Proposed Steganographic Method for Data Hiding in Microsoft Word Documents Structure

*Prof.Dr.*Abul Monem.S.Rahma *, *Dr.H.*B.AbdulWahab*, A.Y.Al-Noori*

Abstract

This paper proposed a method of data hiding by taking advantage of the physical characteristics of computer system and how it stores document file and treating it as a compound file. The unused block in Microsoft Compound Document File Format (MCDFF) is used to hide data. The possibilities provided by Microsoft Word Processor program have also been utilized, such as Tools, to generate cover for hiding. The proposed system embeds steganography text in structure (Binary File Format) of digital and printed text document file which is a file of Microsoft Word Document file (Doc.) using two Processes: *Cover Generation* and *Embedding Processes. Cover Generation Process*: 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 among authors using Track Changes tool.
Embedding Process **hiding text string in unused block of binary file format of that document cover. This paper proposed a new technique, which gives good results, such that the user can hide 63byte in 34KB document cover size with informed about size of empty document=10/11KB, in addition, using Track Changes tool does not affect on hidden data and no problem was detect on hidden data at stego-document mailing or copying.**

* University of Technology

1. Introduction

Steganography is the art and science of communicating in a way which hides the existence of the communication [9]. The goal of steganography is to hide message inside other harmless message in a way that does not allow any others to detect that there is a second secret message present (to avoid drawing suspensions) [9]. Harmless message may be: text, disks, storage devices, network traffic protocols, images, audio, video and any other digitally represented code or transmission [11].

2. Hiding Data in Text

Most of the research concentrated on images, audios and videos as cover because it can hides large amount of information. Written text can be used as a method to transmit secret messages. Only small amounts of data can be hidden when hiding data in text. For that reason there are few studies about information hiding in texts.

2.1 Data Hiding Techniques

The techniques in text hiding are differing from one method to another in the following some popular techniques:

2.1.1 Encoding Information Directly in the Text

Many ways have been proposed to hide information directly in text like Syntactic, Semantics, P.Waynar, Chapman, Translation and HTML.

- **Syntactic method: Where the structure of sentences is transformed without significantly altering their meaning, this method utilizes punctuation, diction [15].**

- **Semantics method: Where words are replaced by their synonyms and/or sentences are transformed via suppression or inclusion of noun phrase coreferences [15].**
- **P.Wayner method: Peter Wayner proposed a Mimic Function which exploits the statistical profile of a message he uses (CFG) to create cover-text and chooses the productions according to the secret message to be transmitted, the secret information is not embedded in the cover, and the cover itself is the secret message Chapman and Davida method: Chapman and Davida proposed a system which consists of two functions, NICETEXT and SCRAMBLE. Given a large dictionary of words of different types, and a style source, describes how words of different types can be used to form a meaningful sentence. NICETEXT transforms secret message bits into sentence by selecting words out of the dictionary which conform to a sentence structure given in style source [1].**
- **Translation- based steganography: Use the expected errors in the translation process especially in machine translation to solve the issue of producing implausible text. Information is hidden in the noise that occurs in language translation [11].**
- **HTML: Information is hidden in HTML files by adding useless spaces and line breaks or by changing the case of letters in the tags [8].**

2.1.2 Encoding Information in the Text Format

Information can be embedded in the format rather than in the message itself. In the following popular methods:

- **Open Space method**

Encode through manipulation of white space (unused space) on the printed page. many methods for using white space like Inter-Sentence Spacing which encode a binary message into a text by placing one or two spaces after sentence, such that one space represents "0" and two spaces represent "1", End-of-line spaces insert spaces at the end of lines and Inter-word-Spaces one space between words is interpreted as a "0", two spaces are between words are interpreted as a "1".

- **Line-Shift Coding:**

Text lines are vertically shifted (moved up or down) according to the secret message bits.

- **Word-shift Coding:**

In this method, codeword are coded in to a document by shifting the horizontal or vertical locations of words within text lines, while maintaining a natural spacing appearance.

- **Feature Encoding:**

Where feature such as Shape, Size, or Position are manipulated .In this method certain text feature are altered, or not altered depending on the codeword [1].

- **Color quantization:**

The main idea of this method is to quantize the color or luminance intensity of each character in such a manner that the human visual system is not able to distinguish between the original and quantized characters, but it can be easily performed by a specialized reader machine [15].

- **Halftone Quantization:**

This method relies on half toning, a widely used printing technology that enables continuous tone images to be printed with one color ink(grayscale) or a few color inks(color) [15].

3. File Structure

A *file format* is a particular way to encode information for storage in a computer file [16]. Each format uses structure (a way to organize data for storing) in a file [5]. There are several types of ways to structure data in a file. The most usual ones are:

3.1 Raw Memory Dumps/Unstructured Formats (RMD)

Earlier file formats used raw data formats that consisted of directly dumping the memory images of one or more structures into the file. This has several drawbacks. Unless the memory images also have reserved spaces for future extensions, extending and improving this type of structured file is very difficult. On the other hand, developing tools for reading and writing these types of files are very simple. The limitations of the unstructured formats led to the development of other types of file formats that could be easily extended and be backward compatible at the same time [16].

3.2 Chunk based Formats (CBF)

In this kind of file structure, each piece of data is embedded in a container that contains a signature identifying the data, as well the length of the data (for binary encoded files). This type of container is called a chunk. The signature is usually called a chunk id, chunk identifier, or tag identifier. With this type of file structure, tools that do not know certain chunk identifiers simply skip

those that they do not understand. Even XML can be considered a kind of chunk based format, since each data element is surrounded by tags which are akin to chunk identifiers [16].

3.3 Directory based Formats (DBF)

This is another extensible format, that closely resembles a file system (OLE Documents are actual file systems), where the file is composed of 'directory entries' that contain the location of the data within the file itself as well as its signatures (and in certain cases its type). Good examples of these types of file structures are disk images, OLE documents [16].

4. Microsoft Word Document and its Components

Documents in word have a hierarchical structure as shown in figure (1).Different types of properties apply to different units in hierarchy [10].

4.1. Annotation and collaboration Microsoft Word Document Tools

As a linguist, will often be working together with someone else on a document either as a co-author, or in a student-teacher relationship. Word has some easy-to-use tools to facilitate such collaborative work [10].

4.1.1 Track Changes

The "Track Changes" tool gives access to a simple method of keeping track of the changes a particular user makes to a document. Insertions will display in color and underlined; deletions and format changes will display in bubbles like comments [10]. Track Changes is a way for Microsoft Word to keep track of the changes make to a document. Track Changes is also known as redline, or redlining.

This is because some industries traditionally draw a vertical red line in the margin to show that some text has changed [14] example of Track Change can be shown in figure (2).

4.1.2 Comments

The "Comment" feature allows comments to be added to the document. In Page Layout view, recent versions of Word will display comments in "bubbles" on the right side of the text (moving text over to make room in the margin for the comment). Comments from different reviewers will appear in different colors [10] example of comments can be shown in figure (3).

5. Microsoft Compound Document File Format (MCDFF)

A word file containing Excel sheet and chart, an image, a table, and some macros is an example of compound file. Files which use MCDFF (Microsoft Compound Document File Format) include output files from MS Office 97-2003, which consists of the applications MS Word, PowerPoint, and Excel [2]. 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 [13]. A storage component may exist as a standalone component. Each storage component may have one or more sub-storage components and stream components. Also the root component may have stream components directly within it [7].

6. Structure of a Word Documents files

Let's take a look at the structure of a Word document with an embedded Excel object, shown in figure (4). The binary format for Microsoft Word 97 and later versions is based on a structure referred to as a .doc file or compound file. A Word .doc file consists of a [13]:

- **Word Document (Main stream)**

- **Summary information stream**
- **Table stream**
- **Data stream**
- **Custom XML storage (Added in Word 2007).**

0 or more object streams which contain private data for OLE 2.0 objects embedded within the Word document [13]. The 'MS Word' component is the root component containing 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. The figure (4) displays samples of the sub-storage Excel component. The Excel Sheet itself is a storage component within the ObjectPool and has its own streams of information the Workbook, Summary Information and Document Summary Information

[7]. The main stream of a Word binary file (complex format) consists of the Word file header (FIB), the text, and the formatting information.

7. MCDFF metadata

MCDFF uses metadata to manage information about Streams, Storage. Metadata of MCDFF is header, Block Allocation Table (BAT), Sector Allocation Table (SAT) and Directory. The exact format structure of these metadata was provided by the Spreadsheet Project of Open Office.org Documentation of the Microsoft Compound Document File Format [3] and the Apache POIFS Project of Apache.org. [12] POIFS file systems are called "file system", they contain multiple embedded files in a manner similar to the traditional file if had a word processor file

with the extension ".doc", would actually have a POIFS file system with a document file archived inside of that file system [12]. Most operating systems, including Microsoft Windows manage hard disk drives by dividing their storage space into units known as *partitions*. So that can access a partition, before being able to store data on a partition, it must *formatted*. Formatting a partition organizes the associated space into what is called a *filesystem*, which provides space for storing the names and attributes of files as well as the data they contain. Microsoft Windows supports several types of filesystems, such as FAT and FAT32, Formatting a disk divides the disk into tracks and sectors, each track is divided into *sectors* sometimes called *disk blocks* as shown in figure (5) where Partitions comprise the *logical structure* of a disk drive, the way humans and most computer programs understand the structure. However, disk drives have an underlying *physical structure* that more closely resembles the actual structure of the hardware. MCDFF uses two types of data unit: Small Block (Sector) and Big Block (Block) [6]. If the Stream size is less than 4096, the file is stored in small blocks and the SBAT is used to walk the small blocks making up the file.

If the file size is 4096 or larger, the file is stored in big blocks and the main BAT is used to walk the big blocks making up the file [12].

7.1 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. The header also contains an *array of block*

***numbers*. These block numbers 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)* [12]. The *block allocation table* or *BAT*, along with the *property table* specifies which blocks in the file system belong to which files [12]. The contents of the compound document header structure are described in the Table (1).**

***property table*, also known as the *root element*, and a pointer to the *small Block Allocation Table (SBAT)* [12]. The *block allocation table* or *BAT*, along with the *property table* specifies which blocks in the file system belong to which files [12]. The contents of the compound document header structure are described in the Table (1).**

7.2 Sector File Offsets [3]

With the values from the header it is possible to calculate a file offset from a SecID:

$$sec\_pos(SecID) = 512 + SecID \cdot sec\_size \quad ...(1)$$

$$= 512 + SecID \cdot 2^{ssz}$$

Example: with ssz = 10 and SecID = 5

$$sec\_pos(SecID) = 512 + SecID \cdot 2^{ssz}$$

$$= 512 + 5 \cdot 2^{10}$$

$$= 512 + 5 \cdot 1024$$

$$= 5632.$$

Note: The previous equation is used to calculate Block Position too.

7.3 Property Table (Directory)

The Property Table is essentially nothing more than the directory system. Properties (directories) are 128 byte records contained within the 512 byte blocks.

Each directory entry refers to storage or a stream in thecompound document. The zero-based index of a directory entry is called *directory entry identifier* (DirID). There is a special directory entry at the beginning of the directory (with the DirID 0). It represents the root storage and is called *root storage entry* [3]. The contents of the directory entry structure are described in the Table (2).

7.4 Block Allocation Table (BAT)

The BAT (Block Allocation Table) is the main table for space within MCDFF, which is needed to read any other Stream in the file [6]. The BAT blocks are pointed at by the bat array contained in the header these blocks form a large table of integers. These integers are block numbers. The Block Allocation Table holds chains of integers [12]. The elements in these chains refer to blocks in the files. The starting block of a file is NOT specified in the BAT. It is specified by the property of a given file. The elements in this BAT are both the block number (within the file minus the header) and the number of the next BAT element in the chain. This can be thought of as a linked list of blocks. The

BAT array contains the links from one block to the next, including the end of chain marker [12].
The BAT format structure is shown in table (3). Here's an example: Let's assume that the BAT begins as follows:

BAT [0] = 2, BAT [1] = 5, BAT [2] = 3, BAT [3] = 4, BAT [4] = 6, BAT [5] = -1, BAT [6] = 7, BAT [7] = -2

Document which is Microsoft Word file 2003.The proposed system embeds steganography string in unused block of Microsoft Compound Document Binary File Format (MCDFF). It is provides two processes: Cover Generation and Embedding process as shown in the block diagram-figure(7). Now, BAT if have a file whose Property Table entry says it begins with index 0, walk the BAT array and

see that the file consists of blocks 0 (because BAT[0] is 2), 3(BAT[2] is 3), 4(BAT[3] is 4), 6 (BAT[4] is 6), and 7(BAT[6] is 7). It ends at block 7 because BAT [7] is -2, which is the end of chain marker. Similarly, a file beginning at index 1 consists of blocks 1 and 5 and block 5 refers to unused block. Other special number in a BAT array is:

- **-3, which indicate a "special" block, such as a block used to make up the Small Block Array, the Property Table, the main BAT, or the SBAT [12].**

In the physical structure of an MCDFF file, each Block is numbered with an index number under a header. Figure (6)

shows the process of accessing "Sample *A* Stream". The first index number for "Sample *A* Stream" is included in its Directory entry. It accesses the BAT to find the index number of the other Blocks that "Sample *A* Stream" uses – in this Example, if the first index number is 1st in Directory Entry, "Sample *A* Stream" consist of three Blocks as 1st, 4th and 5th from BAT [6].

7.5 Sector Allocation Table (SAT)

The Sector Allocation Table (SAT) is an array of SecIDs. It contains the SecID chain of all user streams. The size of the SAT (number of SecIDs) is equal to the number of existing sectors in the compound document file [3].

8. Proposed hiding system in Microsoft Compound Document file format (MCDFF)

The proposed system is an implementation of text steganography methods. This system will be used for embedding a steganograhy string into a document which is Microsoft Word file 2003 . the proposed system embedded steganography Compound Document Binary File Format (MCDFF). It is provides two processes . Cover

Generation and Embedding process as shown in the block diagram – figure(7).

8.1 Cover Generation Process

Cover Generation process make data embedding is disguised to be the product of a collaborative document authoring effort. That is, the stegodocument is made to appear to be the work of multiple authors. To facilitate communication of the authors during the collaborative document authoring process, the word processor records the exact modifications by an author and embeds the ways of revision as change tracking information into the document. From such change tracking information, can discern the exact changes made by a prior author, and can recover a prior version of the document if necessary (*see section 4.1 Annotation and collaboration Microsoft Word Document Tools).* Where an author is modifying a document and the word processor has tracked the author's modifications. Each collaborating author can accept or reject individual or all modifications made by another author.

It is a common practice for a collaborating author to review and then accept or reject each modification in a document first before performing his or her own corrections. Once upon a time; Microsoft invented "Track Changes". "Authors" put "changes" into their documents. More recently, "Reviewers" make "revisions" to their documents and "revisions" are one kind of "markup". The basic idea of the proposed system is to degenerate the contents of a cover document *D* to arrive at another document *D'* and embedding a secret message *M* in *D'* during the Embedding process, as shown in Fig. (8). the degeneration introduces errors into the degenerated document *D'* such that the degenerated document appears to be a preliminary work by a virtual author A', which is to be revised later by another author. A binary secret

message *M* **is embedded inside a cover document** *D'* **to obtain a stegodocument** *S***. Chosen Microsoft Word documents as cover media, which provide change tracking facilities to materialize the proposed method. Communications via Word documents are commonplace for personal, business, or academic purposes these days, used more in Middle East so transmissions of such documents will not be under close scrutiny. Most of the works cited in the introduction use the technique of modifying a cover medium to embed information.**

This type of data hiding generally assumes that the cover medium used is unknown to an adversary, or otherwise, the discrepancies between the cover medium and the corresponding stegomedium will arouse suspicion. On the other hand, the proposed method provides legitimate cases in using a known cover document. For example, an already published document that is collaboratively authored can be used as a cover document .

The stegodocument *S* **appears to be the version of the paper before change tracking information removal and submission for publication. The transmission of** *S* **by one of the collaborating authors to another author, a colleague, or a supervised student of the author is reasonable.**

A colleague or a student receiving the document containing the change tracking information can learn of the mistakes made by a colleague and the appropriate corrections to be made thereof .

8.2 Embedding, Extracting Process

This method of hiding data in MCDFF is to hide information in unused space, unused space occurs as unused block as follows.

Algorithm-1: Hiding Data

Input: Document of Microsoft Compound Document File Format (MCDFF**)**

Output: Stegodocument

Steps:

Step1: Open MCDFF file.

Step2: Read secret message from user**.**

Step3: Encode secret message with Huffman Coding**.**

Step4: Search for unused block in MCDFF file.

Step 5: insert secret message into unused block of MCDFF file.

Step 6: Save the content of document file format**.**

Step7: End.

Algorithm-2: Search Unused Block

Input: Document of Microsoft Compound Document Binary File Format (MCDFF).

Output: Unused Block Location.

Steps:

Step1: Loading Compound Document header of MCDFF file.

Step2: Extracting information and offset from header like (Microsoft signature, Block size, Block index of the first block of the property table (first Directory), byte ordering, Block Allocation Table (BAT) ID, minimum size of a stream).

Step3: Go to the first Directory (Root) Address.

Step4: Extract index of first block in file (starting Block).

Step5: Go to the Block Allocation Table (BAT) Address.

Step6: Loading Block Allocation Table (BAT).

Step7: Accessing from index of the first block in file to all other blocks.

Step8: if Block index = -1

- Calculate the Address of block index in file.
- Record the block as unused block.

Step 9: Else if (Not End of BAT) Go to step7.

Step 10: End.

Algorithm-3: Extracting hidden data
Input: Stegodocument
Output: Hidden data
Steps:
Step1: Open stegodocument
Step2: Search for unused block in stegodocument
Step3: Extract secret e from unused block of stegodocument.

Step4: Decode secret message.
Step5: End.

9. System Implementation

Implementation of proposed system is explained. The proposed system is built using Microsoft Visual C sharp .Net 2003 under Windows Xp as Operating System, Microsoft Word Document 2003, Office Automation Technique provided by Microsoft. In this section, the stages of the system will be discussed; these stages are shown in figure (9).

9.1 Document before Hiding

Having opened cover Document file, the Tracking Changes Tool will be used to modify document to be like collaborative writing between many authors. The cover document is shown in figure (10).When the button Document is clicked before hiding, it will open the Window in figure (11).The secret message will be:

TWENTY IN COUNTER AT TEN PM FROM CELING OUR TARGET DIAMOND THERE ARE ELEVEN GUARDS OUT

Encoding the secret message with Huffman Coding:

0010 10000 000 0110 000 0011 0110 000 000 10011 000 11101 000 0101 11100 10101 0011 0110 10010 10001 0100 10101 0010 0010 11011 000 0101 0010 11001 0111 0101 10100 0100 10101 0101 0010 000 0110 0011 0010 0010 000 0101 11000 10110 10111 0110 0100 10110 10100 000 0111 10011 0111 0101 11100 0100 10101 0110 0010 0011 0110 11100 000 0010 10010 0111 0011 10110 0100 0101 10010

When the embedding process button is clicked, it will open the:After writing the secret message, the Embed button must be pressed to hide this message. Button Exit will close the current form.

9.2 Document after Hiding:

This button will open the document after hiding a secret message the window as shown in figure (13). That message shows that this Document contains Tracking Change information and will ask if you want to continue saving this information.

9.3 Extracting Process

This stage describes the implementation of extracting method in the following steps:

First step: is to read the compound header as in extracting process.

Second step: finding starting block of a file.

Third step: Loading BAT array to accessing the Unused Block and extract the secret message from it.

Fourth step: Decoding the secret message with Huffman Coding.

When click Extraction Process button will open: to extract the hidden data press button Extract. Button Exit will close the current form.

10. Conclusions

The proposed system provides new method for embedding text in text, a number of conclusions were derived from this study:-

1. **The Cover Generation process in hiding system will increase the security of hidden system and avoiding drawing suspensions that there is hidden data**
2. **Hidden data in document will not affected by copying / mailing the stegodocument.**
3. **The proposed system hides English text in another text and gives good results.**
4. **This method of hiding data in MCDFF is only a few of many ways to hide or encrypt data.**

The difference between original Cover-Text size and Cover –Text size after embedding process is acceptable, for example in my Cover original cover-Text size is (34 KB),Cover –Text size after embedding is (34.5 KB) informed about size of empty Document is 10KB/ 11 KB and hidden message size63bytes.

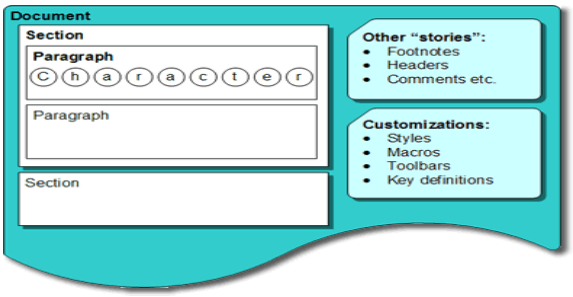

Figure (1) External Structure of a Word Document [10]

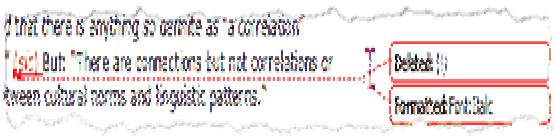

Figure (2) Track change example [10]

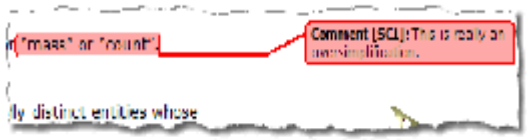

Figure (3) comments example [10]

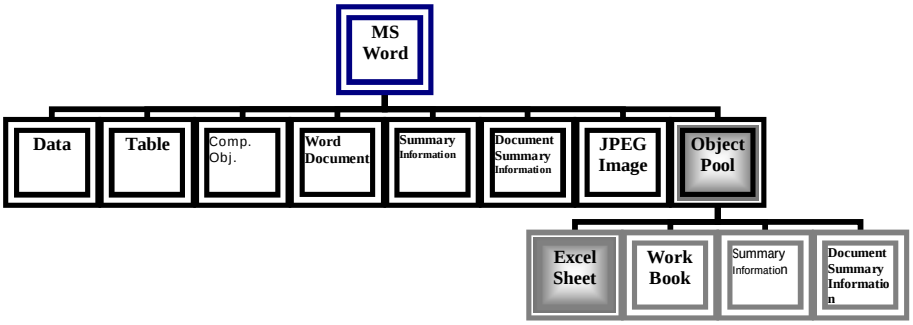

Figure (4) Sample Word document storage format [7]

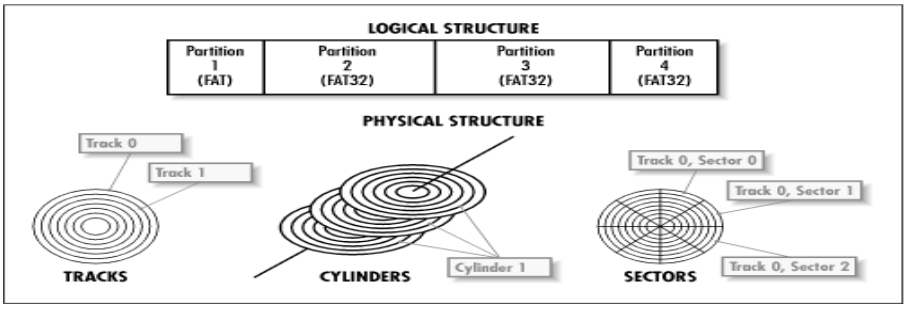

Figure (5) the structure of a hard disk

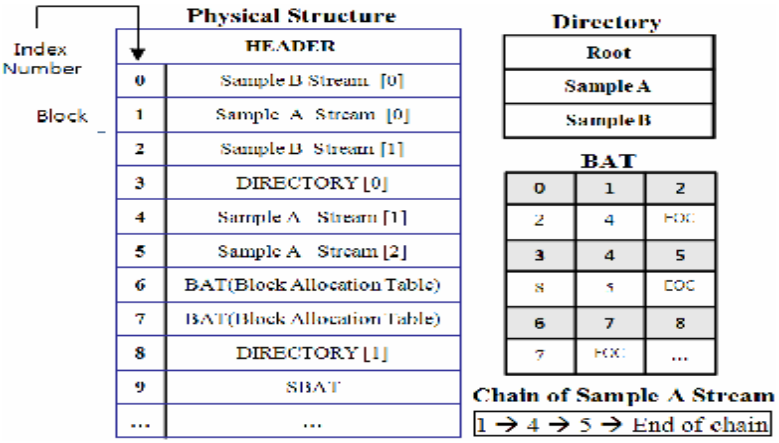

Figure (6) MS Compound files structure [6]

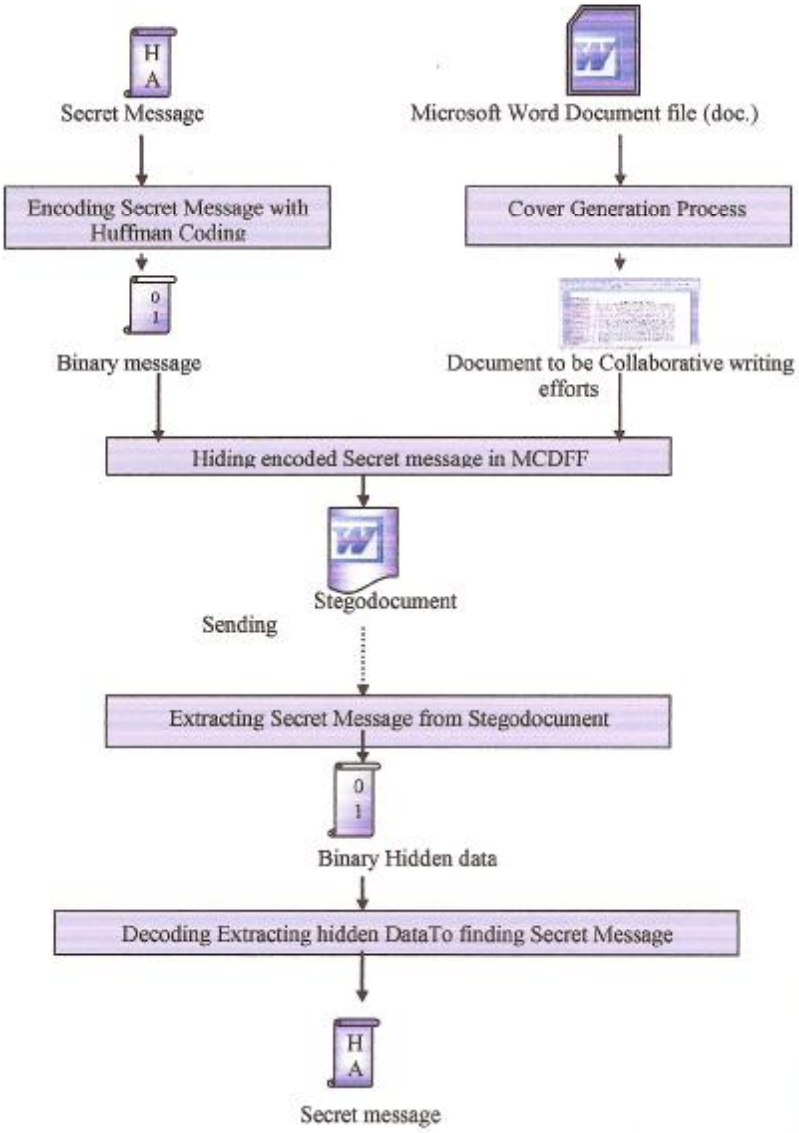

Figure (7) Block Diagram for Proposed System

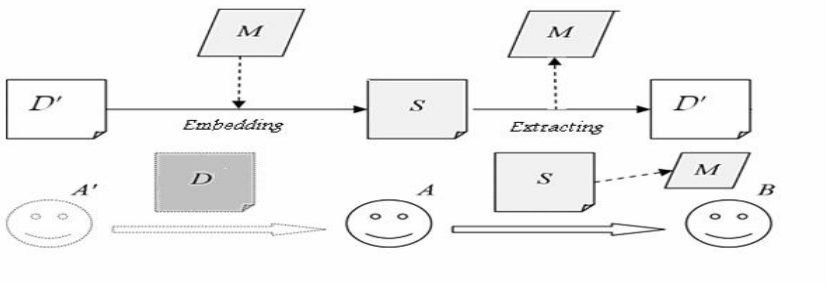

Figure (8) Author A sends a stegodocument S with an embedded message M to a recipient B after embedding M into a cover document D' to form S that appears to be the collaborative product of multiple authors A and A'.

Figure (9) the main menu for the Proposed system

Figure (10) Cover Document

Figure (11) Cover Document after Track changes

Figure (12) Embedding Process window

Figure (13) Document after Hiding

Figure (14) Extracting Process Window

Table (1) compound document header structure [3]

offset	Size	Contents
0	8	Compound document file identifier: D0 CF 11 E0 A1 B11AE1
8	16	Unique identifier (UID) of this file
24	2	Revision number of the file format (most used is 003E)
26	2	Version number of the file format (most used is 0003)
28	2	Byte order identifier FEH FFH = Little-Endian FFH FEH = Big-Endian
30	2	Size of a sector in the compound document file in power-of-two (ssz), real sector size is sec_size = 2ssz bytes (minimum value is 7 which means 128 bytes, most used value is 9 which means 512 bytes)
32	2	Size of a short-sector in the short-stream container stream in power-of-two (sssz),) real short-sector size is short_sec_size = 2sssz bytes (maximum value is sector size ssz, see above, most used value is 6 which means 64 bytes)
34	10	Not used
44	4	Total number of sectors used for the sector allocation table
48	4	SecID of first sector of the directory stream
52	4	Not used
56	4	Minimum size of a standard stream (in bytes, minimum allowed and most used size is 4096 bytes), streams with an actual size smaller than (and *not* equal to) this value are stored as short-streams
60	4	SecID of first sector of the short-sector allocation table or -2 (End Of Chain SecID) if not extant
64	4	Total number of sectors used for the short-sector allocation table
68	4	SecID of first sector of the master sector allocation table or -2 (End Of Chain SecID) if no additional sectors used
72	4	Total number of sectors used for the master sector allocation table
76	436	First part of the master sector allocation table containing 109 SecIDs

Table (2) Property -- 128 (0x80) byte block [12

Field	Description	Offset	Length	Default value or const
NAME	A unicode null-terminated uncompressed 16bit string (lose the high bytes) containing the name of the property.	0x00, 0x02, 0x04, ... 0x3E	Short[]	0x0000 for unused elements, field required, 32 (0x40) element max
NAME_SIZE	Number of characters in the NAME field	0x40	Short	Required
PROPERTY_TYPE	Property type (directory, file, or root)	0x42	Byte	1 (directory), 2 (file), or 5 (root entry)
NODE_COLOR	Node color	0x43	Byte	0 (red) or 1 (black)
PREVIOUS_PROP	Previous property index	0x44	Integer	-1
NEXT_PROP	Next property index	0x48	Integer	-1
CHILD_PROP	First child property index	0x4c	Integer	-1
SECONDS_1	Seconds component of the created timestamp?	0x64	Integer	0
DAYS_1	Days component of the created timestamp?	0x68	Integer	0
SECONDS_2	Seconds component of the modified timestamp?	0x6C	Integer	0
DAYS_2	Days component of the modified timestamp?	0x70	Integer	0
START_BLOCK	Starting block of the file, used as the first block in the file and the pointer to the next block from the BAT	0x74	Integer	Required
SIZE	Actual size of the file this property points to. (Used to truncate the blocks to the real size).	0x78	Integer	0

Table (3) Block Allocation Table Block [12]

Field	Description	Offset	Length	Default value or const
BAT_ELEMENT	Any given element in the BAT block	0x0000, 0x0004, 0x0008, ... 0x01FC	Integer	-1 = unused -2 = end of chain -3 = special (e.g., BAT block) All other values point to the next element in the chain and the next index of a block composing the file.

References

[1] **R., A., Al-Shamkhy,"** *Hiding Text in Text Using Dictionary Method"* **Msc. Thesis, Department of Computer Science and Information System, Baghdad, 2001.**

[2] **M., Chand, "***Structure Storage: A COM way to read/write persistent data",* **http://www.dotnetheaven.com/Uploadfile/mahesh/_com10425200508125 0AM/_com1.aspx?ArticleID=307eca4f-723b-4ed5-b823-2a05e71ai402, June 26, 2000.**

[3] **R., Daniel, "***OpenOffice.org's Documentation of the Microsoft Compound Document ",* **OpenOffice.org, the Speardsheet Project, June 2007.**

[4] **Dialogika, Makz, Math, Wk and Divo,** *"How to Retrieve Text from a Binary .doc File"***, March 2008.**

[5] **M.,Folk, J., Zoellick, and B., Riccardi, ,** *"File Structures an Object-Oriented Approach with C+ +",* **ADDISON-WESLEY, 1998.**

[6] **K. Hyukdon, K.Yeog, and L. Sangjin, "***A Tool for Detection of Hidden Data in Microsoft Compound Document File Format"* **, 2008 International Conference on Information Science and Security, 2008 IEEE , 2008.**

[7] **K., Jithra ,***"Microsoft Office Security, Part one",* **http://www.securityfocus.com/infocus/1874, 2006-08-22.**

[8] **F. Johnson, and S. Jajodia,** *"Steganalysis: The Investigation of Hidden Information,"* **in Proc. IEEE Information Technology Conf., Syracuse, NY, Sep.1998, pp.113-116.**

[9] **S. Katzenbeisser, and F. Peticolas, "***Information Hiding Techniques for Steganography and Digital Watermarking***", Artech House Inc, USA, 2000.**

[10] **T.-Y. Liu and W.-H. Tsai, "***A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique* **", IEEE Transactions on Information Forensics And Security, Vol. 2, No. 1, March 2007.**

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

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

[13] **R. Villan, , S. Voloshynovskiy, , O. Koval, , J. Vila, , E. Topak, , F. Deguillaume , Y. Rytsar, , and T. Pun, ,** *"Text Data-Hiding for Digital and Printed Documents: Theoretical and Practical Considerations"***, Computer Vision and Multimedia Laboratory – University of Geneva, 2006.**

[14] **Wikipedia, the free encyclopedia,** *"File Format",* **wikipedia®, 29 March 2009.**

[15] **Shaunakelly,** *"How does Track Changes in Microsoft Word Work?"***,Melbourne, Astralia, Sep., 2008.**

[16] **Linguistics,** *"Structure of a Word document",* **linguistics education, 2004.**

طريقة مقترحة لاخفاء البيانات في هيكلية وثائق المايكروسوفت ورد

أ.د.عبد المنعم صالح ابوطبيخ * د. هالة بهجت عبد الوهاب *
أماني يوسف البغدادي *

المستخلص

هذا البحث يقدم طريقة لأخفاء البيانات بالاستفادة من الخصائص الفيزيائية لنظام الحاسوب وكيفية خزنه ومعالجته لملف (.doc).الطريقة تستخدم الجزء الغير مستخدم (الفارغ) في الهيكلية المعقدة لملف مايكروسوفت ورد لاخفاء البيانات كذلك يتم الاستفادة من الامكانيات التي يقدمها برنامج المايكروسوفت ورد مثل أدواته لتوليد الغطاء للأخفاء. النظام المقترح يخفي نص بهيكلية (الصيغة الثنائية للملف) لنص اخرمطبوع بوثيقة المايكروسوفت ورد باستخدام مرحلتين : مرحلة *توليد الغطاء* و مرحلة *عملية التضمين*.
***توليد الغطاء*: الغطاء هو وثيقة من وثائق برنامج مايكوسوفت ورد اصدار 2003 سيظهر ليبدو كانه انتاج جهود كتابة تعاونية بين عدة مؤلفين باستخدام اداة تعقب التغيير.**
***عملية التضمين*: هذه العملية يتم فيها اخفاء سلسلة نصية بالكتلة الغير مستخدمة (الفارغة) بالهيكلية الثنائية لذلك الملف .**
التقنية المقترحة اعطت نتائج جيدة, كماوفرت مرونة في اخفاء رسالة حجمها 63 بايت بغطاء حجمه 34 كيلوبايت مع الاخذ بنظر الاعتبار ان حجم ملف .doc فارغا= 10 او 11 كيلوبايت فضلا عن استخدام اداة تعقب التغير لم يؤثر على البيانات المخفية و ان ارسال او نسخ الغطاء مع الرسالة السرية لم يؤثر على البيانات المخفية.

* الجامعة التكنولوجية / قسم علوم الحاسبات