# Implementation of Audio Steganogarphy Using C#

**Author**

**Vinay Yadav[1], Sumeet Gupta[2], Vijay Jaiswal[3]**

*[1](Asst. Professor/Department of CSE/SR Group of Institutions, Lucknow)*
*[2](Asst. Professor /Department of CSE/SR Group of Institutions, Lucknow)*
*[3](Research Scholar /Department of CSE/UPTU, Lucknow)*

**Abstract:** *Audio steganography is focused in hiding secret information in an innocent cover audio file or signal securely and strongly. Communication security and robustness are vital for transmitting important information to authorized entities, while denying access to not permitted ones. By embedding secret information using an audio signal as a cover medium, the very existence of secret information is hidden away during communication. This is a serious and vital issue in some applications such as battlefield communications and banking transactions. In a computer-based audio steganography system, secret messages are embedded in digital sound. The secret message is embedded by slightly altering the binary sequence of a sound file. Existing audio steganography software can embed messages in WAV, AU, and even MP3 sound files*.
**Keyword:** *Sound , Least Significant Bit (LSB), Analogue Signal, human visual system (HVS), WAV, AU, and even MP3 sound files, Spectrum, Encryption, Decryption*.

**Introduction:** Steganography is the study of techniques for hiding the existence of a secondary message in the presence of a primary message. Steganography, coming from the Greek words stegos and it means roof or covered and graphia which means writing, is the art and science of hiding the fact that communication is taking place. The primary message is referred to as the carrier signal or carrier message; the secondary message is referred to as the payload signal or payload message. Generally, in steganography the following operations are performed:

1) Write a non-secret cover message.
2) Produce a stego-message by concealing a secret embedded message on the cover message by using a stego-key.
3) Send the stego-message over the insecure channel to the receiver.
4) At the other end, on receiving the stego-message, the intended receiver extracts the secret embedded message from the stego-message by using a pre agreed stego-key.

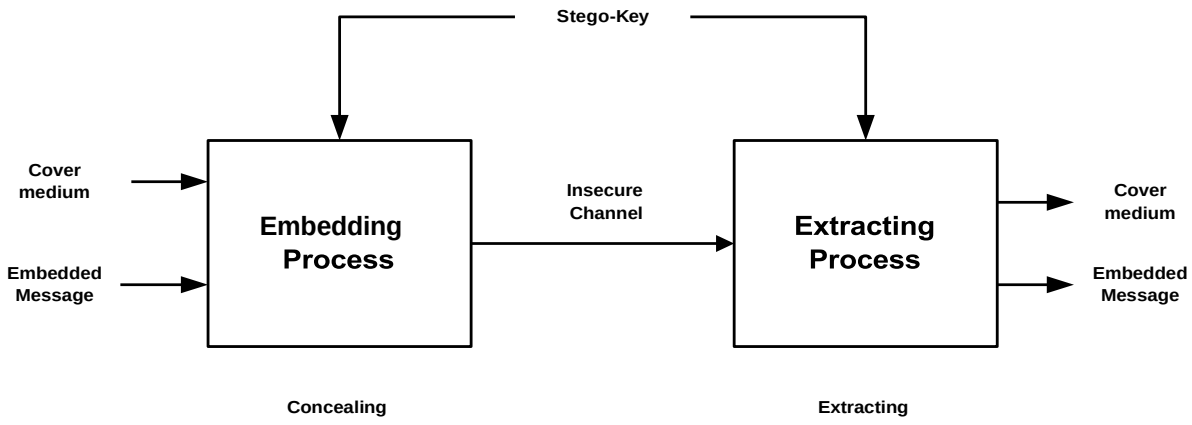


***Figure 1 General Steganography System***

Information hiding techniques are receiving much attention today. The main motivation for this is largely due to fear of encryption services getting illegal, and copyright owners who want to track confidential and intellectual property copyright against unauthorized access and use in digital

materials such as music, film, book and software through the use of digital watermarks. Advance security is not maintained by the password protection but it is gained by hiding the existence of the data which can only be done by Steganography. Steganography is "data hiding" technique, In the modern day sense of the word usually refers to information or a file that has been concealed inside a digital Picture Video or Audio file. This is derived from the lizard "Stegosaurs" covered or secret and graphy meaning writing or drawing. Therefore steganography literally means covered writing. It simply takes one piece of information and hides it within another. It is not intended to replace cryptography but supplement it. Hiding a message with Steganography methods reduces the chance of a message being detected.

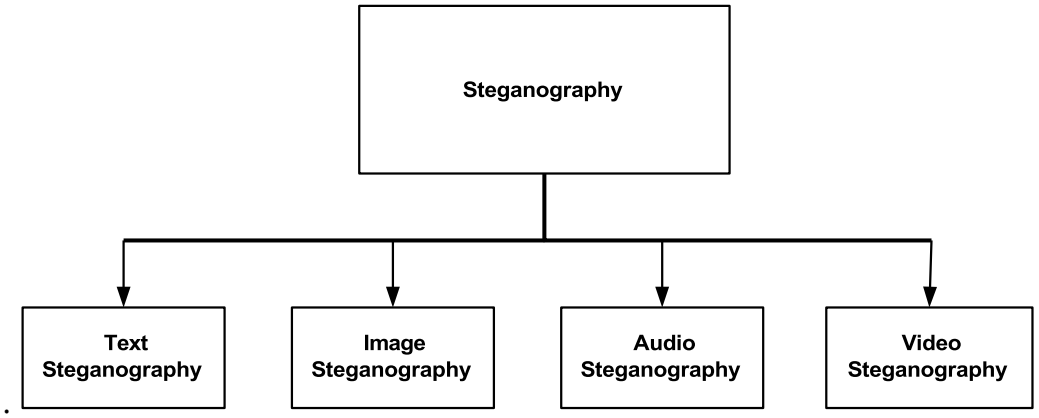


**Figure 2 : Types of Steganography**

## Types of Steganography

In a computer-based audio steganography system, secret messages are embedded in digital sound. The secret message is embedded by slightly altering the binary sequence of a sound file. Existing audio steganography software can embed messages in WAV, AU, and even MP3 sound files. Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media, such as digital images. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced[17].

1. LSB Encoding:
2. Parity Coding:
3. Phase coding:
4. Spread spectrum:
5. Echo Hiding:

## The Idea

After recording a sound onto an audio cassette, and re-recording it into a Wave file, we cannot rely on the binary data to be the same or in any way similar. So, changing a few bits is not enough anymore. We have to change the sound in a way we can recognize, even behind loads of noise and arbitrary changes.
Your first idea may be to insert a very short *beeeep* every *n* seconds, where *n* is a byte from the secret message. The recipient can extract the noises with a band pass filter, put the intervals into a stream, and just read the message from it. But trying that, you'll soon run out of tape:

**A = 65**
**B = 66**
**C = 67**
**ABC = 65 + 66 + 67 = 198**

Inserting a recognizable frequency in intervals that stand for the hidden bytes, we would need 198 seconds only for a short message like *ABC*.
Your second idea may be to split each secret byte into high and low half-byte, so that the maximum interval between two *beeps* is 15, and no byte can fill more than 30 seconds on the tape. For example, the character "z", that would block 122 seconds by the first try, needs only 17 seconds this way:

z = **122** = 1111010
half bytes = **7** (0111) and **10** (1010)

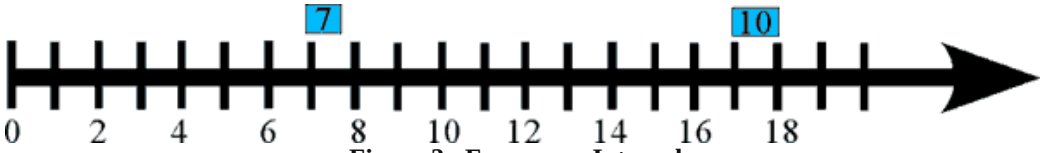


**Figure 3 : Frequency Intervals**

That is exactly what this article's application does. It allows you to search the carrier wave for an unused or low-volume frequency, and inserts very short, hardly hearable noises of just that frequency. You can then play the result, and record it with the tape recorder. To extract the hidden message, you just play the tape, record the sound with an audio recorder software (such as Gold Wave etc.), and remove the silence from beginning and end. Then you can open the file with this application, enter the frequency that had been used for hiding, and watch as a band pass filter isolates the *beeps* and your message gets re-constructed.

## How it Works

### 1) Hide a Message

While hiding a message, the user performs five steps:

1) Select a Wave file, and enter the secret message.
2) Check if the message fits into the sound, and shorten it, if necessary.
3) Guess a frequency that occurs only in low volume, or not at all.
4) If "Check sound" produces a warning, raise the threshold volume or frequency until the warning disappears.
5) Write the new sound.
6) Play the new sound, and record it onto the tape.

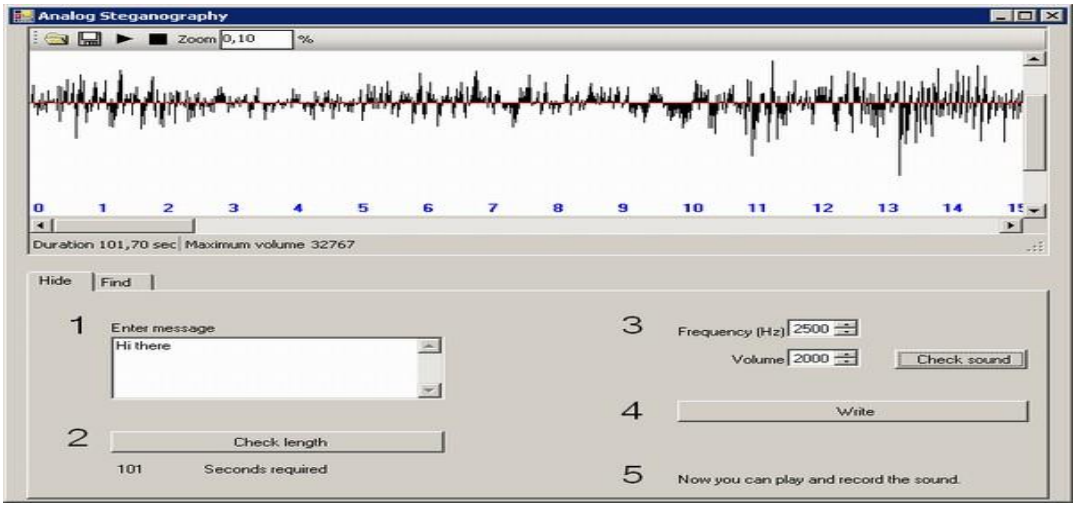


**Figure 4 : Hiding a Message**

Steps three and four are not self-explaining. Let's start with step three: Checking a frequency for existence and maximum volume. The user guesses a frequency and clicks the "Check sound" button. To check if the highest amplitude of such a frequency is lower than the selected volume value (that means, we can use the selected combination), we first have to isolate the frequency with a band pass filter. SoX does that for us. Then we compare the result's samples to the selected maximum amplitude (call it volume, that's nearly the same in this case), and count the samples that are too loud.

### C# Function to Hide message

```
public void Hide(Stream message, int frequencyHz, int volumne)
{
   Stream preparedMessage = PrepareMessage(message);
   int messageByte;
   int offset = 0;
   while ((messageByte = preparedMessage.ReadByte()) > -1)
         {
               offset += messageByte;
               InsertBeep(offset, frequencyHz, volumne);
         }
}
```

## 2) Extract a Message

Before reading a hidden message, the user has to filter the recorded sound. If the tape player added very bad noise in just our frequency, so that wrong *beeps* are detected, those errors can be deselected.

1) Enter the frequency of the expected *beeps*, and band pass filter the sound. The second filter button - threshold volume - is not really necessary. You can use it to remove samples from the graphic, that will anyway be treated as silence.
2) Find the noises. A *beep* is a group of samples that are greater than the selected threshold volume. In the graphic, the beginning and end of each detected *beep* are marked with red lines. The Checkboxes allow you to exclude single *beeps* from evaluation, if you're sure that they don't belong to the message.
3) Read the message. The last step lists the intervals between the selected noises, and re-constructs the hidden message.

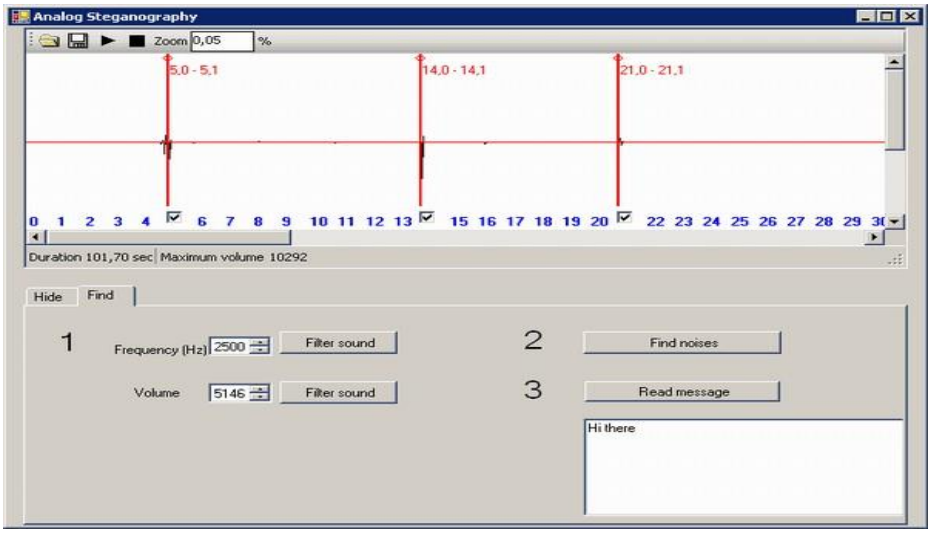


**Figure 5 : Extract a Message**

"Filter sound" applies the same band pass filter we already know. This time, we don't count big samples, but open and display the filtered wave.

### C# Function to Find Message

```
public void FindAnything(short tolerance) {
   //size of scan window in samples
   int scanWindowSize = waveSound.Format.SamplesPerSec / beepLength;
   //size of scan window in seconds
   float scanWindowSizeSeconds = (float)scanWindowSize /
            (float)waveSound.Format.SamplesPerSec;

   int startIndex = -1;
   int endIndex = -1;
   int countSilentSamples = 0;
   for (int n = 0; n < waveSound.Count; n++) {
      if (Math.Abs(WaveSound[n]) > tolerance) { //found a sound
         countSilentSamples = 0;
         if(startIndex < 0){
            startIndex = n;
         }
      } else if (startIndex > -1) { //searched and found silence
         countSilentSamples++;
         if (countSilentSamples == scanWindowSize) {
            endIndex = n - scanWindowSize;

            //tell the caller to mark a found beep in the wave
            NotifyOnBeep(startIndex, endIndex, scanWindowSizeSeconds);

            //scan next time window
            countSilentSamples = 0;
            startIndex = -1;
         }
      }
   }

   if (startIndex > -1) { //wave ends with a beep
      NotifyOnBeep(startIndex, waveSound.Count-1, scanWindowSizeSeconds);
   }
}
```

### Advantages of Audio Steganography

1. Audio based Steganography has the potential to conceal more information:
   - Audio files are generally larger than images
   - Our hearing can be easily fooled
   - Slight changes in amplitude can store vast amounts of information
2. The flexibility of audio Steganography is makes it very potentially powerful :
   - For example, two individuals who just want to send the occasional secret message back and forth might use the LSB coding method that is easily implemented. On the other hand, a large corporation wishing to protect its intellectual property from "digital pirates" may consider a more sophisticated method such as phase coding, SS, or echo hiding.
3. Another aspect of audio Steganography that makes it so attractive is its ability to combine with existing cryptography technologies.

- Users no longer have to rely on one method alone. Not only can information be encrypted, it can be hidden altogether.

4. Many sources and types makes statistical analysis more difficult :
   - Greater amounts of information can be embedded without audible degradation

## Disadvantages of Audio Steganography

1. Embedding additional information into audio sequences is a more tedious task than that of images, due to dynamic supremacy of the HAS over human visual system.
2. Robustness: Copyright marks hidden in audio samples using substitution could be easily manipulated or destroyed if a miscreant comes to know that information is hidden this way.
3. Commercialized audio Steganography have disadvantages that the existence of hidden messages can be easily recognized visually and only certain sized data can be hidden.
4. Compressing an audio file with lossy compression will result in loss of the hidden message as it will change the whole structure of a file. Also, several lossy compression schemes use the limits of the human ear to their advantage by removing all frequencies that cannot be heard. This will also remove any frequencies that are used by a Steganography system which hides information in that part of the spectrum

## Conclusion

Steganography is an information hiding technique where secret message is embedded into unsuspicious cover signal. An effective audio steganography scheme should possess the following three characteristics: Inaudibility of distortion (Perceptual Transparency), Data Rate (Capacity) and Robustness. These characteristics (requirements) are called the magic triangle for data hiding.
We have presented a high capacity and high stego-signal quality audio steganography scheme based on samples comparison in DWT domain where selected coefficient of a segment are compared with pre determined threshold value T and based on comparison bits are embedded. The strength of our algorithm is depend on the segment size and their strength are enabled the algorithm to achieve very high embedding capacity for different data type that can reach up to 25% from the input audio file size with lest of 35 dB SNR for the output stego signal.

## References

[1]. R. C. Gonzalez and R. E. Woods, Digital Image Processing (2nd Edition), Prentice Hall, January 2002.
[2]. S. G. Mallat, \A theory for multiresolution signal decomposition: the wavelet representation," vol. 11, pp. 674{693, July 1989.
[3]. J. M. Shapiro, \Embedded image coding using zerotrees of wavelet coficients," vol. 41, pp. 3445{3462, Dec. 1993.
[4]. A. Said and W. A. Pearlman, \A new, fast, and e_cient image codec based on set partitioning in hierarchical trees," vol. 6, pp. 243{250, June 1996.
[5]. D. Taubman, \High performance scalable image compression with ebcot," in Proc. International Conference on Image Processing ICIP 99, vol. 3, pp. 344{348, Oct. 24{28, 1999.
[6]. Y.-S. Zhang, \Multiresolution analysis for image by generalized 2-d wavelets," Master's thesis, 2008.
[7]. I. Daubechies, Ten lectures on wavelets, vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1992.
[8]. S. Mallat, A Wavelet Tour of Signal Processing, 3rd ed., Third Edition: The Sparse Way. Academic Press, 3 ed., December 2008.
[9]. I. Daubechies, \Orthonormal bases of compactly supported wavelets, "Communications on Pure and Applied Mathematics, vol. 41, no. 7, pp. 909{996, 1988.
[10]. P.-S. T. Tinku Acharya, JPEG2000 Standard for Image Compression, ch. Coding Algorithms in JPEG2000, pp. 163{196. 2005.
[11]. Mei-Yi, W., Yu-Kun, H. , Jia-Hong, L. (2004): An Iterative Method of Palette-Based Image Steganography, Journal of Patern Recognition Letters, Vol (25).