EUROPEAN ACADEMIC RESEARCH, VOL. I, ISSUE 7/ OCTOBER 2013
ISSN 2286-4822, www.euacademic.org
IMPACT FACTOR: 0.485 (GIF)
DRJI Value: 5.9 (B+)

Data Hiding Method in Ignored Processing Area of Microsoft Compound Document File Format

ABDULMONEM S. RAHMA
Computer Science Department, University of Technology
Baghdad, Iraq
WESAM S. BHAYA
College of Computer Technology, University of Babylon
Babil, Iraq
DHAMYAA A. AL-NASRAWI
Computer Science Department, College of Science
Kerbala University, Karbala, Iraq

Abstract:

One of the important issue in network is security, especially when computer usage is increasing for both social and business areas. So, the secure communications through public and private channels are the main objective for researches. Data hiding is one of approach to obtain a secure communication medium and protecting the information during transmission. Text steganography is most challenging because redundant information in text documents is very less as compared to the audio and images. In this paper, insert the secret message in specified areas in a document that will be ignored by the processing. For example, in MS-Word document contain an 0x0D or paragraph marker, and the process will stop when it reaches the 0x0D because it is not displayed to or editable by the user, because it is outside of any document part. The message was encrypted before embedding process, in order to increase the security levels by Triple_DES. The capacity of hiding in binary file format based on the available ignored processing area. The results show that there is no increasing in size of stego document and the proposed method has an excellent perceptual transparency because there is no difference between cover document and stego document when open them in Microsoft Office.

Key words: Data Hiding, Ignored Processing Ares, Microsoft Compound Document File Format, MS-Word document.

1. INTRODUCTION

Steganography is a process that involves hiding a message in a proper carrier such as image, text or voice. It is means "covered or hidden writing" in Greek. For secret communication purpose, steganography protects both messages and its destination parties so that the authenticated receiver only knows the existence of the messages that the carrier sent. There is a large amount of works being done by a variety of research in steganography (Bhattacharyya *et al*. 2011). Steganography is the hiding of a secret information within an ordinary information in some other media, and the extraction of it at its destination. The information to be hidden is secret message and the medium in which the information is hidden is called cover document. Stego-document is the cover document with hidden message. The used algorithms for hiding the message in the cover medium and extracting the hidden message from the stego-document at the sender and receiver end respectively is called stego system (Garg 2011).

In this paper, we use ignored processing areas in a Ms-Word document for data hiding. That will be ignored by the processing. In order to increase the security levels of proposed data hiding system, the secret message was encrypted before the embedding process by Triple_DES.

The organization of the paper is as follow: Section 2 shows the related works that have been done in the field of data hiding in text media; definition of Ms-Word file format is demonstrated in section 3; Section 4 presents the proposed data hiding method in detail; Section 5 displays the results and concludes the paper.

2. RELATED WORKS

There are a number of researches that have already explored new steganographic techniques in texts, such as white spaces (Matthew 1998), Synonyms (Niimi et al. 2003), Word

Shifting (Kim 2003), and Line shifting (Alattar and Alattar 2004).

This paper focuses on researches which used binary file format for data hiding. L. Tsung-Yuan and T. Wen-Hsiang proposed a method for hiding by taking text segments in the document and degenerating them, mimicking to be the work of an author with inferior writing skills, with the secret message embedded in the choices of degenerations. The degenerations are then revised with the changes being tracked, making it appear as if a cautious author is correcting the mistakes. The change tracking information contained in the stego document allows the original cover, the degenerated document, and, hence, the secret message to be recovered (Tsung-Yuan and Wen-Hsiang 2007).

Amani Y. proposed a hiding method with two phases: Cover generation phase, where the cover is a document of Microsoft Word Document file format 2003 (doc) and will appear to be the product of a collaborative writing effort between authors, and the embedding phase in which the hiding text string appears in the unused block of binary file format of that cover document. The researcher concludes that the cover generation will increase the security of hidden system and the stego text will not be affected by copying or mailing the stego document, while the size of stego document is acceptable (Amani 2009).

3. Definition of MS-Word File Format

A file format is the presence of a file in terms of how the data within the file is organized. A program that uses the data in a file must be able to recognize and possibly access data within the file. A particular file format is often indicated as part of a file's name by a file name extension (suffix). The Microsoft Office File Formats documentation provides detailed technical specifications for file formats implemented in certain Microsoft Office applications (Microsoft 2013). The structure of Documents in Word is hierarchical, different types of properties applying to different units in the hierarchy: section, paragraph, characters,…etc. see figure 1:

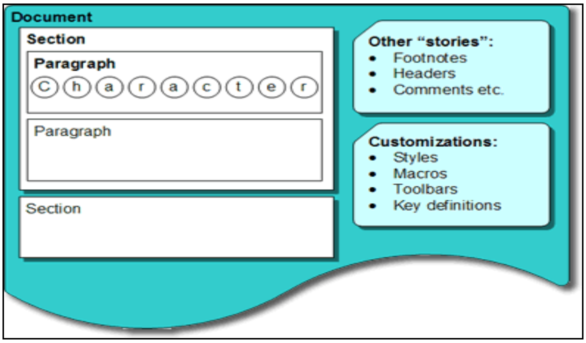


Figure 1: Hierarchical Structure of Word Document

MS-office file is a Compound File which is Microsoft's implementation of structure storage. Structured storage allows hierarchical storage of information within a single file. Elements of a structured storage object are storages and streams. Storages are related to directories, and streams are related to files. Compound file binary storage format is like a file system, similar to FAT. All of these data are potentially fragmented across the file in various sectors described by the internal FAT.

The Microsoft Compound Document File Format (MCDFF) 2003 is a document file format based on OLE (Object Linking and Embedding), which is used for saving various resources as an integrated document in Microsoft (Microsoft Corporation 2007).

Compound file binary storage format is a method of storing structure of information within the file which is introduced by Microsoft. There may be different paragraphs, each paragraph may have different formats, fonts and colors, there also may be pictures between paragraphs in a full file.

A Word (.doc) file consists of a:

- Word Document (Main stream)
- Summary information stream
- Table stream
- Data stream
- Custom XML storage (Added in Word 2007)

Zero or more object streams contain private data for OLE 2.0 objects embedded within the Word document (Microsoft Corporation 2007).

The root component is 'MS Word' component which contains several streams and one storage item. Different parts of the document, such as the actual contents, any table inserted, the *CompObj* associated with the DLL files for the objects, the Summary Information for the content, any image inserted, and the Document Summary Information, all take the form of streams under the root component. The *ObjectPool* is the collective storage of all the sub-storage components (Khushbu 2006).

3.1. FORMAT OF THE MAIN STREAM

The main stream of a Word binary file (complex format) consists of the Word file header (FIB), the text, and the formatting information. The header of a Word file begins at offset 0x00 in the file. This gives the beginning offset and lengths of the document's text stream and subsidiary data structures within the file. It also stores other file status information.

The FIB contains a "magic number" and pointers to the various other parts of the file, as well as information about the length of the file. The FIB is defined in the structure definition section of this document (Microsoft Corporation 2007).

3.2. COMPOUND DOCUMENT HEADER

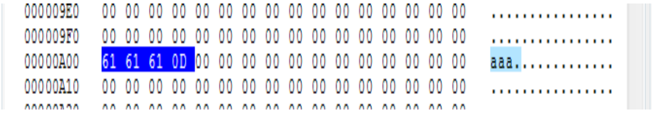
The compound document header (simply "header" in the following) contains all data needed to start reading a compound document file. The header is always located at the beginning of the file; this implies that the first sector (with SecID 0) always starts at file offset 512. The first 64 bits of the header form id or magic number identifier of office file (OpenOffice 2004). The header also contains an array of block numbers which refer to blocks in the file. When these blocks are read together they form the Block Allocation Table. The header also contains a pointer to the first element in the property table, also known as the root element, and a pointer to the small Block Allocation Table (SBAT) (Marc 2007).

The block allocation table or BAT, along with the property table specifies which blocks in the file system belong to which files.

3.3. DOCUMENT PARTS

The range of CPs (Character Position) in a document is separated into multiple logical parts. Many features operate within the individual parts and use CPs relative to the beginning of the part in which they operate rather than relative to the beginning of the document. This section defines the document parts and specifies the corresponding range of CPs (OpenOffice 2004).

All documents MUST include a non-empty Main Document part. In addition, if any of the other document parts are non-empty, the document MUST include one additional paragraph mark character (Unicode 0x000D) beyond the end of the last non-empty document part. That character is not displayed to or editable by the user, because it is outside of any document part, in the following displayed the contents of documents (aaa):



```
000009E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000009F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000A00  61 61 61 0D 00 00 00 00 00 00 00 00 00 00 00 00   aaa.............
00000A10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Main Document: The main document contains all content outside any of the specialized document parts, including anchors that specify where content from the other document parts appears. The main document begins at CP zero, and is *FibRgLw97.ccpText* characters long. The last character in the main document MUST be a paragraph mark (Unicode 0x000D).

Footnotes: The footnote document contains all of the content in the footnotes. It begins at the CP immediately following the Main Document, and is *FibRgLw97.ccpFtn* characters long.

Headers: The header document contains all content in headers and footers as well as the footnote and endnote separators. It begins immediately after the footnote document and is *FibRgLw97.ccpHdd* characters long.

Comments: The comment document contains all of the content in the comments. It begins at the CP immediately following the Header Document and is *FibRgLw97.ccpAtn* characters long.

Endnotes: The endnote document contains all of the content in the endnotes. It begins at the CP that immediately follows the Comment Document and is *FibRgLw97.ccpEdn* characters long.

Textboxes: The textbox document contains all of the content in the textboxes whose anchors are in the Main Document. It begins at the CP immediately following the Endnote Document and is *FibRgLw97.ccpTxbx* characters long.

Header Textboxes: The header textbox document contains all of the content in the textboxes whose anchors are in the Header Document. It begins at the CP immediately following the Textbox Document and is *FibRgLw97.ccpHdrTxbx* characters long (OpenOffice 2004).

4. PROPOSED DATA HIDING METHOD

In this method, insert the secret message in specified areas in a document that will be ignored by the processing. For example, the MS-Word document contains an 0x0D or paragraph marker, and the process will stop when it reaches the 0x0D because it is not displayed to or editable by the user, because it is outside of any document part.

The structure of proposed system is explained in figure 2.

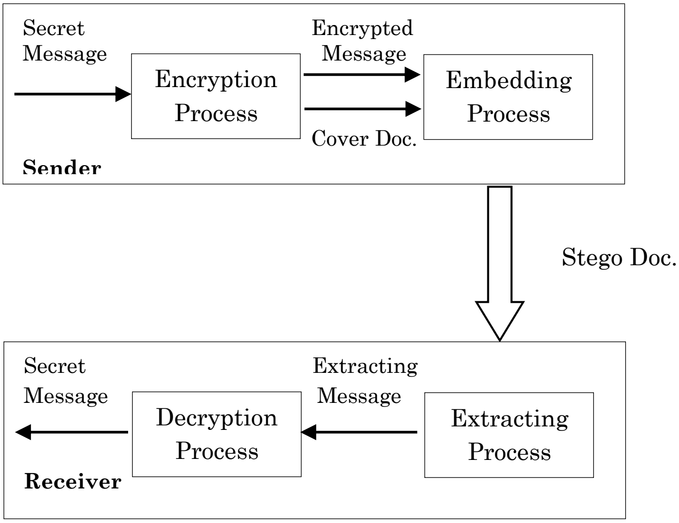


Figure 2: Structure of Proposed Hiding System

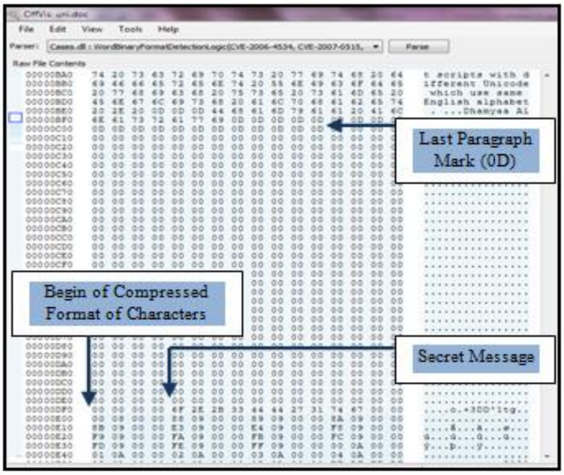
The secret message was entered in English alphabets and then encrypted by triple_DES function built-in visual C#.net functions.

Embedding Process

The idea of this method is to hide secret data in specified areas after document parts. Table 1. and Figure 3. show areas that will be hiding it.

Table 1: Areas that will be hiding data

Offset(h)	Areas	Size
0x00	Header	512 Bytes
0x200	FIB	1618 Bytes
0xA00	Document parts	Variable
Unknown	Compressed format of char stchpxFKps	Variable
Unknown	Compressed format of paragraph stchpxFKps	Variable
	. . .	


OffVis: uni.doc
File
Edit
View
Tools
Help
Parser: Cases.dll : WordBinaryFormatDetectionLogic(CVE-2006-4534, CVE-2007-0515,...
Parse
Raw File Contents
Last Paragraph Mark (0D)
Begin of Compressed Format of Characters
Secret Message

```
00000BA0   74 20 73 63 72 69 70 74 73 20 77 69 74 68 20 64    t scripts with d
00000BB0   69 66 66 65 72 65 6E 74 20 55 6E 69 63 6F 64 65    ifferent Unicode
00000BC0   20 77 68 69 63 68 20 75 73 65 20 73 61 6D 65 20     which use same 
00000BD0   45 6E 67 6C 69 73 68 20 61 6C 70 68 61 62 65 74    English alphabet
00000BE0   20 2E 20 0D 0D 0D 44 68 61 6D 79 61 61 20 41 6C     . ...Dhamyaa Al
00000BF0   4E 61 73 72 61 77 69 0D 0D 0D 0D 0D
00000C00   0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D
00000C10   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C20   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C30   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C40   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C50   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C60   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C70   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000C80   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000C90   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CA0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CB0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CC0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CD0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CE0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000CF0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00    ................
                                   00 00 00 00 00 00 00 00
00000D80   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000D90   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000DA0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000DB0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000DC0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000DD0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000DE0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
00000DF0   00 00 00 00 6F 2E 2B 33 44 44 27 31 74 67 00 00    ....o.+3DD'1tg..
00000E00   00 08 00 00 88 09 00 00 89 09 00 00 8A 09 00 00    ................
00000E10   8B 09 00 00 E3 09 00 00 E4 09 00 00 F8 09 00 00    ....ã...ä...ø...
00000E20   F9 09 00 00 FA 09 00 00 FB 09 00 00 FC 09 00 00    ù...ú...û...ü...
00000E30   FD 09 00 00 FE 09 00 00 FF 09 00 00 00 0A 00 00    ý...þ...ÿ.......
00000E40   01 0A 00 00 02 0A 00 00 03 0A 00 00 04 0A 00 00    ................
```

Figure 3: Embedding Data in Binary File Format

The steps of embedding secret data are explained in the following algorithm:

The embedding process start at location CFC-1, then decreases it to embed another symbol of secret message.

Extracting Process

On the other hand, the extracting stage is implemented by many functions. Based on this method, the extracting process depends on the location of the compressed format of characters area which has previously been explained. The extracting process in this method starts from CFC-1 location. Then you have to get current symbols, decrease CFC, and repeat this until the current symbol is 0x00.

5. RESULTS AND CONCLUSIONS

Steganographic techniques embed a message inside a cover. Many features characterize the methods depending on the application. These are: capacity, invisibility, security, and robustness.

- *Capacity*: The total number of bits hidden and successfully recovered by the data hiding system.
- *Invisibility*: is based on the properties of the human visual system. The embedded information must be imperceptible, that means the human is unable to distinguish between carriers and stego documents.
- *Security*: It depends on the total information about the embedded algorithm and secret key. The embedded algorithm is secure if the embedded information is not subject to removal after being discovered by the attacker.
- *Robustness*: the ability of the embedded data to remain unbroken if the steganographic system undergoes transformation (Soniya 2011).

Extracting Algorithm of Binary File Format

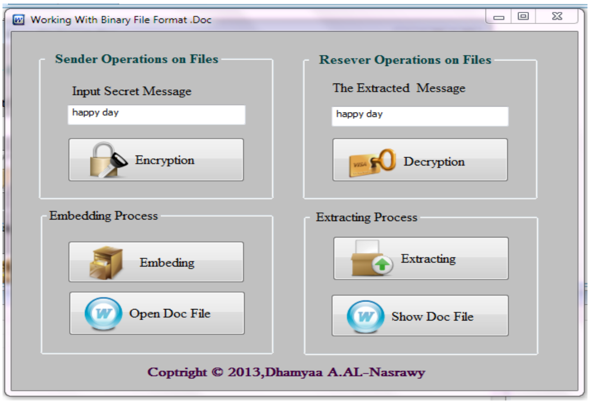
Input : *Stego document*

Output : *Secret Message*

```
1. Open Stego document.
2. Get location of compressed format of characters CFC.
3. Decrease CFC
6. while current symbol not 0
      - extract current symbol
      - decrease CFC
7. Return Secret Message.
```

In this section, the experimental results are given and interpreted for all the contributions that are presented in our work. The proposed method was evaluated based on above features, as well as explaining the detecting capability of the existing hidden message.

The proposed system provides friendly GUI that is easy and helpful to implement by user to manage the encryption and hiding processes. The proposed method is based on hiding data in binary file format MS-word documents with extension (.doc). Figure 4 shows the GUI of proposed system.



Working With Binary File Format .Doc

Sender Operations on Files

Input Secret Message: happy day

Reserver Operations on Files

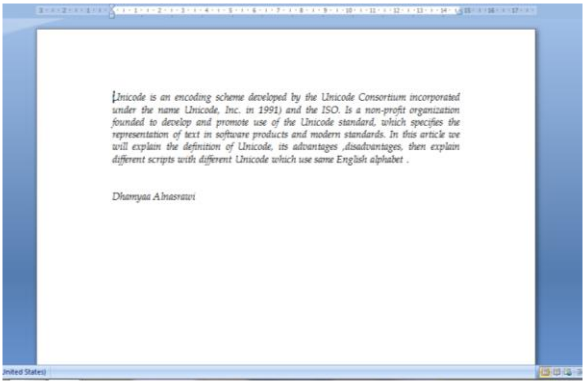
The Extracted Message: happy day

Coptright © 2013,Dhamyaa A.AL-Nasrawy

Figure 4: GUI of proposed System

The steps of proposed system were explained above with more details, open cover document, encryption, and hiding processes. On the other hand, the receiver implements the inverse of these steps, open stego document, extracting hidden data, and decryption data.

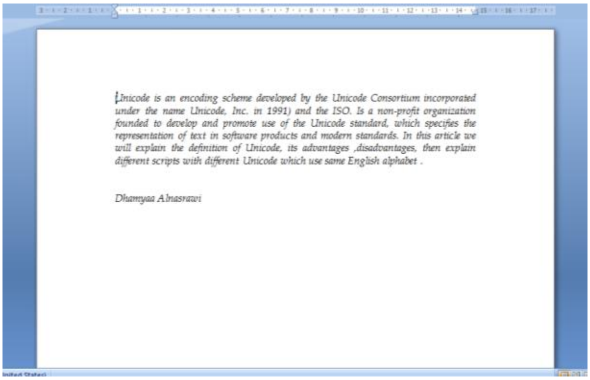
According to this method there cannot be noted changes in size of stego file, because the area of storage was structured already. The results of file before and after the embedding process were explained in the following figures:



Unicode is an encoding scheme developed by the Unicode Consortium incorporated under the name Unicode, Inc. in 1991) and the ISO. Is a non-profit organization founded to develop and promote use of the Unicode standard, which specifies the representation of text in software products and modern standards. In this article we will explain the definition of Unicode, its advantages ,disadvantages, then explain different scripts with different Unicode which use same English alphabet .

Dhamyaa Alnasrawi

Figure 5: Cover Document for hiding in Binary file format



Unicode is an encoding scheme developed by the Unicode Consortium incorporated under the name Unicode, Inc. in 1991) and the ISO. Is a non-profit organization founded to develop and promote use of the Unicode standard, which specifies the representation of text in software products and modern standards. In this article we will explain the definition of Unicode, its advantages ,disadvantages, then explain different scripts with different Unicode which use same English alphabet .

Dhamyaa Alnasrawi

Figure 6: Stego Document for hiding in Binary file format

Capacity: The capacity of hiding in binary file format based on the available area in the cover document, which is based on the document itself.

Invisibility and Security: The proposed method has an excellent perceptual transparency because there is no difference btween the cover document and stego document when opened in the microsoft office, with a high level of security because the secret message was encrypted before the embedding process. That means that the visual detection can fail (see Figures 5 and 6).

Robustness: This method is robust to digital copy-past operation, which means that copying and pasting the text between computer programs preserve hidden information.

The important point in this mthod is that there is no icreasing in size of the stego document.

BIBLIOGRAPHY

Alattar A., and Alattar O. 2004. “Watermarking Electronic Text Documents Containing Justified Paragraphs and Irregular Line Spacing.” Proceedings of SPIE – Vol 5306, Security, Steganography, and Watermarking of Multimedia Contents VI, 685-695.

Amani Y. 2009. *Steganographic Method for Data Hiding in Microsoft Word Documents Structure by a Change Tracking Technique*. MSc. diss, University of Technology, Baghdad, Iraq.

Bhattacharyya S., Indu P., Dutta S., Biswas A., and Sanyal G. 2011. "Text Steganography using CALP with High Embedding Capacity." *Journal of Global Research in Computer Science* 2(5): 29-36.

Garg, M. 2011. "A Novel Text Steganography Technique Based on Html Documents." *International Journal of Advanced Science and Technology* 35: 129-138.

Khushbu, J. 2006. "Microsoft Office Security, Part one", available at: http://www.securityfocus.com/infocus/1874.

Kim Y., Moon K., and Oh I. 2003. "A Text Watermarking Algorithm based on Word Classification and Inter-word Space Statistics." Proceeding of the 7th International Conference on Document Analysis and Recognition, Aug. 3-6, IEEE Xplore Press, USA, 775-779.

Marc, J. 2007. "POIFS File System Internals", the Apache POI Project, the Apache Software Foundation.

Matthew, K. 1998. "SNOW", Darkside Technologies Pty Ltd ACN 082 444 246 Australia. Last modified 20 June 2013. http://www.darkside.com.au/snow/index.html

Microsoft 2013. "Microsoft Office File Formats", available at:

http://msdn.microsoft.com/enus/library/cc313118(v=office.12).aspx

Microsoft Corporation 2007, "Microsoft Office Word 97-2007 Binary File Format (.doc) Specification", Microsoft Open Specification Promise.

Niimi M., Minewaki S., Noda H., and Kawaguchi E. 2003. “A Framework of Text-based Steganography Using SD Form Semantics Model.” Pacific Rim Workshop on Digital Steganography. Kyushu Institute of Technology, Kitakyushu, Japan.

OpenOffice 2004. "Microsoft Compound Document File Format". available at: www.openoffice.org/sc/compdocfileformat.pdf

Soniya, V. 2011. "Image Steganography Based On Polynomial Functions". 2(3): 13-15.

Tsung-Yuan L, Wen-Hsiang T. 2007. "A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique." IEEE Transactions on Information Forensics and Security, 2(1): 24-30.