New Approach of Hidden Data in the portable Executable File without Change the Size of Carrier File Using Statistical Technique

A.W. Naji*, Teddy S. Gunawan, A.A.Zaidan, Othman O. Khalifa, B.B.Zaidan, Wajdi F. Al-Khateeb, Shihab A. Hameed**
Electrical and Computer Engineering Department, Faculty of Engineering,
International Islamic University Malaysia, 53100 Gombak, Kuala Lumpur, Malaysia.

Summary
The rapid development of multimedia and internet allows for wide distribution of digital media data. It becomes much easier to edit, modify and duplicate digital information. In additional, digital document is also easy to copy and distribute, therefore it may face many threats. It became necessary to find an appropriate protection due to the significance, accuracy and sensitivity of the information. The strength of the hiding science is due to the non-existence of standard algorithms to be used in hiding secret messages. Also there is randomness in hiding methods such as combining several media (covers) with different methods to pass a secret message. Furthermore, there is no formal method to be followed to discover a hidden data. In this paper, a new information hiding system is presented. The aim of the proposed system is to hide information (data file) in an execution file (EXE) without change the size of execution file. The new proposed system is able to embed information in an execution file without change the size of execution file. Meanwhile, since the cover file might be used to identify hiding information, the proposed system considers overcoming this dilemma by using the execution file as a cover file.

Key words:
Information Hiding, Steganography, Statistical Technique, portable executable file.

1. INFORMATION HIDING

Information hiding is a general term encompassing many sub disciplines, is a term around a wide range of problems beyond that of embedding message in content. The term hiding here can refer to either making the information undetectable or keeping the existence of the information secret. Information hiding is a technique of hiding secret using redundant cover data such as images, audios, movies, documents, etc. This technique has recently become important in a number of application areas. For example, digital video, audio, and images are increasingly embedded with imperceptible marks, which may contain hidden signatures or watermarks that help to prevent unauthorized copy [1].It is a performance that inserts secret messages into a cover file, so that the existence of the messages is not apparent. Research in information hiding has tremendous increased during the past decade with commercial interests driving the field [1].

2. STEGANOGRAPHY

Steganography is the art and science of writing hidden messages in such a way that no-one can realizes there is a hidden message in data (e.g. images file, documents file, sounds file … etc) except the sender and intended recipient. The word steganography is of Greek origin and means "covered, or hidden writing" .Cryptography obscures the meaning of a message, but it does not conceal the fact that there is a message, cryptography is the art of secret writing, which is intended to make a message unreadable by a third party but does not hide the message in a communication [1]. Although steganography is separate and distinct from cryptography, but there are many analogies between the two, and some authors categorize steganography as a form of cryptography since hidden communication is a form of secret writing [2]. A Watermark is a recognizable image or pattern in paper that appears as various shades of lightness/darkness when viewed by transmitted light (or when viewed by reflected light, atop a dark background), caused by thickness variations in the paper ,Digital Watermarking is the process of embedding information into a digital signal. The signal may be audio, pictures or video, for example. If the signal is copied, then the information is also carried in the copy. Steganography and digital watermarking are the same but the purpose of last one is use to copyright protection systems, which are intended to prevent or deter unauthorized copying of digital media. Steganography is an application of digital watermarking, where two parties communicate a secret message embedded in the digital signal [2]. Classical Steganography often used methods of completely obscuring the message so it was unnoticeable to those who didn’t know the specific covert method it was using for example Invisible Inks, which was able to write a confidential letter with any other non-value-confidential and usually write between lines. Modern Steganography refers to hide information in digital picture, audio or text files …etc , each one of this digitals

Manuscript received July 5, 2009
Manuscript revised July 20, 2009

data has a many techniques can use with it , for example in digital image the JPHide/JPSeek uses the coefficients in a JPEG to hide information, this method alter the image. In digital audio file several packages also exist for hiding data in audio files, Such as MP3Stego not only effectively hides arbitrary information, but also claims to be a partly robust method of watermarking MP3 audio files [4]. The Windows Wave format lets users hide data using Steghide, it alters the least significant bits (LSB) of data in the carrier medium [3].In a growing number of applications like digital rights management, covert communications, hiding executables for access control, annotation etc .all these application scenarios given the multimedia steganography techniques have to satisfy two basic requirements. The first requirement is perceptual transparency or noticeable perceptual distortion, i.e. cover object (object not containing any additional data) and stego object (object containing secret message) must be perceptually indiscernible [3]. The second constraint is high data rate of the embedded data. All the stego-applications, besides requiring a high bit rate of the embedded data, have need of algorithms that detect and decode hidden bits without access to the original multimedia sequence (blind detection algorithm) [4].

3. STATISTICAL TECHNIQUE.

Statistical Steganography techniques utilize the existence of "1-bits" Steganography schemes, which embed one bit of information in a digital carrier. This is done by modifying the cover in such a way that some statistical characteristics change significantly if a "1" is transmitted. Otherwise, the cover is left UN changed. So the receiver must be able to distinguish unmodified covers from modified ones. A cover is divided into l (m) disjoint blocks B1...B l (m). A secret bit, mi is inserted into the ith block by placing "1" in to Bi if mi=1.Otherwise, the block is not changed in the embedding process [5].

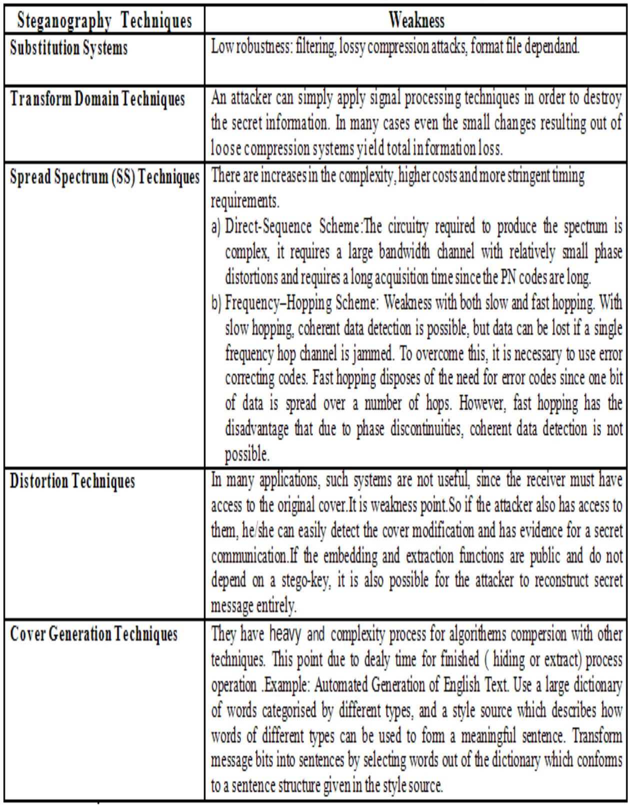
From the above table, most of the techniques are very complex and not suitable to be used with exe.file. In order to use exe.file,. Thus, we choose to apply statistical technique because it is not complex and suitable to be implemented with the structure and characteristic of the exe.file.

4. PORTABLE EXECUTABLE FILE (PE-FILE)

The planned system uses a portable executable file as a cover to embed an executable program as an example for the planned system.This section is divided into three parts [6],[7]:

- Executable files types.
- Techniques related with PE-file.
- PE File Layout.

Table 1: Weakness of Steganography Techniques



Steganography Techniques	Weakness
Substitution Systems	Low robustness: filtering, lossy compression attacks, format file dependand.
Transform Domain Techniques	An attacker can simply apply signal processing techniques in order to destroy the secret information. In many cases even the small changes resulting out of loose compression systems yield total information loss.
Spread Spectrum (SS) Techniques	There are increases in the complexity, higher costs and more stringent timing requirements. a) Direct-Sequence Scheme:The circuitry required to produce the spectrum is complex, it requires a large bandwidth channel with relatively small phase distortions and requires a long acquisition time since the PN codes are long. b) Frequency–Hopping Scheme: Weakness with both slow and fast hopping. With slow hopping, coherent data detection is possible, but data can be lost if a single frequency hop channel is jammed. To overcome this, it is necessary to use error correcting codes. Fast hopping disposes of the need for error codes since one bit of data is spread over a number of hops. However, fast hopping has the disadvantage that due to phase discontinuities, coherent data detection is not possible.
Distortion Techniques	In many applications, such systems are not useful, since the receiver must have access to the original cover.It is weakness point.So if the attacker also has access to them, he/she can easily detect the cover modification and has evidence for a secret communication.If the embedding and extraction functions are public and do not depend on a stego-key, it is also possible for the attacker to reconstruct secret message entirely.
Cover Generation Techniques	They have heavy and complexity process for algorithems compersion with other techniques. This point due to dealy time for finished (hiding or extract) process operation .Example: Automated Generation of English Text. Use a large dictionary of words categorised by different types, and a style source which describes how words of different types can be used to form a meaningful sentence. Transform message bits into sentences by selecting words out of the dictionary which conforms to a sentence structure given in the style source.

A. *Executable File Types*

The number of different executable file types is as many and varied as the number of different image and sound file formats. Every operating system seems to have several executable file types unique to it. These types are (Mega, 2004):

- EXE (DOS"MZ")

DOS-MZ was introduced with MS-DOS (not DOS v1 though) as a companion to the simplified DOS COM file format. DOS-MZ was designed to be run in real mode and having a relocation table of SEGMENT: OFFSET pairing. A very simple format that can be run at any offset, it does not distinguish between TEXT, DATA and BSS.The maximum file size of (code + data + bss) is one-mega bytes in size. Operating systems that use are: DOS, Win*, Linux DOS.

- EXE (win 3.xx "NE"):

The WIN-NE executable formatted designed for windows 3.x is the "NE" new-executable. Again, a 16-bit format, it alleviates the maximum size restrictions that the DOZ-MZ has. Operating system that uses it is: windows 3.xx.

- EXE (OS/2 "LE"):

The "LE" linear executable format was designed for IBM's OS/2 operating system by Microsoft supporting both 16 and 32-bit segments operating systems that are used in: OS/2, DOS.

- EXE (win 9x/NT "PE"):

With windows 95/NT a new executable file type is required, thus was born the "PE" portable executable. Unlike its predecessors, the WIN-PE is a true 32-bit file format, supporting releasable code. It does distinguish between TEXT, DATA, and BSS. It is in fact, a bastardized version of the common object file format (COFF) format. Operating systems that use it are: windows 95/98/NT/2000/ME/CE/XP.

- ELF:

The ELF, Executable Linkable Format was designed by SUN for use in their UNIX clone. A very versatile file format, it was later picked up by many other operating systems for use as both executable files and as shared library files. It does distinguish between TEXT, DATA and BSS.
TEXT: the actual executable code area.
DATA: "initialized" data, (Global Variables).
BSS : "un- initialized" data, (Local Variables).

B. Techniques Related with PE

Before looking inside the PE file, we should know special techniques some of which are [6]:

- General view of PE files sections

A PE file section represents code or data of some sort. While code is just code, there are multiple types of data. Besides read/write program data (such as global variables), other types of data in sections include application program interface (API) import and export tables, resources, and relocations.

Each section has its own set of in-memory attributes, including whether the section contains code, whether it's read-only or read/write, and whether the data in the section is shared between all processes using the executable file. Sections have two alignment values, one within the desk file and the other in memory. The PE file header specifies both of these values, which can differ. Each section starts at an offset that's some multiple of the alignment value. For instance, in the PE file, a typical alignment would be 0x200. Thus, every section begins at a file offset that's a multiple of 0x200.Once mapped into memory, sections always start on at least a page boundary. That is, when a PE section is mapped into memory, the first byte of each section corresponds to a memory page. On x86 CPUs, pages are 4KB aligned, while on the Intel Architecture IA-64, they're 8KB aligned.

- Relative Virtual Addresses (RVA)

In an executable file, there are many places where an in-memory address needs to be specified. For instance, the address of a global variable is needed when referencing it. PE files can load just about anywhere in the process address space. While they do have a preferred load address, you can't rely on the executable file actually loading there. For this reason, it's important to have some way of specifying addresses that are independent of where the executable file loads. To avoid having hard coded memory addresses in PE files, RVAs are used. An RVA is simply an offset in memory, relative to where the PE file was loaded. For instance, consider an .EXE file loaded at address 0x400000, with its code section at address 0x401000. The RVA of the code section would be:

(Target address) 0x401000 – (load address) 0x400000 = (RAV) (1)

To convert an RVA to an actual address, simply reverse the process: add the RVA to the actual load address to find the actual memory address. Incidentally, the actual memory address is called a Virtual Address (VA) in PE parlance. Another way to think of a VA is that it's an RVA with the preferred load address added in.

- Importing Functions

When we use code or data from another DLL, we're importing it. When any PE files loads, one of the jobs of the windows loader is to locate all the imported functions and data and make those addressees available to the file being loaded.

C. PE File Layout

There are two unused spaces in PE file layout [8], and these unused spaces are suggested to hide a watermark. The size of the second unused space is different from one file to another .The most important reason behind the idea of this system is that the programmers always need to create a back door for all of their developed applications, as a solution to many problems such that forgetting the password. This idea leads the customers to feel that all programmers have the ability to hack their system any time. At the end of this discussion all customers always are used to employ trusted programmers to build their own application. Programmers want their application to be safe any where without the need to build ethic relations with their customers. In this system a solution is suggested for this problem [8].The solution is to hide the password in the executable file of the same system and then other application to be retracted by the customer himself. Steganography needs to know all files format to find a way for hiding information in those files. This technique is difficult because there are always large numbers of the file format and some of them have no way to hide information in them.

5. CHARACTERISTICS OF EXECUTABLE FILES

The characteristics of the Executable file does not have a standard size, like other files, for example the image file (BMP) the size of this file is between (2-10 MB), Other example is the text file (TXT) the size often is less than 2 MB.Through our study the characteristics of files have been used as a cover, it found that lacks sufficient size to serve as a cover for information to be hidden. For these features of the Executable file, it has unspecified size; it can be 650 MB like window setup File or 12 MB such as installation file of multi-media players. Taking advantage of this feature (disparity size) make it a suitable environment for concealing information without detect the file from attacker and discover hidden information in this file.

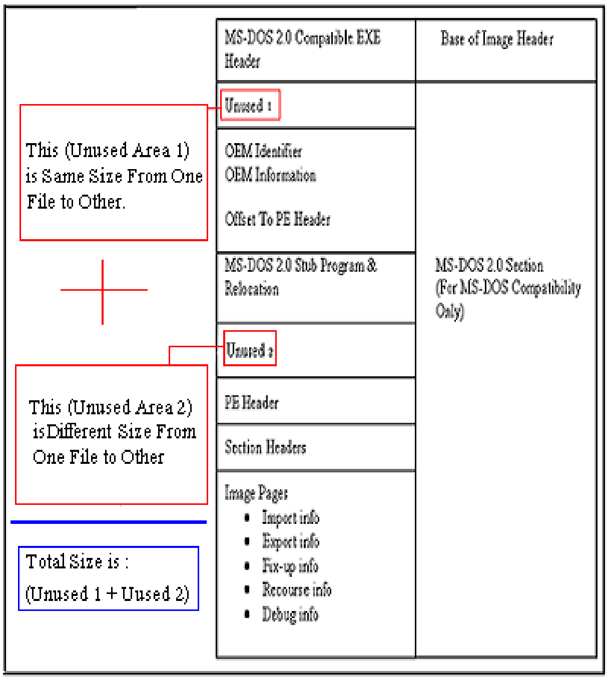

MS-DOS 2.0 Compatible EXE Header
Base of Image Header
Unused 1
This (Unused Area 1) is Same Size From One File to Other.
OEM Identifier
OEM Information
Offset To PE Header
MS-DOS 2.0 Stub Program & Relocation
MS-DOS 2.0 Section (For MS-DOS Compatibility Only)
Unused 2
This (Unused Area 2) is Different Size From One File to Other
PE Header
Section Headers
Image Pages
Import info
Export info
Fix-up info
Recourse info
Debug info
Total Size is :
(Unused 1 + Used 2)

Figure 1.Typical 32-bit Portable .EXE File Layout.

6. ADVANTAGE OF PE-FILE

The addition of the Microsoft® windows NT™ operating system to the family of windows™ operating systems brought many changes to the development environment and more than a few changes to applications themselves. One of the more significant changes is the introduction of the Portable Executable (PE) file format. The name "Portable Executable" refers to the fact that the format is not architecture specific [6]. In other words, the term "Portable Executable" was chosen because the intent was to have a common file format for all flavors of windows, on all supported CPUs [6].The PE files formats drawn primarily from the Common Object File Format (COFF) specification that is common to UNIX® operating systems. Yet, to remain compatible with previous versions of the MS-DOS® and windows operating systems, the PE file format also retains the old familiar MZ header from MS-DOS [7].The PE file format for Windows NT introduced a completely new structure to developers familiar with the windows and MS-DOS environments. Yet developers familiar with the UNIX environment will find that the PE file format is similar to, if not based on, the COFF specification .The entire format consists of an MS-DOS MZ header, followed by a real-mode stub program, the PE file signature, the PE file header, the PE optional header, all of the section headers, and finally, all of the section bodies [7].

7. METHODOLOGY

A. System Overview

The most important reason behind the idea of this system is that the programmers always need to create a back door for all of their developed applications, as a solution to many problems such that forgetting the password. This idea leads the customers to feel that all programmers have the ability to hack their system any time. At the end of this discussion all customers always are used to employ trusted programmers to build their own application. Programmers want their application to be safe anywhere without the need to build ethic relations with their customers. In this system a solution is suggested for this problem. The solution is to hide the password in the executable file of the same system and then other application to be retracted by the customer himself. Steganography needs to know all files format to find a way for hiding information in those files. This technique is difficult because there are always large numbers of the file format and some of them have no way to hide information in them.

B. System Concept

Concept of this system can be summarized as hiding the password or any information beyond the end of an executable file so there is no function or routine (open-file, read, write, and close-file) in the operating system to extract it. This operation can be performed in two alternative methods:

- Building the file handling procedure independently of the operating system file handling routines. In this case we need canceling the existing file handling routines and developing a new function which can perform our need, with the same names. This way needs the customer to install the system application manually as shown in Figure 2.
- Developing the file handling functions depending on the existing file handling routines. This way can be performed remotely as shown in Figure 3.

The advantage of the first method is it doesn't need any additional functions, which can be identified by the analysts. The disadvantage of this method is it needs to be installed (can not be operated remotely). The advantage of the second method is it can be executed remotely and suitable for networks and the internet applications. So we choose this concept to implementation in this paper.

C. System Features

- There is no change on the exe.file size, where you can increase hide information size of exe.file dependent on size of (unused area 1 + unused area 2) with in executable file, because the unused area 2 in the exe.file is different size from one file to on other and unused area 1 in the exe.file is same size from one file to on other by useful from structure on the property of the unused area 2 with in exe.file. Other world disparity in the size of the unused area 2 with in executable files, so that the exe.file cannot identify the size of the unused area 2 with in exe.file, by computation between two unused areas, they can increase the size of information hiding within exe.file.

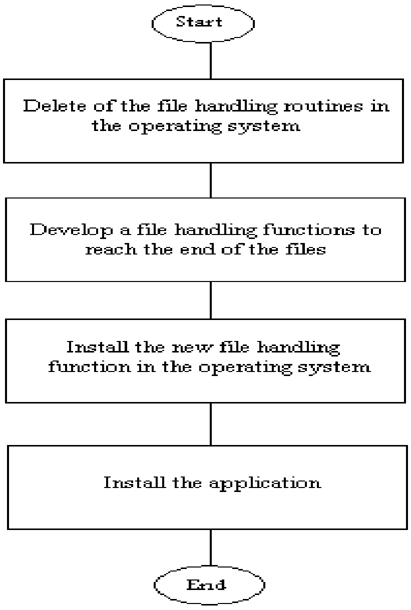

Start
Delete of the file handling routines in the operating system
Develop a file handling functions to reach the end of the files
Install the new file handling function in the operating system
Install the application
End

Figure 2. First Method of the System Concept

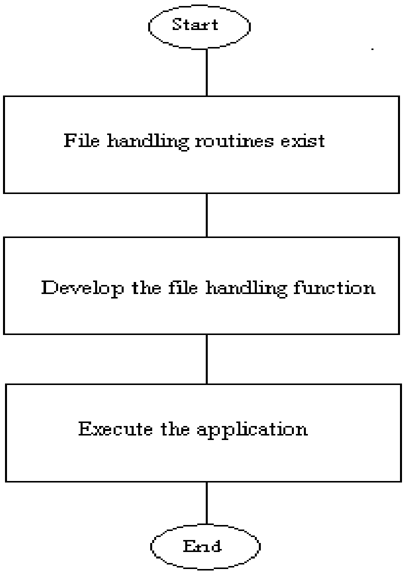

Start
File handling routines exist
Develop the file handling function
Execute the application
End

Figure 3. Second Method of the System Concept

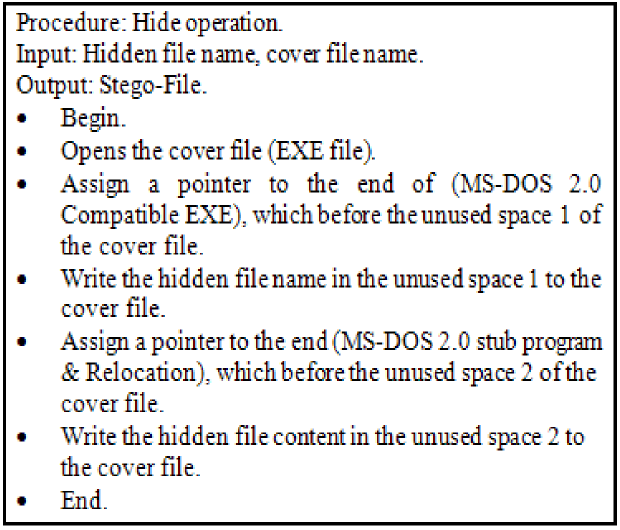
- This system has the following feature: The hiding operation of (unused area 1+ unused area 2 within EXE File) increases the degree of security of hiding technique which is used in the proposed system because the size of cover file doesn't change. So the attacker can not be attack the information hidden.

- The cover file can be executed normally after hiding operation. Because the hidden information already hide in the (unused area 1+ unused area 2) within exe.file and thus cannot be manipulated as the exe.file, therefore, the cover file still natural, working normally and not effected, such as if the cover is exe.Files (WINDOWES XP SETUP) after hiding operation it'll continued working, In other words, the exe.file can be installed of windows.

D. The Proposed System Structure

To protect the hidden information from retraction the system encrypts the information by the built-in encryption algorithm provided by the JAVA. The block flow of hiding operation can be performed as shown in Figure 5. The following algorithm is the hiding operation procedure:

The following algorithm is the hiding operation procedure:



Procedure: Hide operation.
Input: Hidden file name, cover file name.
Output: Stego-File.

- Begin.
- Opens the cover file (EXE file).
- Assign a pointer to the end of (MS-DOS 2.0 Compatible EXE), which before the unused space 1 of the cover file.
- Write the hidden file name in the unused space 1 to the cover file.
- Assign a pointer to the end (MS-DOS 2.0 stub program & Relocation), which before the unused space 2 of the cover file.
- Write the hidden file content in the unused space 2 to the cover file.
- End.

Figure 4. Shows Algorithm for Hiding Operation.

8. CONCLUSION

The .EXE file is one of the most important file in operating systems. Hiding information in such file is the basic goal for this paper, because most users can't alter or modify the content of any .EXE file. The following are the conclusions:

- One of the important conclusions in implementation of the proposed system is the solving of the problems that are related to the size of cover file when we used EXE File. Therefore, the proposed hiding method makes the relation between the cover and the message independent.
- The proposed hiding technique is flexible and very useful in hiding any type of data with maximum size (Unused area 1+ Unused area 2).
- PE files structure is very complex because they depend on multi headers and addressing and insertion of data to PE files without full understanding of their structure may damage them. Therefore, the optimum choice is to hide the information beyond the structure of these files.

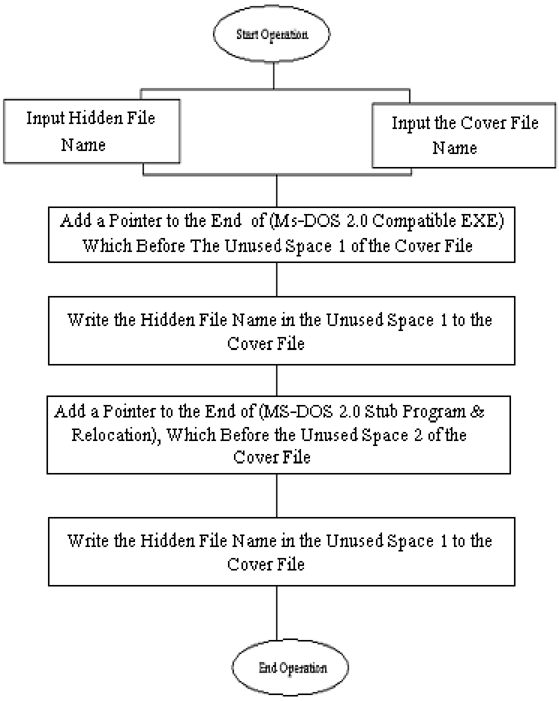

Start Operation
Input Hidden File Name
Input the Cover File Name
Add a Pointer to the End of (Ms-DOS 2.0 Compatible EXE) Which Before The Unused Space 1 of the Cover File
Write the Hidden File Name in the Unused Space 1 to the Cover File
Add a Pointer to the End of (MS-DOS 2.0 Stub Program & Relocation), Which Before the Unused Space 2 of the Cover File
Write the Hidden File Name in the Unused Space 1 to the Cover File
End Operation

Figure 5. Block Flow of Hiding Operation.

References

[1] A.W. Naji,Teddy S. Gunawan and Shihab A. Hameed, B.B Zaidan, A.A Zaidan " "Stego-Analysis Chain, Session One" Investigations on Steganography Weakness Vs Stego-Analysis System for Multimedia File ", International Conference on IACSIT Spring Conference (IACSIT-SC09) , Session 9, P.P 393-397, 2009, Singapore .

[2] A.W.Naji, Shihab A.Hameed, Md Rafiqul Islam, B. B. Zaidan,Teddy S. Gunawan, and A. A. Zaidan," "Stego-Analysis Chain, Session Two" Novel Approach of Stego-Analysis System for Image File ", International Conference on IACSIT Spring Conference (IACSIT-SC09) , Session 9, P.P 398-401 , 2009, Singapore .

[3] Mohamed Elsadig Eltahir, Laiha Mat Kiah, B.B.Zaidan and A.A.Zaidan," High Rate Video Streaming Steganography", International Conference on Information Management and Engineering (ICIME09), Session 10, P.P 550-553, 2009, Kuala Lumpur, Malaysia.

[4] Davern, P.S, M.G, "Steganography It History and Its Application to Computer Based Data Files", School of Computer Application (SCA), Dublin City University. Working Paper. Studies (WPS), Baghdad, Iraq, 2007.

[5] Fazida.Othman, Miss.Laiha.Maktom, A.Y.Taqa, B.B.Zaidan, A.A.Zaidan, "An Extensive Empirical Study for the Impact of Increasing Data Hidden on the Images Texture", International Conference on Future Computer and Communication (ICFCC 09), Session 7, P.P 477-481, 2009, Kuala Lumpur, Malaysia .

[6] Johnson, N. F. S. D, Z., "Information Hiding: Steganography and Watermarking-Attacks and Countermeasures", Center for Secure Information Systems (CSIS), Boston/Dordrecht/London, George Mason University, 2006.

[7] Katzenbeisser, S. P., A. P, "Information Hiding Techniques for Steganography and Digital watermarking", available from: Artech house pub, 2005.

[8] Katzenbeisser S. & Petitcolas, F. A., "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, USA, 2001.

Dr. Ahmed Wathik Naji - He obtained his 1st Class Master degree in Computer Engineering from University Putra Malaysia followed by PhD in Communication Engineering also from University Putra Malaysia. He supervised many postgraduate students and led many funded research projects with more than 50 international papers. He has more than 10 years of industrial and educational experience. He is currently Senior Assistant Professor, Department of Electrical and Computer Engineering, International Islamic University Malaya, Kuala Lumpur, Malaysia.

Aos Alaa Zaidan - He obtained his 1st Class Bachelor degree in Computer Engineering from university of Technology / Baghdad, followed by master in data communication and computer network from University of Malaya. He led or member for many funded research projects and He has published more than 40 papers, he has done many projects on Steganography. He is PhD candidate in the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.

Bilal Bahaa Zaidan - he obtained his Bachelor degree in Mathematics and Computer Application from Saddam University, Baghdad followed by master from the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, He led or member for many funded research projects and He has published more than 40 papers at various international and national conferences and journals. He is PhD candidate in the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.

Dr. Shihab A Hameed - He obtained his PhD in software Engineering from UKM. He has three decades of industrial and educational experience. His research interest is mainly in the software engineering, software quality, surveillance and monitoring systems, health care and medication. He supervised numerous funded projects and has published more than 60 papers at various international and national conferences and journals. He is currently Senior Assistant Professor, Department of Electrical and Computer Engineering, International Islamic University Malaysia. Malaya, Kuala Lumpur.

Dr. Teddy Surya Gunawan - He received his B.Eng degree in Electrical Engineering with cum laude award from Institut Teknologi Bandung (ITB), Indonesia in 1998. He obtained his M.Eng degree in 2001 from the School of Computer Engineering at Nanyang Technological University, Singapore, and PhD degree in 2007 from the School of Electrical Engineering and Telecommunications, The University of New South Wales, Australia. His research interests are in speech and image processing, biomedical processing, image processing, and parallel processing. He is currently Assistant Professor and Academic Advisor at Department of Electrical and Computer Engineering, and Senior Assistant Professor at Department of Electrical and Computer Engineering, International Islamic University Malaysia.

Othman O. Khalifa received his Bachelor’s degree in Electronic Engineering from the Garyounis University, Libya in 1986. He obtained his Master degree in Electronics Science Engineering and PhD in Digital Image Processing from Newcastle University, UK in 1996 and 2000 respectively. He worked in industrial for eight years and he is currently Professor and Head of the department of Electrical and Computer Engineering, International Islamic University Malaysia. His area of research interest is Communication Systems, Information theory and Coding, Digital image / video processing, coding and Compression, Wavelets, Fractal and Pattern Recognition. He published more than 130 papers in international journals and Conferences. He is SIEEE member, IEEE computer, Image processing and Communication Society member.

Dr. Wajdi Fawzi Al-Khateeb- He received his PhD from the International Islamic University, Malaysia and his MSc from the Technical University of Berlin, Germany. His research interest is mainly in the Reliability Engineering, Fault Tolerant Systems, QoS Networking, Microwave Radio Links. He is currently an assistant Professor in the department of Electrical and Computer Engineering, International Islamic University Malaysia.