An Efficient Method of Audio Steganography with GUI

Pushpa B[1]
CSE, GRIET
Hyderabad, India

Pittala Omkar
CSE, GRIET
Hyderabad, India
omkaroln@gmail.com

Vinay Kumar Reddy Seelam
CSE, GRIET
Hyderabad, India
vinaykumarreddy920@gmail.com

Chiliveru Bhargav
CSE, GRIET
Hyderabad, India
bhargavch2984@gmail.com

Shirdish Mohan Dodle
CSE, GRIET
Hyderabad, India
shirdishdodle14@gmail .com

***Abstract*— Steganography is the art and science where the writing messages are hidden in such a way that no user at the end side knows exactly what the message delivered. Data hiding is the characteristic related to object-oriented programming where the objects are associated with the data that is having a predefined template. Thus, all data not required by an object can be said to be "hidden". The word "Steganography" is of Greek origin and means "covered, or hidden writing". An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen. Steganography used in electronic communication includes steganographic coding inside of a transport layer, such as a WAV file, or a protocol, such as UDP. The advantage of steganography over cryptography alone is that messages do not attract attention to themselves, to messengers, or to recipients. A steganographic message (the plaintext) is often first encrypted by some traditional means, and then a hidden text is modified in some way to contain the encrypted message (ciphertext), resulting in stegotext.**

Keywords—Stenography, Hiding Messages, Microsoft net, GUI, UDP.

I. INTRODUCTION (*HEADING 1*)

The human auditory system (HAS) operates across a wide dynamic range, making data hiding in audio signals particularly difficult. The HAS can detect signals with a power of more than one billion to one and a frequency of more than one thousand to one [1]. Additive random noise sensitivity is similarly high. In a sound file, disturbances as small as one part in ten million can be detected (80 dB below ambient level). However, there are some "gaps" in the system. While the has a broad dynamic range, its differential range is somewhat minimal. As a result, big noises tend to obscure quiet noises. Furthermore, the HAS can only discern the relative phase and not the absolute phase. Finally, some environmental distortions exist [2].

II. EXISTING SYSTEMS

Digital image security is important in all professions, but notably insensitive domains like the military. The military and medical worlds are two worlds that have a lot in common. With the rise of cloud computing, the advancement of information technology has resulted in the development of computers. where there are major security issues including secrecy and authentication Illegal actions such as cybercrime pose a persistent danger to national security and integrity. Information hacking, copying, or criminal use [3]. However, Steganography is the art (and science) of communicating in an unobtrusive manner. In military jargon, this is known as "transmission security," or TRANSEC. Its purpose is to hide messages inside other "harmless" messages in such a way that an "enemy" would be unaware of the presence of a second secret message [4]. Another survey conducted [5] It's no wonder that steganography has moved into the digital era as more data is stored on computers and exchanged via networks. Steno apps allow anyone to disguise any form of a binary file into a variety of other binary files on computers and networks, while image and music files are the most frequent carriers nowadays. Other studies proposed by Manohar and Peetla [6] show the encryptions and decryptions for data through stenography. The authors elaborate that in the process of hiding the data.

The first phase is video steganography involves grabbing any video stream file, embedding it with a cover file, and then converting it to a steno file with encryption and decryption before sending it to the recipient. The data is extracted from the steno file into a video file and hidden data by the receiver using steno tools [2,4]. The flow of covert communication with the quality of data is contained in the data or information. The sender and receiver sides of communication are both involved in steganography. There were also numerous steganography tools and file types available. However, in the present study, the “Data Hiding in Audio Files” was mainly developed to embed or extract the messages into audio files. This project basically deals with two important network security concepts namely steganography and encryption. For encryption, the AES algorithm is used. The plaintext is given as input [3,5]. The plaintext is encrypted using the AES algorithm. The ciphertext is given as output. The output ciphertext is hidden into the audio files using Steganography. For steganography, the LSB algorithm is used. The audio file is in the wave format is the chosen medium to conceal and transmit the secret information. Since the audio file is in ASCII format, the contents of the text file are also converted to the bitstream. The encrypted file is now embedded behind the audio file by mixing the contents together using the LSB algorithm [6]. At the other end, the encrypted file is separated from the audio file. The encrypted file is then decrypted and the original text file contents are then viewed.

III. PROPOSED SYSTEM

A. *Data Hiding in the Audio Files*

In this proposed work the "Data Hiding in Audio Files" project is developed where firstly creating and extracting the information messages from the audio files. This project focuses on steganography and encryption, two fundamental network security concepts. The AES algorithm is used for encryption. As input, the plaintext is provided. The AES technique is used to encrypt the plaintext. The output is the encryption text. Steganography is used to hide the output ciphertext in the audio files. The LSB algorithm is used for steganography. The chosen medium for concealing and transmitting the secret information is an audio file in wave format. The contents of the text file are also translated to the bitstream because the audio file is in ASCII format. The audio file now has the encrypted file embedded behind it.

System Requirement:

Hardware Specifications:

Processor	: Pentium IV (Minimum)
RAM	: 1 GB min
CPU Clock	: 1.6 GHz

Software Specifications:

Front End	: C# and .NET
Platforms	: Windows
Webservers	: HS 6.0

B. *Software Implementation and Tools Used*

In this work, the author used three major tools to create the GUI for the proposed system.

1) *Microsoft .Net*

In February 2002, Microsoft released the.net (pronounced dot net) framework. It's the company's largest project since the introduction of Windows in 1991. Net is a ground-breaking multi-language platform that connects different areas of application development to the Internet. Above the operating system, the framework covers all layers of program development. Microsoft will create a variety of software to achieve this purpose [7]. Every actor in the industry, whether a software developer or a device manufacturer, must adopt.Net in order to be integrated. The.Net initiative aims to make data transmission across networks, PCs, and devices as simple as possible, regardless of the platforms, architecture, or solutions used. Many of the best ideas in the world have been taken by the Microsoft industry, as well as some concepts of their own, and combined into a single, cohesive package. Microsoft's next-generation platform for developing web apps and web services is known as Net. It is a Microsoft platform for XML web services areas [8].

2) *.NET Framework*

The.Net Framework contains classes, interfaces, and value types that aid in the development process and provide access to system functionality. C# was built from the bottom up to work with Microsoft's new.NET Framework [9]. The Common Language Runtime, a set of class libraries, a set of programming languages, and the ASP.NET environment make up the.NET Framework. The.NET Framework was created with three main objectives in mind. For starters, it was designed to make Windows applications far more dependable while also delivering a higher level of security. Second, it was created to make it easier to create Web apps and services that function not only on desktop computers but also on mobile devices [10].

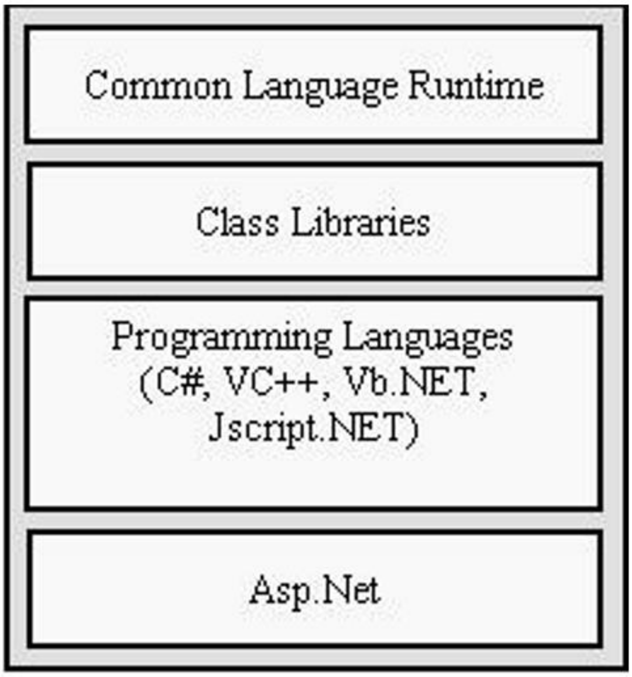


Fig. 1. Components of .Net Framework.

3) *C - Sharp*

C# (pronounced C-Sharp) is a new programming language that came with the Microsoft.NET framework and is without a doubt the preferred language in the.NET environment. It was first developed as part of Microsoft's. NET strategy in the late 1990s. It's a whole new language with no backward compatibility issues and a slew of interesting and promising new features [11]. It's an Object-Oriented Programming language having key similarities to Java, C++, and Visual Basic.

Reflections, attributes, marshalling, remoting, threads, streams, data access using ADO.NET, and other new/exciting features are available in C#. The C# programming language was created from the ground up with the Microsoft.Net environment in mind [12]. The Common Language Runtime (CLR), which offers runtime support to MS.Net (and consequently C#) programs, runs on top of it.

4) *Features*

Data-hiding strategies should be able to embed data in a host signal while adhering to the following constraints and features:

- The embedded data should be barely discernible and the host signal should be non-objectively degraded. The idea is to keep the data secret, it is feasible to hide anything while it is still visible; all you have to do is block the person from looking at it. The terms concealed, inaudible, imperceivable, and invisible refer to the fact that an observer does not perceive the data's presence, even if it is perceptible.

- Embedded data should be directly encoded into the media, rather than in a header or wrapper, to ensure that the data is preserved across different data file formats [13].
- Channel noise, filtering, resampling, cropping, encoding, lossy compressing, printing and scanning, digital-to-analog (D/A) conversion, and analog-to-digital (A/D) conversion, among other things, should be immune to modifications ranging from intentional and intelligent attempts at removal to anticipated manipulations.
- Because the goal of data hiding is to maintain the data in the host signal, not necessarily to make it difficult to access, asymmetrical coding of the embedded data is preferable [14].
- To ensure data integrity, error-correcting coding should be utilized. When the host signal is changed, it is unavoidable that the embedded data suffer some deterioration.
- Self-clocking or arbitrarily re-entrant data should be embedded. This ensures that when just portions of the host signal are available, such as when a sound bite from an interview is taken, data stored in the audio segment can be recovered.
- Because there is no need to refer to the original host signal, this feature also aids the automatic decoding of the hidden data [15].

5) Applications

There are many applications in which data hiding techniques are used. The amount of embedded data and the degree of immunity to host signal alteration are both trade-offs. A data-hiding approach can operate with either a high embedded data rate or a high resistance to alteration, but not both, by limiting the degree of host signal deterioration. When one rises, the other must fall. While some data-hiding systems, such as a spread spectrum, can be demonstrated analytically, it appears to hold for all data-hiding systems. Redundancy can be used to trade bandwidth for robustness in any system. The amount of embedded data and the degree to which the host signal is modified varies by application. As a result, different approaches are used for various applications. In this part, a few potential uses for data concealing is discussed.

- The placement of a digital watermark is an application that only requires a little quantity of embedded data [16]. The embedded data serves the same purpose as an author's signature or a company emblem in that they are utilized to place an indicator of ownership in the host signal. The coding techniques utilized must be resistant to a wide range of possible modifications because the information is essential and the signal may confront sophisticated and intentional attempts to destroy or erase it.
- Tamper-proofing is a second application for data concealing. It is used to indicate that the host signal's authored status has been changed. The presence of changes to the embedded data shows that the host signal has been altered [17].
- A third use is the feature location, which necessitates the inclusion of additional data. Embedded data is buried in precise spots within an image in this application. Individual content elements, such as the name of the person on the left vs the right side of an image, can be identified. In most cases, feature location data is not intentionally removed. The elaborated use of the application can be seen in the study conducted by [18]
- • However, it is expected that the host signal will be altered in some way; for example, images are frequently scaled, cropped, and tone-scale enhanced. As a result, feature location data-hiding strategies must be resistant to both geometrical and nongeometrical host signal alterations.

IV. SYSTEM DESIGN

In this work, the author considered a number of subsystems along with various phases of the system development lifecycles. This system lifecycle helps to implement the system coding in a smooth way to avoid any kind of complications and this allows the software engineer to avoid errors. The process can be seen and elaborated in Fig. 2 for the proposed system.

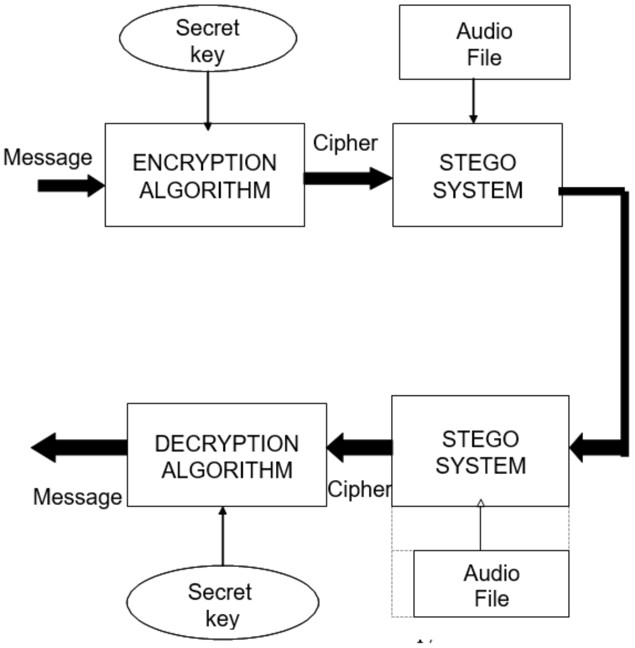


Fig. 2. Proposed System Process.

1) Encryption

Encryption or enciphering is the process of transforming plaintext to cypher text. The original communication is the plaintext, and the unreadable message is the ciphertext. The encryption algorithm generates the ciphertext from the secret key and plaintext. However, on plaintext, the encryption algorithm makes numerous substitutions and transformations.

2) Decryption

Decryption or decoding is the process of recovering the plaintext from the ciphertext. The decryption algorithm creates the original plaintext by combining the cypher text with the secret key.

The Data flow here is shown through two levels zero and one. The flow in level zero can be seen in Fig. 3. there are two stages through which the file goes through. Firstly, the Encryption and again decryption of the file. However, the process is elaborated in Fig.4. The secret keys related at all stages ensure secure transmission and files security as well. The proposed flow diagram is efficient enough to be considered for encryption and data hiding.

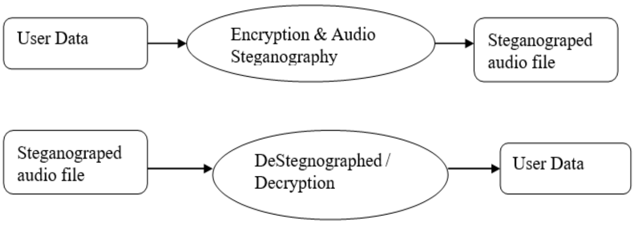


Fig. 3. Representation of Level 0 DFD.

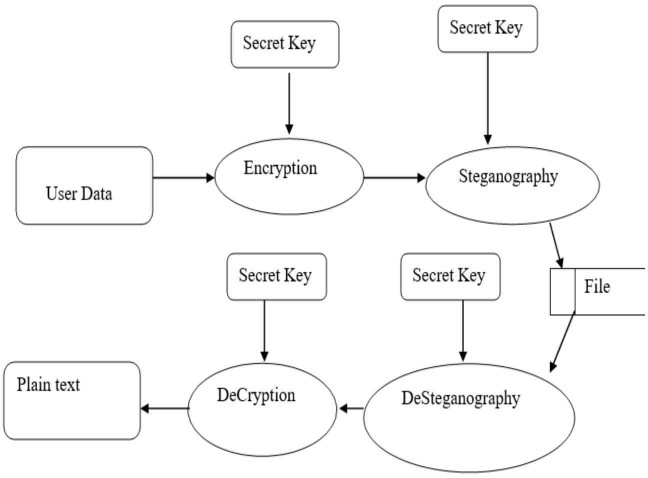
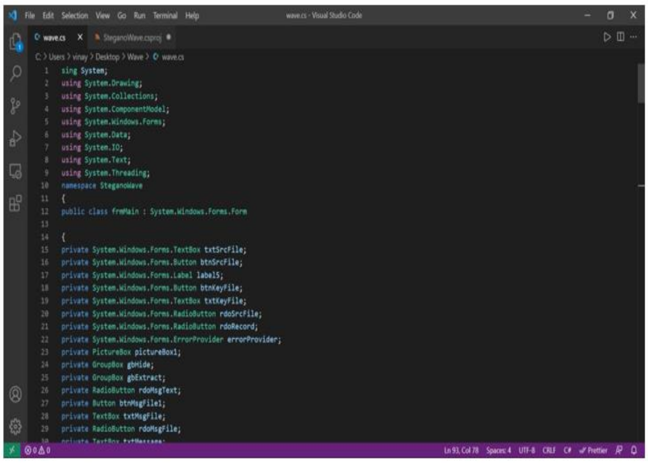


Fig. 4. Representation of Level 1 DFD.

Hence, the model handles the files for encryption and decryption in the system. These diagrams make it easy to understand and analyze the process in the proposed system.

V. IMPLEMENTATION OF THE PROPOSED WORK

This section implements the system and analyzes the results by where initially installing all the required libraries that are used for the development. The code that is used for deploying the required output is shown in Fig. 5.

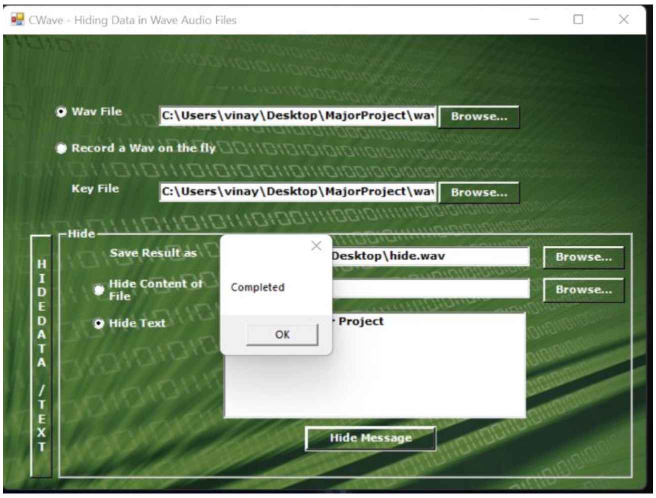

File Edit Selection View Go Run Terminal Help
wave.cs - Visual Studio Code
wave.cs
SteganoWave.csproj
C: > Users > vinay > Desktop > Wave > wave.cs

```
sing System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;
using System.Text;
using System.Threading;
namespace SteganoWave
{
public class frmMain : System.Windows.Forms.Form

{
private System.Windows.Forms.TextBox txtSrcFile;
private System.Windows.Forms.Button btnSrcFile;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Button btnKeyFile;
private System.Windows.Forms.TextBox txtKeyFile;
private System.Windows.Forms.RadioButton rdoSrcFile;
private System.Windows.Forms.RadioButton rdoRecord;
private System.Windows.Forms.ErrorProvider errorProvider;
private PictureBox pictureBox1;
private GroupBox gbHide;
private GroupBox gbExtract;
private RadioButton rdoMsgText;
private Button btnMsgFile1;
private TextBox txtMsgFile;
private RadioButton rdoMsgFile;
private TextBox txtMessage;
```

Fig. 5. Implementation of the code.

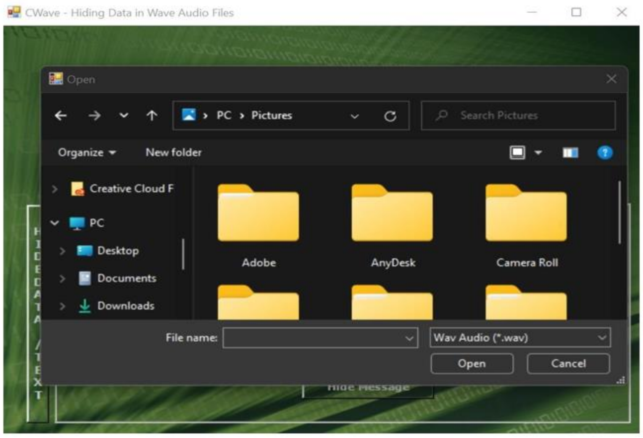
After writing the code now a GUI window will be opening, which asks for data need to browse. the selecting the data file from the browse option and hiding the data message by assigning it with a key-value as shown in Fig. 6.



● Wav File C:\Users\vinay\Desktop\MajorProject\wa
● Record a Wav on the fly
Key File C:\Users\vinay\Desktop\MajorProject\wa
Save Result as Desktop\hide.wav
○ Hide Content of File
● Hide Text Project
Completed

Fig. 6. Creating the Browser using GUI for Hiding the text.

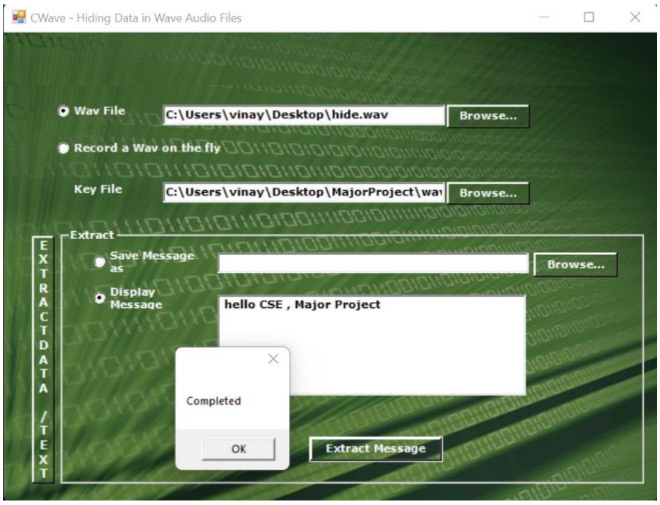
If the data message wanted to be seen then need to browse the file by the respective name as shown in Fig. 7.



File name:
Wav Audio (*.wav)

Fig. 7. Visualizing the Files from the Browser if needed.

Now open the file where the message is hidden and that hidden message needs to be extracted. Then by clicking on the extract message option the data will be displayed as shown in Fig. 8.



CWave - Hiding Data in Wave Audio Files

● Wav File C:\Users\vinay\Desktop\hide.wav

○ Record a Wav on the fly

Key File C:\Users\vinay\Desktop\MajorProject\wa

Extract

○ Save Message as

● Display Message hello CSE , Major Project

Completed

Fig. 8. Extract the Files from the Browser.

Hence concluding that the data messages can be hidden and extracted by using with the key. However, this data cannot be accessed by unauthorized people and the data message will be secured only with the one person who have the accessing key. Through this it can be stated that the data will be highly optimized and there is a less chances of accessing and hacking the data by other people.

VI. CONCLUSIONS

Finally concluding that this project is primarily used to hide data in audio files and deliver data in a secure manner in audio files. This experiment demonstrated how the message may communicate data in a secure manner. In this project, the data is hidden in audio files. As a result, hackers are unable to access the information. Different software techniques are used for building the interface. However, the LSB algorithm extracts the text from the audio file on the receiver side. We decrypt the data from the encrypted format using the AES algorithm.

By using the proposed work further some extensions can be done where the data can be hidden using video files. More security can be achieved through enhancements such as audio file compression.

ACKNOWLEDGMENT

The authors are very much thankful to the management of the GRIET for their relentless support in providing various facilities and a convenient atmosphere for quality education. The authors also thank the Director, Principal, and Head of Computer Science Department, Faculty and Support Staff for their constant support and advice and each stage of student life in this college.

REFERENCES

[1] Dutta, Poulami, Debnath Bhattacharyya, and Tai-hoon Kim. "Data hiding in audio signal: A review." International journal of database theory and application 2, no. 2 (2009): 1-8.

[2] Dong, Xiaoxiao, Mark F. Bocko, and Zeljko Ignjatovic. "Data hiding via phase manipulation of audio signals." In 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. V-377. IEEE, 2004.

[3] Puteaux, Pauline, SimYing Ong, KokSheik Wong, and William Puech. "A survey of reversible data hiding in encrypted images–The first 12 years." Journal of Visual Communication and Image Representation 77 (2021): 103085.

[4] Ghosh, Sumit. Principles of secure network systems design. Springer Science & Business Media, 2002.

[5] Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T. and Jung, K.H., 2018. Image steganography in spatial domain: A survey. Signal Processing: Image Communication, 65, pp.46-66.

[6] Manohar, N., and Peetla Vijay Kumar. "Data encryption & decryption using steganography." In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 697-702. IEEE, 2020.

[7] Grimes, Fergal. Microsoft. Net for Programmers. Manning, 2002.

[8] Barkai, David. "Technologies for Sharing and Collaborating on the Net." In Proceedings First International Conference on Peer-to-Peer Computing, pp. 13-28. IEEE, 2001.

[9] Thai, Thuan, and Hoang Lam. NET framework essentials. " O'Reilly Media, Inc.", 2003.

[10] Chappell, David, and David Wayne Chappell. "Understanding. NET: a tutorial and analysis." (2002).

[11] Hejlsberg, Anders, Mads Torgersen, Scott Wiltamuth, and Peter Golde. The C# programming language. Pearson Education, 2008.

[12] Akunga, Orina E. "Computer code for selecting appropriate point of use water filtration treatment technology." PhD diss., 2013.

[13] Sellars, Duncan. "An introduction to steganography." (2007).

[14] Alam, Fahim Irfan, Fathena Khanam Bappee, and Farid Uddin Ahmed Khondker. "An investigation into encrypted message hiding through images using LSB." International Journal of Engineering Science and Technology (IJEST) 3, no. 2 (2011): 948-960.

[15] Upadhyaya, Amit, Rajesh Kumar Pathak, and Dimple Jayaswal. "A Survey on Different Application of Data Hiding." (2012).

[16] Singh, Prabhishek, and Ramneet Singh Chadha. "A survey of digital watermarking techniques, applications and attacks." International Journal of Engineering and Innovative Technology (IJEIT) 2, no. 9 (2013): 165-175.

[17] Hu, Hwai-Tsu, and Tung-Tsun Lee. "Hybrid blind audio watermarking for proprietary protection, tamper proofing, and self-recovery." IEEE Access 7 (2019): 180395-180408.

[18] Aljazaery, Ibtisam, Haider Alrikabi, and Mustafa Aziz. "Combination of hiding and encryption for data security." (2020): 34-47.