Text Steganography Using Language Remarks

Ammar Odeh[1] *, Khaled Elleithy, Miad Faezipour*

Abstract –With the rapid growth of networking mechanisms, where large amount of data can be transferred between users over different media, the necessity of secure systems to maintain data privacy increases significantly. Different techniques have been introduced to encrypt data during the transfer process to avoid any kind of attack. One of these techniques is to hide the data inside another file which is called Steganography. In steganography, data is hidden inside a carrier file where anyone can see, but the hidden data inside it cannot be discovered. To this end, good algorithms can avoid the suspicion of having any attacker by applying some criteria before sending the data. In this paper, we present an algorithm to hide data using a text file as a carrier. Left-Right Remarks that represent Unicode symbols are used to hide the data inside the text file. Moreover, our algorithm can be applied in different size textual data.

Keywords: Steganography, Carrier file, Text Steganography, Information Hiding, Stegoanalysis.

I. INTRODUCTION

A. Background

Steganography is a security mechanism used to hide data inside a carrier file such as image, sound, video, or text [1]. The main idea is to hide data inside the carrier file and then placing the Stego file in some transport media. Stegoanalysis will start analyzing the data if there is any suspicion about the carrier file. Some file properties will be basic rules for the analyzer to discover the hidden data. The file size and file format are examples of such properties. As shown in Figure 1, Steganography is mainly classified into four categories depending on the type of the carrier file, i.e. image, audio, video, or text. Moreover, Text Steganography can be classified into different categories depending on the file application.

Most of Steganography algorithms are applied on images which contain huge amount of data. The Least Significant Bit replacement algorithm (LSB) is one such Steganography algorithm [2]. Other complex algorithms have also been introduced to be applied on images. However, the main problems are[3]:

1. File Size :- Image file sizes are already relatively large compared to other files.
2. Image Distortion: - The replacement of some bits may destroy/distort the image, and this will enable the Stegoanalysis to acquire the hidden data [4].
3. Deterministic Changes: The same deterministic algorithm will produce the same distribution bits over the image and this will produce the same hidden image area style. In other words, if we try to replace white pixels by red ones, all white pixels will be converted to red, and this way, the original file could be easily extracted.

Audio carrier files also have some weak points, since any audio signal can be converted and processed in frequency domain and by computing the lower control limit and upper control limit, we can deduce if there are any hidden data in that file. Video carrier files have the disadvantages of merging the weaknesses in sound and image files [5][6].

Text files represent the smallest files in terms of size that can be used to transfer data from sender to receiver, when compared with the other carrier files [7]. Moreover, huge amount of textual data over the internet enables us to hide data over different websites and update those websites with a new style of hidden information that can be

[1] University of Bridgeport, Bridgeport CT 06604, aodeh@bridgeport.edu

embedded within the files. On other hand, text files represent the most difficult Steganography carrier files that do not have redundant patterns like other carrier files [3].

B. Main Contributions and Paper Organization

A promising text steganography algorithm is presented in this paper. The main idea is to use the Right-to-Left Remark (U200F) and the Left-to-Right Remark (U200E) to hide secret data. In our algorithm, we also suggest optimization techniques to offer the highest performance to achieve "Magic Triangle Concepts" for Steganography; that is, the function ability to achieve transparency, robustness, and hiding capacity.

The rest of this paper is organized as follows. In Section II we discuss previous text Steganography techniques. Our proposed Remarks algorithm is discussed in Section III. Discussion and analysis of our algorithm are also provided in the same Section. Simulation results are provided in Section IV. Finally, concluding remarks are offered in Section V.

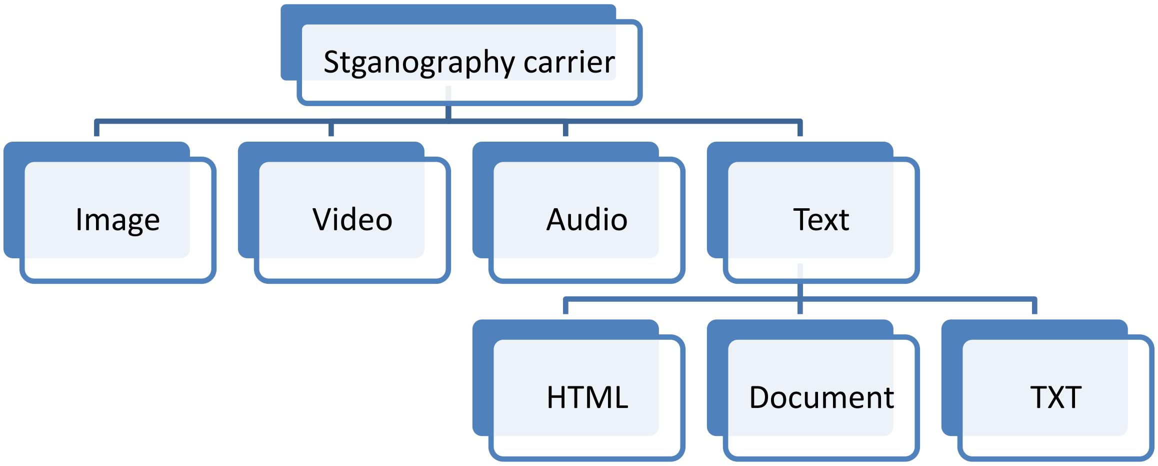

Figure 1. Steganography Carrier Media classification

II. RELATED WORKS

Text Steganography can be classified into three categories depending on the hidden information methods; linguistic, format, and random. Different linguistic methods are classified into two categories. The first one is syntax and the other one is semantic methods [8]. These methods have been developed by creating a dictionary of synonyms and creating representations of each word by bit. Authors in [8] presented a synonyms algorithm to hide data in Bahasa Melayu language, where the hidden algorithm was divided into two phases. The first step converted hidden message into binary code using ASCII codes. Then, a synonyms file was created, where the sender and recipient must have same word list to encrypt and decrypt the message. If the sender wants to insert a zero in the text, there is no need for word replacement. Otherwise, the word is replaced from the synonyms file. The same strategy will be iterated until the end of the secret message is reached. The recipient can decrypt the message by an inverting strategy and comparing if a replacement occurs, in which case the secret code is 1.

Another similar technique was presented in [9]. The algorithm consisted of three input sources; natural language, secret message and the key; and one output which was the Stego-object. By creating lexical substitutions set and variant forms of the same word, after the first scan, the system will recognize each word and to which set it belongs to. The lexical analyzer was then used for Chinese language to embed the correct word in the carrier files and take the context into consideration.

In [10], the authors introduced two linguistic methods using Telugu language (spoken language in the state of Andhra and other states of India). The first method used one of Telugu characteristics by classifying characters into two groups, where the first group would pass 0 and the other group would pass 1. The other method applied Telugu language punctuation marks by distributing them into four groups, each group used to pass two bits.

Another Text Stegonography algorithm was introduced in [11], where the space character was added after words and two bits were encoded. Depending on the number of word letters, and the number of space characters after that word, one of the values in the set {00, 01, 10, 11} would be passed. Authors in [3] also introduced another space method. Single spaces were used to pass 0, and double spaces were used to pass 1. The previous two methods have a problem in which a word processor highlights the additional spaces.

In [11] a new method was introduced to hide data inside Telugu text by horizontally shifting inherent vowel signs. The main advantage of this method is that huge amount of data can be hidden inside the text file. Another algorithm was introduced in [12] by merging between three languages Chinese, Arabic, and English. At the beginning, the authors created two tables; the first one storing Arabic Diacritics and the other table storing English letters. By translating Chinese text into English sentences, each English letter would correspond to two Arabic Diacritics. Then, the Arabic text was created which contained selected Diacritics.

III. PROPOSED ALGORITHM

In this work, the idea is to hide data inside a word file without any change in the file format. Stegoanalysis will try to analyze the file content and formatting. If there is any change in the file format, it can catch the hidden data. In our algorithm, we will use the Right-to-Left Remark (U200F) symbol "⌉" and the Left-to-Right Remark (U200E) symbol "⌈" to hide bits inside the message. Our method will not change the format of the file and can also be applied to different languages regardless of the UNICODE or ASCII coding. Moreover, it is easy to apply this method using Microsoft office word application to hide data.

To avoid the retyping problem that the attacker may employ, we convert our file to PDF, which prevents anyone from ediingt it.

Scenarios to hide the data are as follows:

1. (00) add nothing
2. (01) add Right-to-Left Remark (U200F)
3. (10) Left-to-Right Remark (U200E)
4. (11) Left-to-Right Remark (U200E), Right-to-Left Remark (U200F)

By applying one of these four cases, we can hide data without any changes in the file information.

A. Algorithm I: Hiding Data

```
Input: - Carrier file, hidden bits file

Output: - Stego file (embedded U200E && U200F file)

Step1:- Choose any DOC file

Step 2. Repeat while !(EOF)// repeat until the end of the  hidden file

Step3: Embed hidden data in the selected file

        Step 3a. Start from first letter of the carrier file

        Step 3b. Pack out the first two hidden bits

              If 00 then no U200F nor U200E

              Else if 01 then there is U200F

              Else if 10 then there is U200E

              Else add U200F and U200E.
```

```
Step 5: Go to step 2
Step 6 : Save file as PDF then send it to other side.
```

B. Algorithm II: Data Extraction

```
Input:-Stegofile
Output: - Secure data, original file
Step1:- Open PDF Message
Step 2. Repeat while!(EOF)// repeat until the end of Stego file
Step 3: Embed hidden data in selected file
        Step 3a. Separate each letter
        Step 3b.
              If there is nothing then 00
              Else if only U200F then 01
              Else if U200E then it's a 10
              Else, 11
Step 4: Go to step 3
Step 5: Read hidden data.
```

C. Algorithm: Optimization

Our algorithm has some main advantages which are listed below. Other advantages are also provided in Section IV. The main advantages are:

a) File format will not be affected by embedding the Stego data
b) The algorithm be applied to any language

However, the file size depends on hidden data, which may increase dramatically. Therefore, we suggest the following solution to solve the file size issue.

Before we embed data, we will collect statistical information about the percentage of ones and zeros and apply the following strategy:

$$f(x) = \begin{cases} if \ \ zero's > one's & Apply\ the\ \ same\ algorithm \\ if\ one's > zero's & Switch\ betweeen\ \text{Scenario 1 and Scenario 4} \end{cases} \qquad (1)$$

The best case would be the case where all hidden data are zeros or ones. In this case, the file size will not change at all. However, the worst case is when half of the hidden data are zeros and the other half are ones. Therefore, the best way to optimize our work is to find the largest sequence of string that contains zeros or ones. The file size can be then optimized by considering the relationship in Equation 1.

IV. Simulation results

Our simulation results are divided into two parts. The first one is concerned about which optimization step is employed, as shown in Table I. We created secret messages, converted the messages into ASCII, and computed the number of ones and zeros in each message. Based on Equation 1, the table provides us with a decision as to what would be the most optimized step to proceed with.

Table I. Optimization algorithm decision

Message	Number of Bits	Applied algorithm
Steganography	104	**Switch betweeen Scenario 1 and Scenario 4**
How are you	88	**Apply the same algorithm**
See You	56	**Switch betweeen Scenario 1 and Scenario 4**
At 10	40	**Apply the same algorithm**
See You At 10	104	**Apply the same algorithm**

From our simulation results we conclude that the best way to optimize the embedded message with respect to the file size is to separate our message word by word (where the space binary code is 00100000), and apply formulation (1) to each word. For example, if our secret message is “See You At 10”, and if we apply the Scenario 1, the file size would increase, and this may lead to violating one of the important steganography concepts; transparency. In contrast, if we split the message into parts, and apply the best scenario to each part, one case is that the message “See You At 10” could be divided into two parts, where “See You” will use scenario 4, and “At 10” will use scenario 1. By using the switching scenarios strategy, storage space will be saved as much as possible, and this will improve the transparency goal.

In Table II, we analyze the ability of a few websites to hide bits and also compute the capacity ratio for each (see Equation 2). In our experiments, we assume that hidden bits are inserted between any two words to make it easier to decrypt by finding the space in the file and then finding the Remarks.

Table II. The capacity of articles in web pages for hiding data

#	Website Article	Number of words that can be embedded	Text Size (Kilo Byte)	Capacity Ratio
1	www.nydailynews.com	826	8.8	674
2	www.aljazeera.com	1658	18.7	637
3	www.englisharticles.info	1351	15.9	610
4	www.latimes.com	1208	14.8	586

$$\textit{Capacity Ratio} = (\textit{Number of hidden bits/carrier file size})\ \%100 \qquad (2)$$

It's interesting to note that the average number of word letters in any English file is 9.2 letters [13]. The capacity ratio can be calculated from Equation 2. In addition, the proposed Remarks algorithm can be applied regardless of the language being used.

In summary, the Remarks algorithm has the following advantages:

1. Language independent: - Remarks algorithm can be applied in any language. This feature enables users to hide data in different file formats (Unicode, ASCII). This is while other algorithms depend on language characteristics, which limits the algorithm flexibility.

2. Improved transparency: - This algorithm improves the transparency feature since the Stego file format seems as the original file.

3. File format: - Our method is not dependent on any special format. This allows the use of the carrier text in different formats such as HTML pages, Microsoft Word documents or even plain text format.

4. Algorithm optimization: - Our method suggests optimization steps to reduce the file size change.

5. Hiding capacity: Remarks algorithm enable users to hide huge amount of data between two letters. Any two users can determine where the suitable place to insert bits would be. In our simulations, we used the space between two words to hide one word, where the whole message can also be hidden in one space.

V. CONCLUSION

Different algorithms have been presented to hide data inside text files. Some of these methods were designed to be applied in specific languages [8][9], while others can be applied regardless of the language. In this paper, we presented a promising algorithm that can be used to hide data inside text files of any language by using Remarks (Right, Left). In our method, we pass two bits in each symbol. Moreover, we suggest optimization techniques that can be used to minimize the file size and insert huge amount of data.

REFERENCES

[1] V. Potdar, E. Chang. "Visibly Invisible: Ciphertext as a Steganographic Carrier," *Proceedings of the 4th International Network Conference (INC2004), pp. 385–391, Plymouth, U.K.,* July 6–9, 2004.

[2] T. Morkel, J.H.P. Eloff and M.S. Olivier, "An Overview of Image Steganography," in H.S. Venter, J.H.P. Eloff, L. Labuschagne and M.M. Eloff (eds), *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa,* June/July 2005, *(Published electronically*).

[3] W. Bender,D. Gruhl, N. Morimoto, A. Lu, "Techniques for Data Hiding," *IBM Systems Journal, Vol. 35, pp. 313 - 336*, 1996.

[4] N. Johnson, S. Katzenbeisser, "A Survey of Steganographic Techniques," *Chapter 3 in Stefan Katzenbeisser (ed.), Fabien A. P. Petitcolas (ed.) Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Books*, 2000.

[5] P. Jayaram, H. Ranganatha, H. Anupama , "Information Hiding Audio Steganography - A Survey," *International Journal of Multimedia & Its Applications (IJMA), Vol. 3, pp. 86-9*6, Aug. 2011.

[6] A. Al-Othmani, A. Abdul, A. Zeki, "A Survey on Steganography Techniques in Real Time Audio Signals and Evaluation," *IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, pp. 30-37,* Jan. 2012.

[7] H. Singh, P. Singh, K. Saroha, "A Survey on Text Based Steganography," *Proceedings of the 3rd National Conference, pp. 3-9, INDIACom*, 2009.

[8] R. Prasad, K. Alla, "A New Approach to Telugu Text Steganography," *Proceedings of the IEEE Wireless Technology and Applications Conference (ISWTA), pp. 60 - 65*, 2011.

[9] L. Yuling, S. Xingming, G. Can, W. Hong, "An Efficient Linguistic Steganography for Chinese Text," *Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 2094 - 2097*, 2007.

[10] S. Bhattacharyya, I. Banerjee, G. Sanyal, "A Novel Approach of Secure Text Based Steganography Model using Word Mapping Method (WMM)," *International Journal of Computer and Information Engineering, Vol. 4, No. 2, pp. 96-103*, 2010.

[11] S. Tech, S. Pothalaiah, K. Babu, "A New Approach to Telugu Text Steganography by Shifting Inherent Vowel," *International Journal of Engineering Science and Technology, Vol. 2, No. 12, pp. 7203-7214*, 2010.

[12] A. Shakir, G. Xuemai, J. Min, "Chinese Language Steganography using the Arabic Diacritics as a Covered Media," *International Journal of Computer Applications, Vol. 11, No. 1, pp. 43-46*, Dec. 2010.

[13] R.D. Smith, "Distinct Word Length Frequencies: Distributions and Symbol Entropies," *Journal of Glottometrics 23, pp. 7-22*, 2012.

Ammar Odeh is a PhD. Student in University of Bridgeport. He earned the M.S. degree in Computer Science College of King Abdullah II School for Information Technology (KASIT) at the University of Jordan in Dec. 2005 and the B.Sc. in Computer Science from the Hashemite University. He has worked as a Lab Supervisor in Philadelphia University (Jordan) and Lecturer in Philadelphia University for the ICDL courses and as technical support for online examinations for two years. He served as a Lecturer at the IT, (ACS,CIS ,CS) Department of Philadelphia University in Jordan, and also worked at the Ministry of Higher Education (Oman, Sur College of Applied Science) for two years. Ammar joined the University of Bridgeport as a PhD student of Computer Science and Engineering in August 2011. His area of concentration is reverse software engineering, computer security, and wireless networks. Specifically, he is working on the enhancement of computer security for data transmission over wireless networks. He is also actively involved in academic community, outreach activities and student recruiting and advising.

Dr. Khaled Elleithy is the Associate Dean for Graduate Studies in the School of Engineering at the University of Bridgeport. He has research interests are in the areas of network security, mobile communications, and formal approaches for design and verification. He has published more than two hundereds research papers in international journals and conferences in his areas of expertise. Dr. Elleithy is the co-chair of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE). CISSE is the first Engineering/Computing and Systems Research E-Conference in the world to be completely conducted online in real-time via the internet and was successfully running for six years. Dr. Elleithy is the editor or co-editor of 12 books published by Springer for advances on Innovations and Advanced Techniques in Systems, Computing Sciences and Software.

Dr. Miad Faezipour is an Assistant Professor in the Computer Science and Engineering program at the University of Bridgeport and the director of the D-BEST Lab since July 2011. Prior to joining UB, she has been a Post-Doctoral Research Associate at the University of Texas at Dallas collaborating with the Center for Integrated Circuits and Systems and the Quality of Life Technology laboratories. She received the B.Sc. in Electrical Engineering from the University of Tehran, Tehran, Iran and the M.Sc. and Ph.D. in Electrical Engineering from the University of Texas at Dallas. Her research interests lie in the broad area of biomedical signal processing and behavior analysis techniques, high-speed packet processing architectures, and digital/embedded systems. Dr. Faezipour is a member of IEEE and IEEE women in engineering.