www.czasopisma.pan.pl PAN www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 1, PP. 121–126
Manuscript received November 23, 2022; revised February, 2023. DOI: 10.24425/ijet.2023.144340

Steganography in Audio Files – COTS Software Analysis

Piotr Marszałek, and Piotr Bilski

***Abstract*—The paper presents the analysis of the Commercial Off-The-Shelf (COTS) software regarding the ability to be used in audio steganography techniques. Such methods are a relatively new tool for hiding and transmitting crucial information, also being used by hackers. In the following work, the publicly available software dedicated to audio steganography is examined. The aim was to provide the general operating model of the information processing in the steganographic effort. The embedding method was analyzed for each application, providing interesting insights and allowing classifying the methods. The results prove that it is possible to detect the hidden message within the specific audio file and identify the technique that was used to create it. This may be exploited further during the hacking attack detection and prevention.**

***Keywords*—steganography; audio; information hiding; steganalysis; encryption; compression; cybersecurity; cyberspace; embedding; information security; privacy; signal plane; non-signal plane**

I. Introduction

STEGANOGRAPHY is the art and science of hiding secrets in obvious communication channels without being noticed by third parties. It has been used for centuries, from ancient times utilizing, for instance, human heads or wooden tablets [1], to the era of the Internet - using a digital form of information. Currently, network streams, digital documents, images, audio, and video files are possible carriers of hidden information.

Since the year 2011, when the existence of Duqu malware was discovered, steganography has become another threat in cyberspace [2]. It has potential for unauthorized and clandestine exfiltration of information from organizations or it might be a carrier of malicious code of various forms, e.g., viruses, worms, and Trojan horses [3].

From this point of view, there is a continuous need to develop tools and methods to counteract this type of threat. Due to the complexity involved in developing signal processing algorithms for steganography, it is belived that the majority of cyberattacks involving audio files would be carried out using popular software rather than custom and sophisticated algorithms.

This paper presents results of experiments on contemporary audio steganography tools (see Table I) and embedding methods with the focus on the information processing chain (from the secret message to the output audio file) and embedding algorithms. Table II summarizes findings relevant to the tools examined and is an attempt to fill in the gaps outlined in the literature review below.

TABLE I
SOFTWARE CONSIDERED FOR ANALYSIS

Application	Version	Release year	Ref.
S-Tools	4.0	1996	[4] [5]
Hide4PGP	2.0	2000	[6]
Camouflage	1.2.1	2001	[7]
Xiao Steganography	2.6.1	2005	[8]
StegoStick	1.0	2008	[9]
mp3stegz	1.0.0	2008	[10]
SilentEye	0.4.1	2011	[11]
DeEgger Embedder	1.21	2012	[12]
Clotho	2.4	2012	[13]
QuickCrypto	4.1	2013	[14]
Invisible Secrets	4.8.0	2013	[15]
DeepSound	2.0	2015	[16]
OpenPuff	4.01	2018	[17]
MP3Stego	1.1.19	2018	[18]
Steganos Privacy Suite	22.3.0	2020	[19] [20]

II. Literature Review

Research on digital steganography has been so far focused on images rather than sound files, which can be easily demonstrated by querying popular scientific databases (e.g. Scopus, Web Of Science, IEEExplore, Springer) with two alternative terms - “audio steganography” versus “image steganography” – the latter being clearly dominant. The following section discusses references in the field of audio digital steganography and serves as a background to the conducted research.

A significant number of papers regarding embedding algorithms for audio steganography is published [21], [22]. In most cases sophisticated approaches in various domains (e.g. temporal, transform, cepstral) are considered. This is where the question arises as to whether they are used in practice. The vast majority of previous work concerns embedding algorithms, with less attention paid to the tools themselves - only a few percent of publications (ca. 3% as assessed from [22]) is devoted to software tools.

P. Marszalek is with Doctoral School, Warsaw University of Technology, Warsaw Poland (e-mail: marszalek@idsp.eu).

P. Bilski is with Warsaw University of Technology, Warsaw, Poland (e-mail: piotr.bilski@pw.edu.pl).

TABLE II
SOFTWARE FOR AUDIO STEGANOGRAPHY – FEATURES

Application	Secret message				Cover file	Trans-coding	Stego file	Embedding algorithms		
	Form	Compression	Encryption	Integrity verification	Supported formats		Supported formats	Stego-key	Signal plane	Non-signal plane
S-Tools	anyfile	o	+	-	WAV	+	WAV	+	LSB1	-
Hide4PGP	anyfile, text	-	-	-	VOC	-	VOC	-	adaptive: LSB1, LSB2, LSB3, LSB4.	-
					WAV	-	WAV	-	adaptive: LSB1, LSB2, LSB3, LSB4.	-
Camouflage	anyfile, multihidden	-	+	-	WAV	-	WAV	-	-	suffix
					MP3	-	MP3	-	-	suffix
Xiao Steganography	anyfile, multihidden	-	o	-	WAV	-	WAV	-	Xlsb1	-
StegoStick	anyfile, user text	-	+	?	WAV	-	WAV	-	Xlsb1	-
mp3stegz	anyfile	+	+	+	MP3	-	MP3	-	-	MP3efs
SilentEye	anyfile, user text	o	o	-	WAV	-	WAV	-	optional: LSB1, LSB2, LSB3, LSB4, LSB5, LSB6.	-
DeEgger Embedder	anyfile, multihidden	-	-	-	AU	-	AU	-	-	suffix
					MIDI	-	MIDI	-	-	suffix
					MP3	-	MP3	-	-	suffix
					WMA	-	WMA	-	-	suffix
Clotho	anyfile multihidden	o	o	?	MIDI	-	MIDI	-	-	suffix
					OGG	-	OGG	-	-	suffix
					WAV	-	WAV	-	-	suffix
QuickCrypto	anyfile, user text, anyfile+user text,	-	-/o	-	WAV	-	WAV	-	Xlsb1	-
					MP3	-	MP3	-	-	suffixLSB1
Invisible Secrets	anyfile, multihidden	o	o	-	WAV	-	WAV	-	LSB1	-
DeepSound	anyfile, multihidden	-	o	-	APE	+	WAV	-	optional: LSB1, LSB2, LSB3, LSB4, LSB6, LSB8, LSB12, LSB16. Xbs	-
					FLAC					
					MP3					
					WMA					
OpenPuff	anyfile	-	+	+	AIFF	-	AIFF	+	LSB1op	-
					MP3	-	MP3	+	MP3op	-
					NeXT/Sun	-	NeXT/Sun	+	LSB1op	-
					WAV	-	WAV	+	LSB1op	-
MP3Stego	anyfile	+	+	?	WAV	+	MP3	-	MP3stego	-
Steganos Privacy Suite	multihidden	?	+	+	MP3	-	MP3	-	-	suffix
					M4A	-	M4A	-	-	suffix

Legend: (+) – mandatory use, (-) – not used, (o) – optional use (user's choice), (?) – not known.

There are publications available on steganography tools [23] [24] [25] [26], but they are not dedicated solely to audio files. In general, less than half of the enlisted tools can operate on sound files. These works mainly suffer from the lack of a comprehensive and systematic presentation of audio embedding algorithms being implemented.

To the best of our knowledge, no work is available that deals in more detail with processing steps of a secret message (and even other data) that are performed before audio embedding. That knowledge could be supportive from a steganalysis point of view. In some cases [27], auxiliary data is added in an application-specific manner, representing fingerprints that are readily used by steganography detectors.

III. Experimental Setup and Methodology

During the investigation, the personal computer (PC) with virtualization technology (VMWare Workstation Pro 16) was used. On a dedicated Virtual Machine the audio steganography software shown in Table I was installed. The so-called Analysis Environment (AE) was set up consisting of specialized software and collection of custom crafted files. A shared folder is the interface allowing for data exchange between Virtual Machine and Analysis Environment, see Fig. 1.

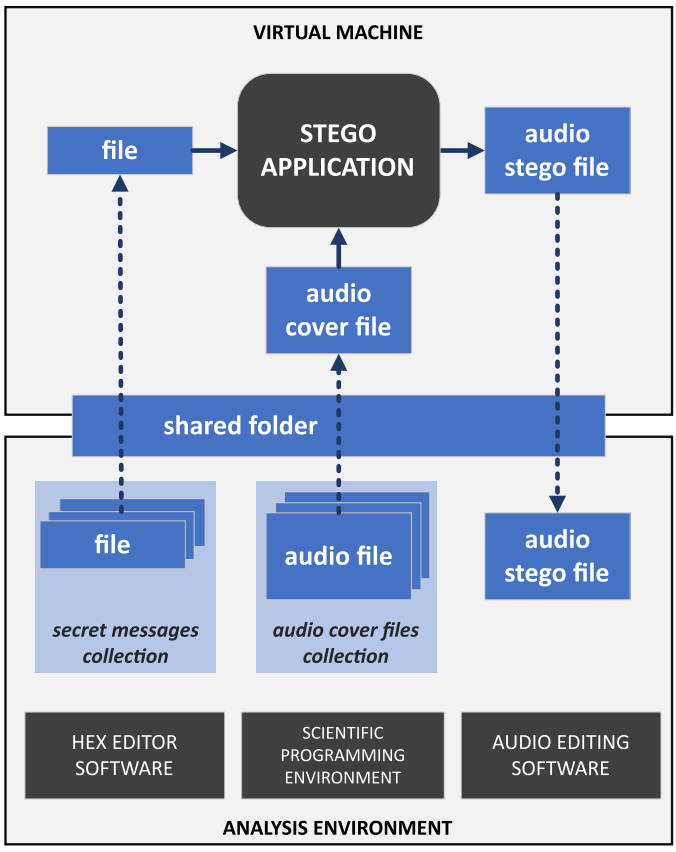

Fig. 1. PC-based experimental setup

The purpose of the AE is twofold. First, it allows for the generation of fabricated files that would serve as secret messages and audio covers for further investigation. Second, it provides means to analyze the behavior of considered applications in a very flexible and creative way. These requirements are made possible by three main players, i.e.: audio editing software (Audacity 3.0.4), hex editor software (Hex Editor Neo 6.54) and scientific programming environment (Spyder 4.1.5 and Matlab R2020b).

The methodology of the investigation is based on the idea of a black box, so a stego application is treated as a homogeneous entity with only control over its inputs (i.e. secret message and audio cover file) and access to the output - an audio file carrying hidden information (audio stego file). Various types of differential analysis can provide information about the operating principles of the box. Consequently, to gain knowledge of the software under examination we don't use "invasive" reverse engineering methods, such as disassembling, decompiling, code modification, etc.

For example, to find out where the hidden information is placed in an audio file a small file is used (60 bytes, all with the content of 0xFF) as a secret message and the two-second wave file of silence (all samples are zero). A simple examination of the resulting stego file in the audio editing software (by zooming in on samples) provides the answer through detecting non-zero values.

IV. The Model of Operation

Based on the analyzed software, it was possbile to propose the general model of operation of a contemporary audio steganographic tool - see Fig. 2. It is worth noting that not necessarily all data processing techniques proposed here exist altogether for a given stego application. Some techniques might be turned on/off by users' conscious decision, while others are hardcoded in an application and referred to as "optional" and "mandatory" respectively. In the following section the data processing steps covered by the Model are described.

A secret message may be a single computer file, multiple files (called "multihidden" in this case), or a text taken from the user's prompt. All files have metadata (e.g. file name, extension, size expressed in bytes) that usually are processed additionally in the next step. Sometimes the user is asked for a passphrase later used to produce keys, cipher and/or stego, in a key generation process whether it is implemented. Also, the user selects the audio cover file that would finally be a carrier for a secret message.

In the particular application, at the beginning of the hiding process, a secret message could be compressed, ciphered, or verified for integrity to enhance the information security level.

At the heart of the steganographic effort is an embedding algorithm. This stage takes the preprocessed secret message data and put it into the audio cover file in a pseudo-random manner (after shuffling utilizing a stego-key) or at fixed positions - usually at the beginning of an audio signal or as a suffix added at the end of a carrier file. In most cases, auxiliary data such as file checksums, encrypted (or not) metadata, or application synchronization data are added. This additional data, embedded along with a secret message, are utilized

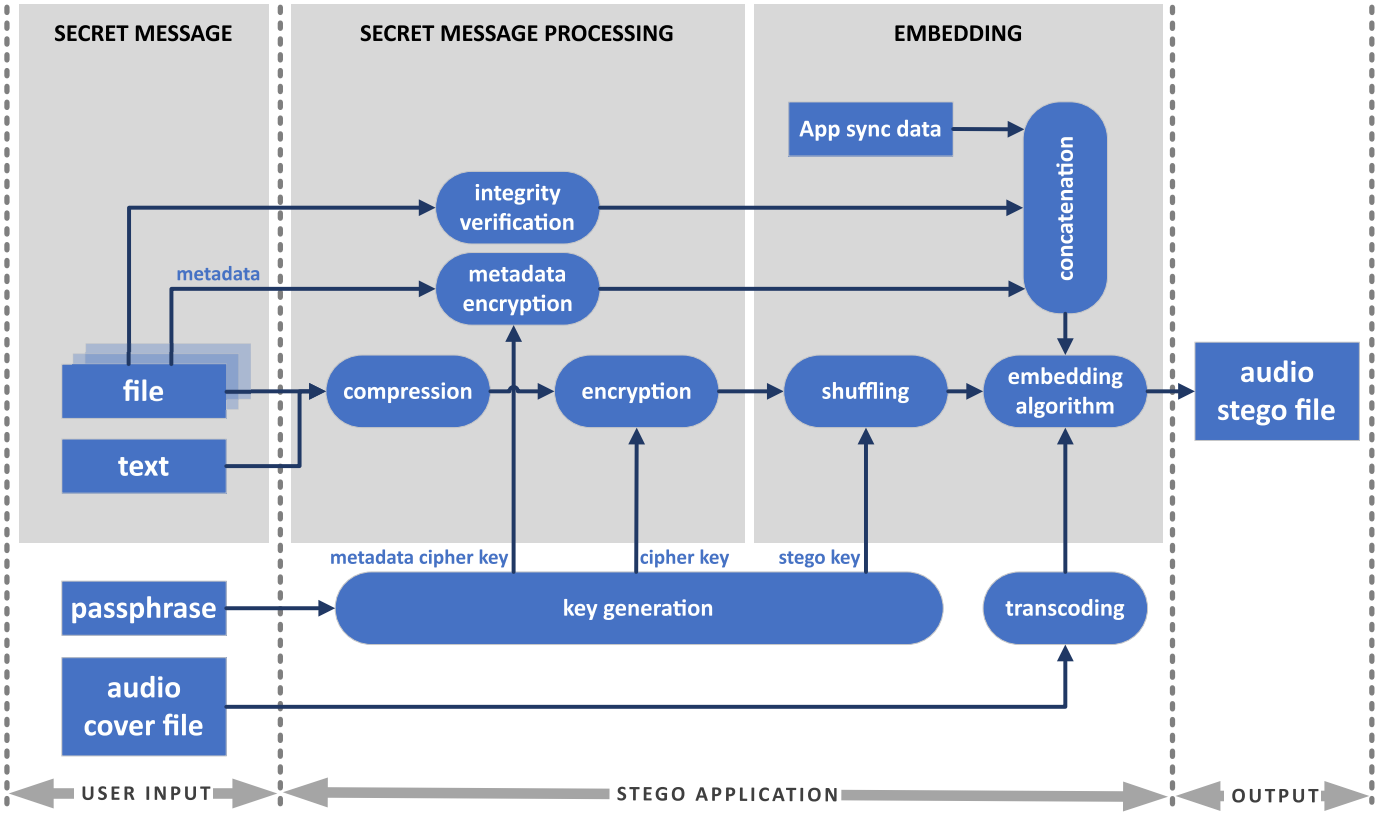

Fig. 2. The General Model of operation of a contemporary audio steganographic software

on the recipient's side during the information extracting process. For some applications, it was observed that cover audio, before embedding, is transcoded from one audio format into another – for instance from MP3 to WAV or the opposite.

The resultant audio stego file consists of the hidden secret message and is meant for sending through ordinary communication channels without suspicion. Usually, to retrieve a secret message, the recipient needs to have at least the same stego application and also must know the proper passphrase.

V. Audio File Embedding Algorithms

The purpose of this section is to demonstrate results, and provide a brief description of the embedding techniques used by the examined steganography tools. Audio file steganography embedding strategies could be generally classified into two groups, depending on whether they operate in the signal domain or not, i.e. signal plane and non-signal plane algorithms. Embedding in the former always introduces noise to an audio cover signal and therefore affects it's quality imperceptibly in principle. Consequently, developing algorithms of this kind is a demanding task due to the need of possessing some level of expertise in the field of Digital Signal Processing (DSP). The second approach is to add secret data in the non-signal plane, i.e. by hiding information outside a signal part (as specified in an audio format) of an audio file. This scheme is relative easy to implement in software, but the hidden message is trivial to detect by simple means.

A. Signal plane algorithms

The bit substitution is the most utilized approach for audio steganography. This easy-to-implement technique, commonly used for lossless audio formats, involves replacing the original bits of a signal sample with bits of a processed secret message, see the Model. To satisfy the demand for steganographic undetectability, such modification should remain unnoticed, therefore the most common approach is to manipulate the least significant bit in a numeric representation of audio samples. It is possible to modify more than one bit of a sample. For the sake of generalization, this approach is referred to as *LSBk*, where *k* stands for the number of original least significant bits altered in a steganographic process. It was observed that *k* number can go up quite high (e.g. 16 as in **DeepSound**). Moreover, the steganographic apps differ in terms of how they manage bit substitution. In some cases, the user is asked to choose the desired level of quality for the resulting steganographic audio file and which consequently leads to utilizing one of the *LSBk* schemes concerning a signal length or signal bitness (i.e. number of bits to represent an audio sample) – this approach may be called "adaptive". In other cases, the user explicitly chooses some option, so their implementation is "optional". The last possibility is when the user has no control

over the bit substitution process and the application dictates the use of a particular scheme, most likely *LSB1*.

The most striking result to emerge from the research on the bit substitution in lossless audio formats is that in some applications this algorithm is not implemented correctly from a DSP perspective and for the sake of steganographic undetectability demand. Namely, steganographic application alters in this case the least significant bit in all bytes of the audio data stream instead of changing only the least significant bit in an audio sample.This malfunction, observed for audio formats with sample bitness more than 8 bit per sample, implies that a resulting stego audio file is biased by significant noise that reveals the presence of a hidden message more tangibly. This defective implementation is referred to as *Xlsb1* on the contrary to *LSB1* mentioned earlier. Furthermore, any other kind of erroneous implementation of the bit substitution algorithm is identified as *Xbs*.

It was observed other kinds of algorithms in the signal plane that do not suit the schemes discussed above. More creative programmers develop custom audio steganography algorithms. For instance, in the **OpenPuff** two algorithms are implemented: *LSB1op* and *MP3op* for lossless and compressed audio formats, respectively. The *LSB1op* algorithm alters the only least significant bit of arbitrary and uniformly selected samples over entire signal duration. The *MP3op* algorithm works in the signal plane and puts secret messages at the beginning and the end of the compressed cover signal. In the **MP3Stego** application secret message embedding is done during compression inner loop – this algorithm can be called *MP3stego*.

B. Non-Signal plane algorithms

The second approach to hide data in audio files is to use the non-signal part of a cover audio file. This metohd requires no DSP knowledge, making it easy to implement even for novice programmers. Possible hiding spots could be audio metadata, unused data fields provided by a data format or other fields to be overwritten, and at the very end of a file.

When secret data is added to the original content of an audio file, is most commonly used by the audio steganography tools that operate in the non-signal plane. We call such a scheme the suffix. The quality of digital content is not degraded after such modification of underlying data.

An interesting modification of the suffix embedding tactic was found in **QuickCrypto** . The term *suffixLSB1* is used to refer to a situation in which steganographic data is carried only by the least significant bit of byte stream added as a suffix. Remaining bits have no particular purpose and come from a random generator.

The last type of embedding in the non-signal layer of compressed audio files was found in the application called **mp3stegz** – this algorithm can be named as *MP3efs* (*efs* stands for “empty frame stuffing” [28]). The principle of steganographic operation, in this case, is to find unused frames in a constant bitrate (CBR) MP3 cover file and stuff them with secret data, all that without introducing any noise.

VI. Conclusions

The main goal of this paper was to provide an insight into how publicly available audio steganography software works and thereby potentially support the steganalysis processes. The original outcome of this study was a comprehensive examination of more than a dozen apps dedicated to steganography with a sole focus on audio carrier files.

The conducted experiments have shown that there are two general ways to conceal information in audio files, i.e. directly in audio signal and without affecting it. For the sake of such a distinction, the terms “signal plane” and “non-signal plane” embedding algorithms have been introduced, respectively.

The second finding was that the most common approach to embedding method for lossless audio formats is to utilize least significant bit modification. Implementation of such an approach was flawed in some applications, resulting in unintended and poor steganographic features.

Based on the conclusions drawn from the analysis a general operational model of an audio steganography application was proposed. It contains possible elements of the processing chain - from a user secret message data and selected carrier file to the resulting stego file.

A natural progression of this work is to provide detailed information about principles of operation of more complex algorithms that are not usually disclosed and can be uncovered by steganalysis, software reverse engineering methods, or analysis of the source code when available.

These findings contribute in several ways to our understanding of audio steganography that can be met in the cyberspace and provide a basis for further investigation in this interesting, still not well explored, subject.

References

[1] J. C. Ingemar, M. L. Miller, A. B. Jeffrey, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. Elsevier Inc., 2008.
[2] McAfee Labs, “McAfee Labs Threats Report: June 2017,” McAfee Labs, Tech. Rep. June, 2017. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-jun-2017.pdf
[3] Joint Task Force, “National Institute of Standards and Technology Special Publication 800-53, Revision 5 : Security and Privacy Controls for Information Systems and Organizations,” *NIST Special Publication*, p. 465, 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-53r5
[4] A. Brown, “S-Tools.” [Online]. Available: http://www.modemac.com/s-tools.html
[5] “S-Tools Version 4.0.” [Online]. Available: https://www.cs.vu.nl/~ast/books/mos2/steg.zip
[6] H. Repp, “Hide4PGP.” [Online]. Available: http://www.heinz-repp.onlinehome.de/Hide4PGP.htm
[7] Twisted Pear Productions, “Camouflage.” [Online]. Available: http://camouflage.unfiction.com/
[8] Int21 and nakasoft.net, “Xiao Steganography.” [Online]. Available: https://xiao-steganography.en.softonic.com/
[9] P. Uma Mahesh and V. Santhosh Kumar, “StegoStick.” [Online]. Available: https://sourceforge.net/projects/stegostick/
[10] Z. Achmad, “mp3stegz.” [Online]. Available: https://sourceforge.net/projects/mp3stegz/
[11] A. Chorein, “SilentEye.” [Online]. Available: https://achorein.github.io/silenteye/
[12] Z.A.Software, “DeEgger Embedder.” [Online]. Available: https://www.softpedia.com/get/Security/Encrypting/DeEgger-Embedder.shtml
[13] H. Nugraha, “Clotho.” [Online]. Available: https://www.softpedia.com/get/Security/Encrypting/Nugraha-Clotho.shtml
[14] Cybernescence Limited, “QuickCrypto.” [Online]. Available: http://quickcrypto.com/

[15] East-Tec SRL, "InvisibleSecrets." [Online]. Available: https://www.east-tec.com/invisiblesecrets/
[16] J. Bátor, "DeepSound." [Online]. Available: http://jpinsoft.net/deepsound
[17] EmbeddedSW, "OpenPuff." [Online]. Available: https://www.embeddedsw.net/OpenPuff_Steganography_Home.html
[18] F. A. Petitcolas, "MP3Stego." [Online]. Available: https://www.petitcolas.net/steganography/mp3stego/
[19] Steganos Software GmbH, "Press Release Generation 22," 2020. [Online]. Available: https://www.steganos.com/images/presse/pdfs/2020-09-10_Press_Release_Generation_22_EN.pdf
[20] ——, "Steganos Privacy Suite." [Online]. Available: https://www.steganos.com/en/products/steganos-privacy-suite
[21] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques," *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 2012, no. 1, pp. 1–16, 2012.
[22] A. A. Alsabhany, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar, "Digital audio steganography: Systematic review, classification, and analysis of the current state of the art," 11 2020.
[23] M. B. Pope, M. Warkentin, E. Bekkering, and M. B. Schmidt, "Digital Steganography—An Introduction to Techniques and Tools," *Communications of the Association for Information Systems*, vol. 30, 2012.
[24] R. Das and I. Karadogan, "An Investigation on Information Hiding Tools for Steganography," *International Journal of Information Security Science*, vol. 3, no. 3, pp. 200–208, 2014.
[25] G. Konakhovich, Y. Symonychenko, A. Symonychenko, and Y. I. Daradkeh, "The Research of Realization of Hidden Channel for Information Transmission with the Use of Steganographic Tools," in *Proceedings of the International Workshop on Cyber Hygiene (CybHyg-2019)*, 2019, pp. 504–514.
[26] U. Pilania, R. Tanwar, P. Gupta, and T. Choudhury, "A roadmap of steganography tools: conventional to modern," *Spatial Information Research*, 2021. [Online]. Available: https://doi.org/10.1007/s41324-021-00393-7
[27] C. Gong, J. Zhang, Y. Yang, X. Yi, X. Zhao, and Y. Ma, "Detecting fingerprints of audio steganography software," *Forensic Science International: Reports*, vol. 2, no. December 2019, p. 100075, 2020.
[28] M. Zaturenskiy, "MP3 files as a steganography medium," *RIIT 2013 - Proceedings of the 2nd Annual Conference on Research in Information Technology*, vol. 1, no. 630, pp. 23–28, 2013.