps produces great difference in length of and number of unvoiced segments.

In this research, audio cover will be compressed at level 128 kbps, in such a case the similarity between the original and compressed file will be very enough to get robust hiding case.

Hiding text in unvoiced blocks consists of the two phases, and they are:

1. Embedding Module.
2. Extracting Module.

6.1 Embedding Module

This module is concerned with embedding secret message in the WAVE audio file; it consists of the following stages:

1. Ciphering the secret message.
2. Header isolation of original file (Cover).

3. Computation of short average energy.
4. Energy shifting.
5. Merge unvoiced blocks.
6. Quantization process.
7. Embedding process.

6.1.1 Ciphering the Secret Message

Data encryption before hiding will give more security immunity to the covered data, so if the attacker extracts the correct encrypted data from the stego object, he will discover that the extracted data have no meaning. In this case, the attacker will think that the extracted data is not correct or it is decrypted, the latter case means he should decrypt the encrypted data (i.e., adding another level of difficulty against attacker).

In the current work, a simple cryptosystem is adopted; it is a sort of stream cipher system. This system uses a key of length (18 characters), it is used to produce the required random numbers, and save them in stream buffer. Each random number in the stream buffer will be arithmetically combined with each character in the secret message by using XOR operation. The ciphered secret data is then converted to binary form, in order to be hided in the cover.

6.1.2 Header Isolation

In this stage the header is extracted from original audio file (cover) and the contents of isolated header has to be written into audio file (stego).

6.1.3 Computation of Short Average Energy

In this stage, the average energy is computed for each segment of audio data cover. The vector (Wav) represents data of the audio cover, it is segmented into the blocks (rectangular windows), and the average energy for each block is estimated.

The decision of whether a block is voiced or unvoiced is evaluated as follows:

Step 1: Determine

$$AE(i) = \sqrt{\frac{1}{w}\sum_{j=i}^{j=i+w}(Wav(j) - 128)^2} \quad \ldots..(4)$$

Step 2: If AE (i) < Threshold then

The block is unvoiced

Else

The block is voiced.

6.1.4 Energy Shifting

In the previous stage, the determination of whether a block (win) is voiced or unvoiced is accomplished. However, errors occur when this file is passed through MP3 compression. These errors may cause a decrease or increase in the energy of some blocks, which in turn will cause an error in the detection of (V/UV) blocks, i.e., block is detected as voiced before compression but after compression it is detected as unvoiced and vice-versa. The errors occur in blocks whose average energy values are very close to the threshold. To avoid the occurrence of overlapped margin caused by compression, we have to shift all the samples of the block whose energy is close to the threshold far away toward the voiced regions as illustrated in figure(2).

6.1.5 Merge Unvoiced Blocks

After allocating the value of threshold, the merging of voiced / unvoiced blocks (V=0/UV=1) will be accomplished. Figure (3) illustrates how the successive voiced blocks and successive unvoiced blocks are merged.

6.1.6 Quantization Process

These unvoiced blocks will be processed to be hosts for secret bits. The question arises here is how these blocks are ready to carry bits; the answer to this question is the lengths of the unvoiced blocks are quantized and then

modulated according to embedded secret bits. The type of quantization used here is uniform quantization.

The size of unvoiced blocks are quantized with an appropriate quantization step value "Q_Step", in order to produce gaps of the same distance between each two successive size of unvoiced blocks, this difference in the size will be the host space to hold the secret data. When sizes of unvoiced blocks are quantized using "Q_Step = Q_i ", then their new sizes after the quantization process will be multiple values of Q {0, $\pm Q_i$, $\pm 2Q_i$, $\pm 3Q_i$, $\pm 4Q_i$ …}. The equation used for quantization is:

$$Q\_size(j)=Q\_Step \times Round(Unvoice\_Info.Size(j) / Q\_Step) \quad ,\ldots\ldots..(5)$$

Where, Unvoice_Info.Size (j) is the size of j th unvoiced block.

Q_size (j) is the quantized size of new j th unvoiced block.

Q_Step is the quantization step value.

6.1.7 Embedding Process

Initially, the secret message is converted to string of bits (zeros and ones), then the embedding process (of secret bits) is done by adding or deleting "δ" value from the quantized length of the unvoiced block, the value of "δ" refers to the amount of decrement or increment in the length of unvoiced runs. The size of "δ" must be less than half of quantization step to make sure that addition or deletion does not cause a jump to previous or next adjacent quantization bins relative to the actual bin of the quantized length of the unvoiced blocks. Bit hiding is achieved by the addition or deletion of a fixed amount of unvoiced samples to the selected unvoiced host (blocks).

$$N_B = \begin{cases} B+\delta & \text{If embedded bit } =1 \\ B-\delta & \text{If embedded bit } =0 \end{cases}$$

Where, N_B is new length of the stego cover block.

B is length of the unvoiced host block.

δ = Win, Win is the minimum possible length of unvoiced hosts.

6.2 Extraction Module

This module is used for extracting secret message from WAVE audio, and it consists of the following stages:

1. Parse header.
2. Skip over part of data in stego file.
3. Computation of short average energy.
4. Voiced / Unvoiced blocks detection.
5. Quantization process.
6. Extracting bits.
7. Combine bits to construct the secret text
8. Deciphering.

In the "Parse Header" stage, the stego audio file is opened and scanned to find the end of the header. The audio data is skipped by 1200 byte, this is due to the results (differences) found when the original audio data is compared with corresponding construct data after it is compressed by using MP3 compression standard.

The (V/UV) blocks will be classified by applying the same sequence of the embedding module; each block of stego file is tested to find out whether it is voiced or unvoiced block.

In the next stage, quantization is performed on the length of unvoiced blocks to get its quantized value (by using the same value of quantization steps (Q_Step).

The secret bit is extracted by comparing the determined unvoiced block length with the corresponding quantized value, such that if Q_value is greater than the determined length of unvoiced block then the extracted secret bit is 0, otherwise secret bit is 1.

Finally, the string of binary bits is converted to bytes to give the secret message.

In the extraction phase, the deciphering algorithm is needed to decrypt the message after extracting it. The decryption algorithm is like the encryption algorithm, it is required to generate the random sequence using the same secret key. So, the receiver should request for the secret key (its length is 18 characters). This key will be used to generate random numbers that in turn will be XORed with the encrypted secret audio cover (stego object). The result from decryption process is the clear embedded message.

7. Experimental Results

Distortion measures are used to measure the amount of error in the stego audio, in other words, they are useful measures to compare between the stego audio and cover audio, and they offer a simple convenient tool for evaluating the information loss. Objective and subjective measures are applied on original audio cover and stego audio file.

Audio files, music, speech and song were used as test samples to assess the performance of the proposed hiding methods. The hiding of secret message in the audio cover is performed by using time modulation of the unvoiced regions of the cover signal. The following parameters are used as control parameters:

1. **Window size:** represents the size of audio data block that should be tested as voiced / unvoiced block. The size of the window could be 10, 20, 30, 40, or 50 samples.
2. **Threshold:** represents the level value used to distinguish unvoiced blocks from voiced blocks. The number could be 1, 2, or 3.
3. **Q_Step:** represents the value that is used in quantization process and it is equal to 3 × minimum block size.

Three audio files are adopted as test samples, Speech_1, Song_1 and Music_1. Tables (4, 5 and 6) show the capacity of the cover in bps, error bits and quality of the stego cover after embedding process.

Table (7) shows the objective and subjective test results performed to evaluate the performance of the hiding method based on time modulation of the unvoiced blocks.

The size of the hidden data is taken to be (1322 bits). In all these test cases, the window size is (20 samples) and threshold value is (1).

Figure (4) shows the audio signal before and after hiding by using time modulation.

7. Conclusion

In this paper, secret data is embedded in WAVE audio file. The secret data hidden into audio data cover resist against the deformations formed by MP3 compression. The secret bits are embedded either by shortening or elongating the time length of unvoiced blocks. Transparency requirement is achieved by optimizing control parameters such as block size, threshold and quantization step. Robustness requirement is achieved by capability of surviving against MP3 compression at level 128 kbps.

9. References

[1] Bender W., Gruhl D., Morimoto N., Lu A., **"Techniques for Data Hiding "**, IBM System Journal, Vol. 35, No. 3&4,1996,URL: http://isj.www.media.mit.edu/isj/SectionA/313.pdf.

[2] Supurovic P., **"MPEG Audio Compression Basics"**, URL: http://www.chested.chalmers.se\~kf96svgu\, 1998.

[3] Fridrich J., **"Applications of data hiding in digital Images"**, Tutorial for the (ISPACS'98) Conference, Melbourne, Australia, November 4-6, 1998, URL: http://www.ssie.Binghamton.edu/~jirif.

[4]Campisi P., Kundur D., Hatzinakos D., Neri A., **"Compressive Data Hiding: An Unconventional Approach for Improved Color Image**

Coding", Eurasip Journal on ASP, 30 April, 2001, URL: http://www.comlab.uniromaz.it/prgetti.htm

[5] Bouman C. A., **"EE438-Laboratory 9: Speech Processing"**, Purdue University: EE438- Digital Signal Processing with Application, October 16, 1998.

[6] Cassidy S., **"COMP449: Speech Recognition"**, Department of Computing, Macquarie University, Australia, 2002, URL: www.comp.mq.edu.au/~cassidy/comp449/html/comp449.html

[7] Levitt, Jason. "Getting Ahead Of The Privacy Curve With Steganography". URL: http://www.iweek.com/author/internet9.htm (3 March 2003).

[8] Sellars D., "**An Introduction to Steganography**", University of Camberge,2003,URL: http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html

[9] Johnson N.F., Duricn Z., Jajodia S., **"Information Hiding: Steganography and Watermarking Attack and Countermeasurments"**, Kluwer Academic Publishers, USA, 2001.

Table (1) Audio file size before MP3 compression and after MP3 decompression.

Test File	WAVE	MP3 Encoder			MP3 Decoder		
	Size (Byte)	64 kbps	96 kbps	128 kbps	64 kbps	96 kbps	128 kbps
F1	69,876	14,527	20,901	27,48	73,798	72,646	72,646
F2	1,728,642	315,457	471,983	628,927	1,732,708	1,730,404	1,730,404
F3	10,895,404	1,978,308	2,966,572	3,955,046	10,899,142	10,897,990	10,897,990

Table (2) Number of unvoiced bytes before and after MP3 compression.

Test File	Number of Unvoiced Byte in the Original File	Number of Unvoiced byte in the Reconstructed File		
		64 kbps	96 kbps	128 kbps
F1	24325	25406	26000	26377
F2	169891	279481	170594	172128
F3	367807	286188	374787	373162

Table (3) Unvoiced blocks before compression and after decompression for test file (*F1*) at compression level 128 kbps.

Before compression				After compression			
Index	**Start**	**End**	**Size**	**Index**	**Start**	**End**	**Size**
1	0	479	480	1	0	1680	1681
2	481	537	57	2	1682	1738	57
3	539	609	71	3	1740	1810	71
4	611	647	37	4	1812	1848	37
5	650	795	146	5	1851	1996	146
6	797	879	83	6	1998	2080	83
7	881	927	47	7	2082	2128	47
8	929	985	57	8	2130	2186	57
9	987	1041	55	9	2188	2242	55
10	1043	1211	196	10	2244	2412	196
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
Number of Unvoiced Blocks = 2753				**Number of Unvoiced Blocks = 2636**			

Table (4) Results of audio Steganography by using time modulation applied to Speech_1 test sample.

Control Parameters			Hiding Rate (bps)	Total Reconstructed Bit	Wrong Reconstructed Bit	Quality of Audio
Window	**Threshold**	**Q_Step**				
10	1	30	0.0021	4196	50	Excellent
20	1	60	0.0008	1578	0	Excellent
30	1	90	0.0005	977	0	Excellent
40	1	120	0.0003	681	0	Excellent
50	1	150	0.0003	531	0	Excellent
10	2	30	0.0032	6414	44	Excellent
20	2	60	0.0014	2771	18	Excellent
30	2	90	0.0008	1580	0	Excellent
40	2	120	0.0005	1091	0	Excellent
50	2	150	0.0004	819	0	Excellent
10	3	30	0.0041	8271	42	Excellent
20	3	60	0.0018	3575	0	Excellent
30	3	90	0.0010	2071	14	Excellent
40	3	120	0.0007	1322	16	Excellent
50	3	150	0.0005	951	44	Excellent

Table (5) Results of audio Steganography by using time modulation applied to Song_1 test sample.

Control Parameters			Hiding Rate (bps)	Total Reconstructed Bit	Wrong Reconstructed Bit	Quality of Audio
Window	Threshold	Q_Step				
10	1	30	0.0001	276	0	Excellent
20	1	60	0.00005	113	0	Excellent
30	1	90	0.00003	59	0	Excellent
40	1	120	0.00003	46	0	Excellent
50	1	150	0.00001	32	0	Excellent
10	2	30	0.00021	467	47	Excellent
20	2	60	0.00008	184	30	Excellent
30	2	90	0.00004	91	0	Excellent
40	2	120	0.00003	70	0	Excellent
50	2	150	0.00001	43	0	Excellent
10	3	30	0.00029	658	53	Excellent
20	3	60	0.00013	299	30	Excellent
30	3	90	0.00008	1754	12	Excellent
40	3	120	0.00005	107	0	Excellent
50	3	150	0.00003	64	0	Excellent

Table (6) Results of audio Steganography by using time modulation applied to Music_1 test sample.

Control Parameters			Hiding Rate (bps)	Total Reconstructed Bit	Wrong Reconstructed Bit	Quality of Audio
Window	Threshold	Q_Step				
10	1	30	0.01207	3370	47	Excellent
20	1	60	0.00307	856	0	Fair
30	1	90	0.00121	337	0	Fair
40	1	120	0.00057	160	0	Fair
50	1	150	0.00027	74	0	Bad
10	2	30	0.02266	6292	46	Fair
20	2	60	0.00934	1253	48	Bad
30	2	90	0.00540	1506	0	Bad
40	2	120	0.00342	953	0	Bad
50	2	150	0.00219	610	0	Bad
10	3	30	0.02644	7287	49	Bad
20	3	60	0.01319	3666	46	Bad
30	3	90	0.00871	2430	0	Bad
40	3	120	0.00597	1661	29	Bad
50	3	150	0.00437	1221	45	Bad

Table (7) Objective and subjective test results of audio hiding method based on time modulation method.

Sample	Size of MP3	Hiding Rate (bps)	MAE	MSE	PSNR	SNR	Impairment Scale
Speech_1	2,001,505	0.00066	17.51	894.25	18.62	12.75	Imperceptible
Song_1	2,241,913	0.00005	40.90	2962.49	13.45	7.85	Imperceptible
Music_1	279,095	0.0019	4.62	38.35	32.29	26.33	Slightly annoying

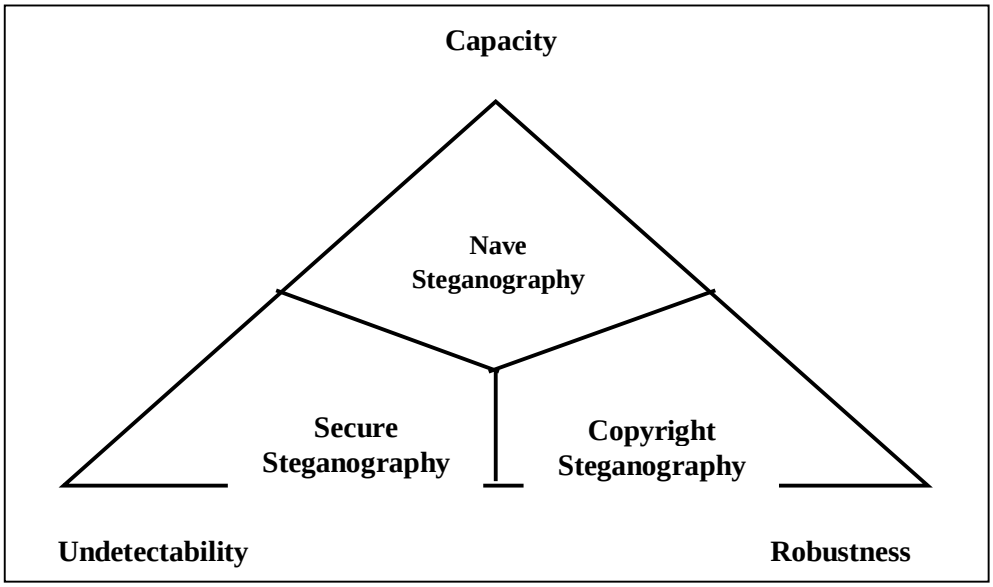


Figure (1) Trade-off among undetectability, capacity and robustness [9].

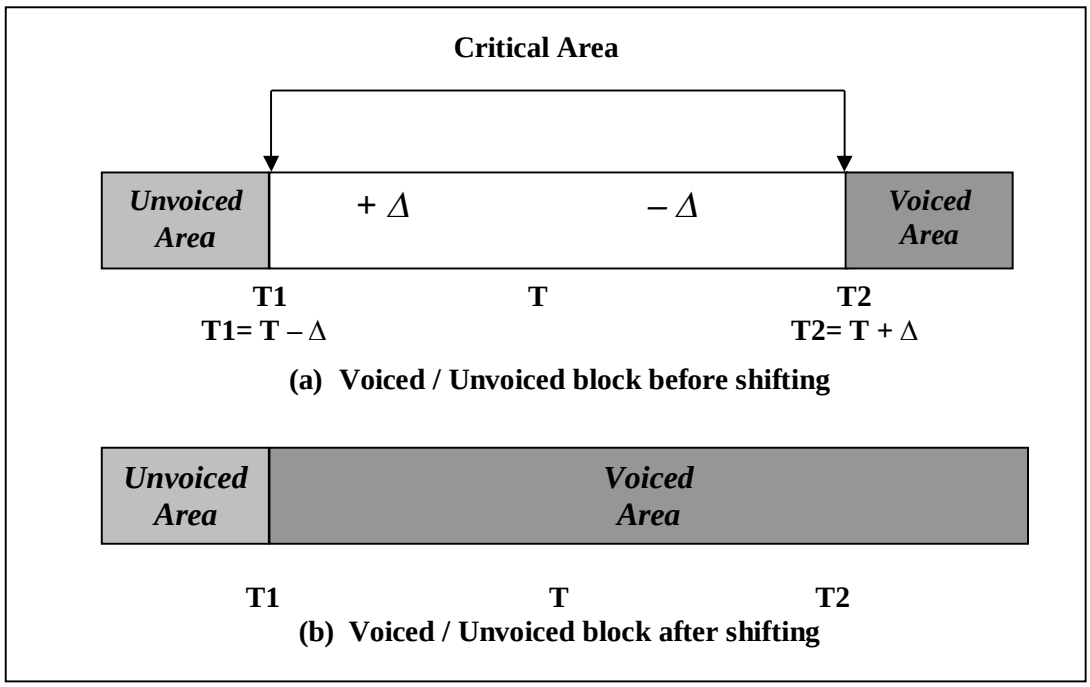


Figure (2) Shifting of sample amplitude

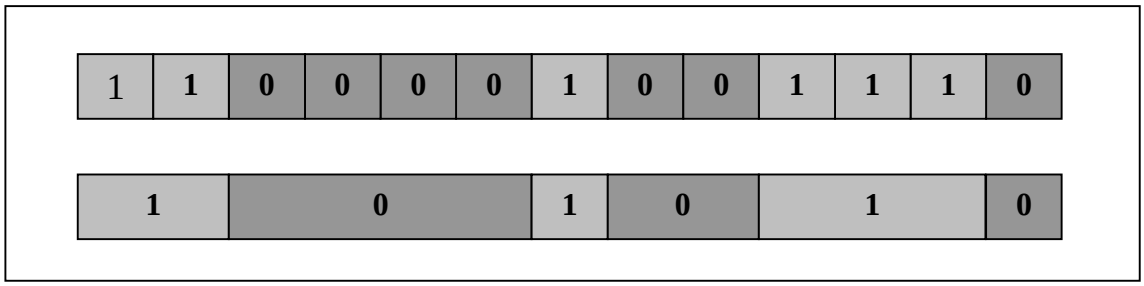
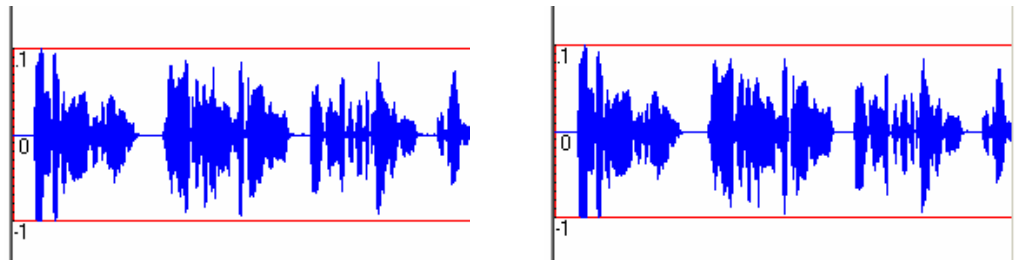
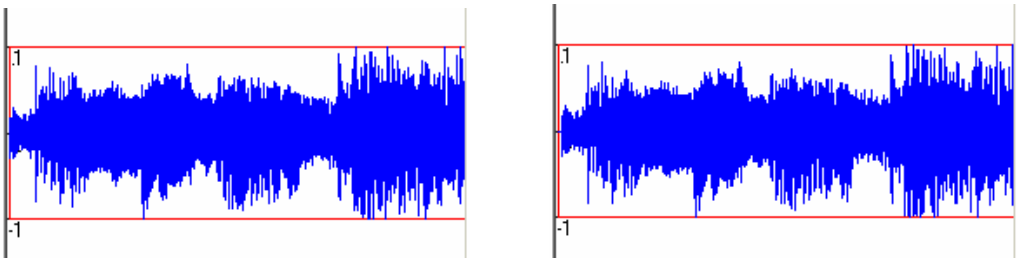


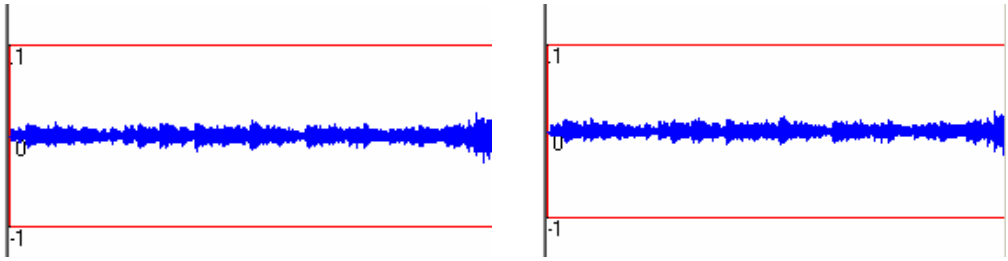
Figure (3) Merging unvoiced successive blocks



(a) Speech_1



(b) Song_1



***(c)* Music_1**

Figure(4) The test samples Speech_1, Song_1 and Music_1 signal before and after hiding by using time modulation method.