Software Tools and Technologies in Steganography

Julijana Mirčevski[1], Biljana Djokić[2], Milesa Srećković[3] and Nikola Popović[4]

Abstract: **In the paper was exposed the performance estimation of more available program tools for steganographic application and message implementation method. The results were classified and illustrated by the tested procedures. The message creation, implementation and detection was shown in oneself programs tool that operate in Visual Basic environment.**

Keywords: **steganography, invisible communication, message embedding, steganographic file detection, wavelet transform**

I. INTRODUCTION

Steganography as a way of invisible communication is very present in current internet communication. A number of steganographic application available with a various performance in domain of message implementation, system requirements, encoding security and executing reliability.

The digital contents database requires a more complexes software tools in order to searching, analysis's, compression and reproduction. Wavelet theory provides the fast discrete algorithms suitable to computer programming application. The clear mathematical theory is a good base to creating a programs environment and a number programs packages. The wavelet transform application are thinkingless without a programming software package but the most used are WaveLab, LastWave, MegaWave i Rice Wavelet Toolbox. Because the mentioned software products belong to free ware software category, it is possible to use that with a registration legality.

Steganography hides the covert message but not the fact that two parties are communicating with each other [1]. The steganography process generally involves placing a hidden message in some transport medium, called the carrier. The secret message is embedded in the carrier to form the steganography medium. The use of a steganography key may be employed for encryption of the hidden message and/or for randomization in the steganography scheme. In summary:

steganography_medium = hidden_message + carrier + steganography_key

On the followed figure it is presented scheme of steganography systematization according to [2]. Only the main items would be explained in text.

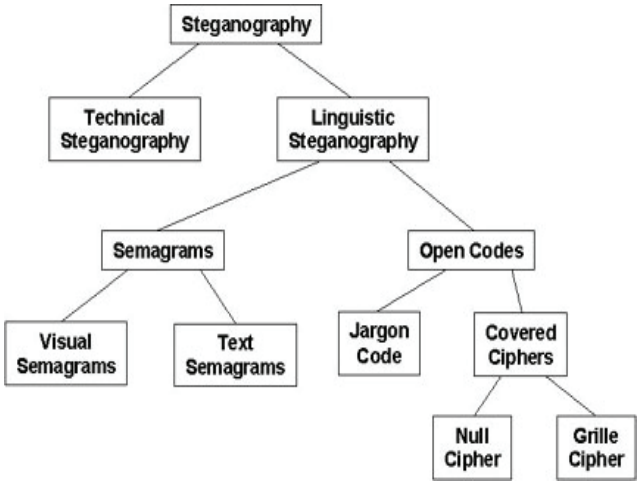

Fig. 1. Classification of Steganography Techniques

-Technical steganography uses scientific methods to hide a message, such as the use of invisible ink or microdots and other size-reduction methods.

- Linguistic steganography hides the message in the carrier in some nonobvious ways and is further categorized as semagrams or open codes.

- Semagrams hide information by the use of symbols or signs.

A visual semagram uses innocent-looking or everyday physical objects to convey a message, such as doodles or the positioning of items on a desk or Website. A text semagram hides a message by modifying the appearance of the carrier text, such as subtle changes in font size or type, adding extra spaces, or different flourishes in letters or handwritten text.

- Open codes hide a message in a legitimate carrier message in ways that are not obvious to an unsuspecting observer. The carrier message is sometimes called the overt communication, whereas the hidden message is the covert communication. This category is subdivided into jargon codes and covered ciphers.

- Jargon code, as the name suggests, uses language that is understood by a group of people but is meaningless to others. Jargon codes include warchalking (symbols used to indicate the presence and type of wireless network signal [3]), underground terminology, or an innocent conversation that conveys special meaning because of facts known only to the speakers. A subset of jargon codes is cue codes, where certain prearranged phrases convey meaning.

- Covered or concealment ciphers hide a message openly in the carrier medium so that it can be recovered by anyone who knows the secret for how it was concealed. A grille cipher employs a template that is used to cover the carrier message.

[1]Julijana Mirčevski, Informatička škola "Educon", Beograd, julijana@afrodita.rcub.bg.ac.yu
[2]Biljana Djokić, Informatička škola "Educon", Beograd, djokicb@eunet.yu
[3]Milesa Srećković, Elektrotehnički fakultet, Beograd, milesa@afrodita.rcub.bg.ac.yu
[4]Nikola Popović, Ministarstvo inostranih poslova, Beograd

II. FEATURES AND APPLICATIONS

Data-hiding techniques should be capable of embedding [3] data in a host signal with the following restrictions and features:

1. The host signal should be nonobjectionally degraded and the embedded data should be minimally perceptible. (The goal is for the data to remain *hidden*. As any magician will tell you, it is possible for something to be hidden while it remains in plain sight; you merely keep the person from looking at it. We will use the words *hidden*, *inaudible*, *imperceivable*, and *invisible* to mean that an observer does not notice the presence of the data, even if they are perceptible.)
2. The embedded data should be directly encoded into the media, rather than into a header or wrapper, so that the data remain intact across varying data file formats.
3. The embedded data should be immune to modifications ranging from intentional and intelligent attempts at removal to anticipated manipulations, such as a channel noise, filtering, resampling, cropping, encoding, lossy compressing, printing and scanning, digital-to-analog (D/A) conversion, and analog-to-digital (A/D) conversion and other.
4. Asymmetrical coding of the embedded data is desirable, since the purpose of data hiding is to keep the data in the host signal, but not necessarily to make the data difficult to access.
5. Error correction coding1 should be used to ensure data integrity. It is inevitable that there will be some degradation to the embedded data when the host signal is modified.
6. The embedded data should be self-clocking or arbitrarily reentrant [3], [4]. This ensures that the embedded data can be recovered when only fragments of the host signal are available, e.g., if a sound bite is extracted from an interview, data embedded in the audio segment can be recovered. This feature also facilitates automatic decoding of the hidden data, since there is no need to refer to the original host signal.

Trade-offs exists between the quantity of embedded data and the degree of immunity to host signal modification. By constraining the degree of host signal degradation, a data-hiding method can operate with either high embedded data rate, or high resistance to modification, but not both. As one increases, the other must decrease. While this can be shown mathematically for some data-hiding systems such as a spread spectrum, it seems to hold true for all data-hiding systems. In any system, you can trade bandwidth for robustness by exploiting redundancy. The quantity of embedded data and the degree of host signal modification vary from application to application. Consequently, different techniques are employed for different applications [5]. Several prospective applications of data hiding are discussed in this section. An application that requires a minimal amount of embedded data is the placement of a digital water mark.

The embedded data are used to place an indication of ownership in the host signal, serving the same purpose as an author's signature or a company logo. Since the information is of a critical nature and the signal may face intelligent and intentional attempts to destroy or remove it, the coding techniques used must be immune to a wide variety of possible modifications.

A second application for data hiding is tamper-proofing. It is used to indicate that the host signal has been modified from its authored state. Modification to the embedded data indicates that the host signal has been changed in some way.

A third application, feature location, requires more data to be embedded. In this application, the embedded data are hidden in specific locations within an image. It enables one to identify individual content features, e.g., the name of the person on the left versus the right side of an image. Typically, feature location data are not subject to intentional removal. However, it is expected that the host signal might be subjected to a certain degree of modification, e.g., images are routinely modified by scaling, cropping, and tonescale enhancement. As a result, feature location datahiding techniques must be immune to geometrical and other nongeometrical signal modification.

III. DIGITAL CARRIER METHODS

There are many ways to be hidden messages in digital media. The most common steganography method in audio and image files employs some type of least significant bit substitution or overwriting. The least significant bit term comes from the numeric significance of the bits in a byte. The high-order or most significant bit is the one with the highest arithmetic value (i.e., 2^7=128), whereas the low-order or least significant bit is the one with the lowest arithmetic value(i.e.,2^0=1).

As a simple example of least significant bit substitution, imagine "hiding" the character 'G' across the following eight bytes of a carrier file (the least significant bits are underlined):

10010101 00001101 11001001 10010110

00001111 11001011 10011111 00010000

A 'G' is represented in the American Standard Code for Information Interchange (ASCII) as the binary string **01000111**. These eight bits can be "written" to the least significant bit of each of the eight carrier bytes as follows:

1001010*0* 00001101 1100100*0* 10010110

0000111*0* 11001011 10011111 0001000*1*

In the sample above, only half of the least significant bits were actually changed (shown above in italics and colored). This makes some sense when one set of zeros and ones are being substituted with another set of zeros and ones.

Least significant bit substitution can be used to overwrite legitimate RGB color encodings or palette pointers in GIF and BMP files, coefficients in JPEG files, and pulse code modulation levels in audio files. By overwriting the least significant bit, the numeric value of the byte changes very little and is least likely to be detected by the human eye or ear.

Least significant bit substitution is a simple, albeit common, technique for steganography. Its use, however, is not necessarily as simplistic as the method sounds. Only the most naive steganography software would merely overwrite every least significant bit with hidden data. Almost all use

some sort of means to randomize the actual bits in the carrier file that are modified. This is one of the factors that make steganography detections so difficult.

All steganographic methods are trying to achieve the minimal amount of modification in order to minimize the view changes through introducing detectable artifacts. However, if the cover-image, was initially stored in the JPEG format (as it is frequently the case), message embedding in the spatial domain will disturb but not erase the characteristic structure created by the JPEG compression. It is possible to recover the JPEG format table from stego-image by carefully analyzing the values of DCT coefficients in all blocks. After message embedding, the cover image will become, with a high probability, grather then pure image and, yet not fully compatible with JPEG format. This can be indicated the steganography file.

IV. STEGANOGRAPGIC SOFTWARE TOOLS

There are a number of steganographic software tools and steganographic techniques today available on the internet [ ]. They can be ranging from freeware software to commercial products of high price. Most of them are creations of amateur enthusiasts available for free while others are products of private companies and can be purchased for a small fee.

The very important function of steganography detection software is to find possible carrier files. Ideally, the detection software would also provide some clues as to the steganography algorithm used to hide information in the suspect file so that the analyst might be able to attempt recovery of the hidden information.

The detection of steganography software continues to become harder for another reason—the small size of the software coupled with the increasing storage capacity of removable media. S-Tools, for example, requires less than 600 KB of disk space and can be executed directly, without additional installation, from a floppy or USB memory key. Under those circumstances, no remnants of the program would be found on the hard drive.

In text followed bellow, would be present some small and powerful software tools.

ChinCrypt is a small and easy to use textmode programme for crypting data. It can be encrypt a text file, executable file or, image file etc. ChinCrypt is the size of 10kB and work under Windows, Linux and Unix.

Gifshuffle is a command-line-only program for Windows which conceals messages in GIF images by shuffling the colourmap. The picture remains visibly intact, only the order of color within the palette is changed. It works with all GIF images, including those with transparency and animation, and in addition provides compression and encryption of the concealed message. Gifshuffle v2.0 0 is freeware and requires only 33kB.

JPegX is an encryption program that hides a important information inside standard JPEG image files. The image is left visually unchanged and messages are encrypted and password protected. To decrypt the message, it's need to open the JPEG file that holds it and enter the password if prompted. **JPegX** is a size of 18kB and freeware category.

Shadow is the powerful data encoder/decoder that has the ability to encode/decode everything and anything that fits on/in computer hard disk drive. It can be encoded texts, pictures, movies, music, applications, and so on. Windows is required to Shadow running and the program size is about 56kB.

Hide4PGP v2.0 is a command-line steganographic program for Windows, DOS, OS/2, and Linux that hides data within BMP, WAV, and VOC files. It is designed to be used with both PGP and Stealth, but also works well as a stand-alone program. Version 2.0 has several new features, including a new stego format which is much more robust against format conversions - only lossy compression formats will loose the hidden data. The source is also included and should compile on any platform without major problems.Hide4PGPv2.0 is size of 114kB and freeware.

V. THE HIDDEN MESSAGE EMBEDDING APPROACH IN AN IMAGE

The practical steganography implementation was executed with program which realized in Visual Basic 6 (VB6) environment. Program is very simple application without intention to demonstrate a high programming efficiency. This program is not comparable with the known steganographic tools, in mentioned sense. The significant characteristics of this program are the simple implementation, wide possibility of modification, and very simple using [6]. The program is realized in order to rich the practical approach to understanding of the fundamental processes to hidden message writing in an image.

The following image contents the hidden message: ***Breza do hramot, osvetlena od sveki livčinja raga.***

Fig. 2. The JPEG image file containing hidden message

Every pixel of the above image has assigned the adequate color value. The text was embedded in the image through changing of the previous color values. The VB6 functions ***SetPixel*** and ***GetPixel*** was used for that. The approach begins with opening new forms in VB6 environment [6], [7]. It is launched a text box and assigned name and then a list box, also. Thesse called the ***Textprvi*** and ***Listaprva***. Further, it introduse a picture box and named as ***Prvaslika***. The scalemode of this box must be set on the ***pixel*** and parameter ***autodraw*** on the ***true*** value. It put in program ***Common Dialog*** control and ***Active X*** control function. The image reading way was solved by oneself procedure. There are defined commands to image decoding, encoding, writing and reading.

The image was added in Design regime, properties function. During the testing procedures was used the both: a JPEG picture format and a bitmaps files. Especially is written the

program part which makes the data file from picture file. The goal of this subprogram is to convert image file in hexadecimal code date file. These hexadecimal code date file is made for pure image and steganographic image. In comparing these files is clear visible the differences between files because the hexadecimal code of steganography image contain secret information [7]. Although even a trained eye wouldn't know the difference, visually, it will change the statistical properties of the pixel values of the "before" and "after" photo. The data hexadecimal files are different, also.

	Steganography image	Pure image
0	0 FEFEFE FFFFFF 65793	FEFEFE FFFFFF
	F8F8EC F8F8F8 12	65793
1	F4F4E2 F4F4F4 18	F9F9F9 F8F8F8 -65793
2	A2A299 A2A2A2 9	FFFFFF F4F4F4 -723723
3	44432B 444345 26	FFFFFF A2A2A2 -118749
4		CECDCF 444345 -9079434
	2B2A1F 2B2A2C 13	919092 2B2A2C -6710886
5	22210A 222125 27	605F63 222125 -4079166
6	201F19 201F23 10	3E3D41 201F23 -1973790
7	232217 232226 15	535256 232226 -3158064
8	393821 39383C 27	ACABAF 39383C -
9	ABAAA6 ABAAAC 6	7566195
	F9F8E8 F9F8FA 18	FCFBFD ABAAAC -
10	FCFCEF FCFCFC 13	5329233
	FCFCFB FCFCFC 1	FFFEFF F9F8FA -394757
11	F8F9E8 F8F9F7 15	F7F7F7 FCFCFC 328965
12	FDFDE9 FDFDFD 20	FCFCFC FCFCFC 0
13	F8F7F9 FDFDFD 329220	FFFFFE F8F9F7 -460295
14	FAF9E0 FAF9FB 27	
	FDFCEF FDFCFE 15	F7F8F6 FDFDFD 394503
15	FEFDEC FEFDFF 19	F8F7F9 FDFDFD 329220
16	F9F8E4 F9F8FA 22	F6F5F7 FAF9FB 263172
17	FCFBF4 FCFBFD 9	FAF9FB FDFCFE 197379
18	FFFEEB FFFEFF 20	FCFBFD FEFDFF 131586
19	FBFAFA FBFAFC 2	F8F7F9 F9F8FA 65793
20	FCFBF4 FCFBFD 9	FBFAFC FCFBFD 65793
21	FCFBEF FCFBFD 14	FCFBFD FFFEFF 197378
22	FCFBF0 FCFBFD 13	F3F2F4 FBFAFC 526344
23	FCFBE2 FCFBFD 27	FAF9FB FCFBFD 131586
24	FCFBEE FCFBFD 15	FAF9FB FCFBFD 131586
25	FCFBF3 FCFBFD 10	FAF9FB FCFBFD 131586
26	FCFBE2 FCFBFD 27	FAF9FB FCFBFD 131586
27	FCFBEA FCFBFD 19	FAF9FB FCFBFD 131586
28	FCFBE7 FCFBFD 22	FAF9FB FCFBFD 131586
29		FAF9FB FCFBFD 131586
30		

Table 1. The data hexadecimal file contents for steganographic image and pure image

The program code it written be in ***code window.*** The user interface is designed very simple and available only with the necessary function. The user interface is shown at the figure 3.

VI. CONCLUSION

Invisable communication is very present in wholle internet virtuel communication today. Two important uses of data hiding in digital media are to provide proof of the copyright, and assurance of content integrity. Other applications of data hiding, such as the inclusion of augmentation data, need not be invariant to detection or removal, since these data are there for the benefit of both the author and the content consumer.

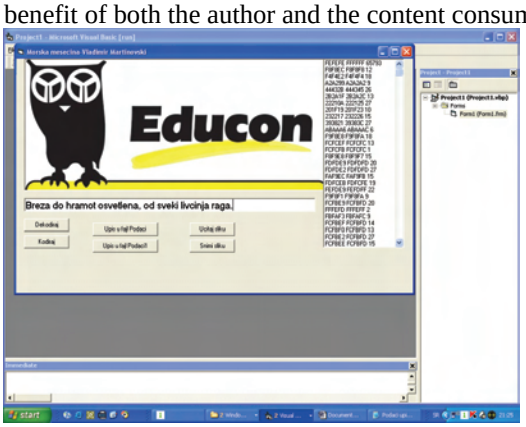

Fig 3. User interface with during embedding message

Thus, the techniques used for data hiding vary depending on the quantity of data being hidden and the required invariance of those data to manipulation [8]. Since no one method is capable of achieving all these goals, a class of processes is needed to span the range of possible applications.

The technical challenges of data hiding are formidable. Any "holes" to fill with data in a host signal, either statistical or perceptual, are likely 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.

REFERENCES

[1] Ames Laboratory, Finding Computer Files Hidden In Plain Sight, May 24, 2006, http://www.fbi.gov/cgi-bin/outside.cgi

[2] Gary C. Kessler, An Overview of Steganography for the Computer Forensics Examiner, Forensic Science Communications, July 2004, Volume 6, Number 3, page 1-27

[3] Bauer, F. L. Decrypted Secrets: Methods and Maxims of Cryptology, 3rd ed. Springer-Verlag, New York, 2002.

[4] Artz, D. Digital Steganography: Hidding data within data, IEEE Internet Computing, 2001, Vol. 3, pag. 75-80

[5] Farid, H. Detecting steganographic messages in Digital Images, Technical Report TR2001- 412, Dartmouth College, Computer Science Department, 2001, http://www.cs.dartmouth.edu/~farid/publications

[6] Julijana Mirčevski, Biljana Djokić, Nikola Popović, Moderne softverske tehnike u prepoznavanju proskribovanih kompjuterskih sadržaja, II Konferencija ZITEH, Tara, novembar 2006

[7] Julijana Mirčevski, Biljana Djokić, Nikola Popović, The wavelet transform based software suitable for a digital contents analyze, V naucno-strucni skup "Nove tehnologije i standardi: digitalizacija nacionalne bastine ", 1-3. jun, Beograd

[8] Steganography, hiding text in images, http://bapuli.reflectionsindia.org/article32006.htm