IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 01 (January. 2014), ||V7|| PP 21-30

A Proposed System for Hiding Information In Portable Executable Files Based on Analyzing Import Section

Mohammad Hussein Jawwad
[1-] *University of Babylon- College of Information Technology- Iraq*

Abstract: - Information-hiding techniques have newly become very important in a many application areas. Many cover mediums have been used to hide information like image, video, voice. Use of these mediums have been extensively studied. In this paper, we propose new system for hiding information in Portable Executable files (PE files). PE is a file format for executable file used in the 32-bit and 64-bit versions of the Windows operating system. This algorithm has higher security than some traditional ones because of the combination between encryption and hiding. We first encrypt the information before hiding it to ensure high security. After that we hide the information in the import table of PE file. This hiding depends on analyzing the import table characteristics of PE files that have been built under Borland turbo assembler. The testing result shows that the result file does not make any inconsistency with anti-virus programs and the PE file still function as normal after the hiding process.

Keywords: - *PE File, Information Hiding, Cryptography.*

I. INTRODUCTION

The development and wide use of computer science and informatics also, the advent of global networks (i.e. Internet) that make the access for information is possible for every one online impose the need for ways to avoid hackers and muggers from getting the important information. one of these ways is the use of information hiding.

Information hiding (Steganography) technology mainly studies how to embed secret data into another cover media which can be transmitted publicly[1]. Since attackers don not know whether the carrier contains secret data, even if they know, it is difficult to extract or remove secret data. Thus, this technology has higher security than the traditional cryptology[2].

In general information hiding techniques tries to embed amount of information in a specific cover file such as image, music or video files such that these information become under the supervision of system managers and under high level of security. This leads to discover new techniques and cover media to hide information in it. One of these new cover media is the use of portable executable files.

A portable executable file (PE file) is the format for file in 32-bit and 64 bit operating systems and it is one of the most important file formats in the Internet. Research that based on PE files to hide information has high significance for the purpose of protecting copyright and ensure secret communication.

Todays, the field of information protection can be classified into : information hiding (Steganography), encryption of information (Cryptography) or a combination between them [3].

Cryptography is a way of storing and transmitting information in a method that only the intended people can understand. The purpose of it is to protect information by encoding it into a format that is unreadable.

This paper presents a way to hide information, after encrypted it to ensure high security, into the import table of PE file that have been built under Borland through making analyzing for the characteristics of import table.

II. INFORMATION HIDING

The use of secret communication is as old as communication itself. In this section we will discover the development of information hiding techniques specially Steganography. We will focus on it because it is an important sub discipline of information hiding.

One of the most important part of the security of the Internet and open systems is the use of Steganographic technologies. Steganography means “covered writing”. This term was derived from Greek language. Cryptography and steganography are both used to protect information from any type of attacks. Both cryptography and steganography are very good tools to accomplish this but neither one of them alone is useful because each one can be broken. For this reason there is a need to use both of them together to add more steps of security [4] .

A general structure for the steganography system is shown in figure (1). From this figure we discover that the sender want to transmit to the receiver through steganographic transmission. The input to the stegano-system is the message to be hidden. [5]

The algorithm may (or may not) use a key for the steganography process as an additional level of security. This key (or other related to it) is used to extract the information again [6].
To extract the information, the receiver reverses the process [7].

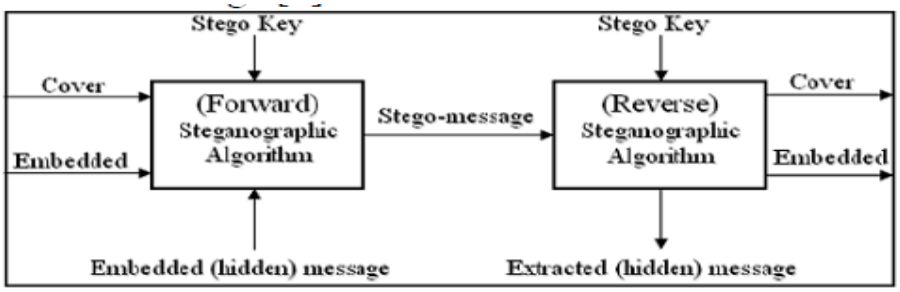


FIGURE (1) GENERAL STEGANOGRAPHY SYSTEM

Steganography works on writing hidden data in a cover file in a way that there is no one can expect there is a hidden data in a file except the sender of the file and the receiver.
Steganography include an array of secret communication methods that hide the message from being seen or discovered.
The goal of steganography is to avoid any distrust to the existence of a hidden data. The use of information hiding has recently become important in many field of application areas.
Digital audio, video, and pictures are increasingly furnished with distinguishing but imperceptible marks, which may contain a hidden copyright notice or serial number or even help to prevent unauthorized copying directly.

III. PORTABLE EXECUTABLE FILE (PE-FILE)

The proposed system uses a portable executable file (PE File) as a cover file to hide information. This section is divided into five parts:

Executable File Types
There are many variations of executable file formats as the variety of image and sound file formats. There are several executable file types that are unique to every operating system. These types cab be described as follow [8]

• EXE (DOS"MZ")
This type was introduced with MS-DOS (also can be run under Linux DOS). It was introduced to be run under real mode and it's relocation table consisting of a pair of SEGMENT:OFFSET. This type has a simple format and can be run at any offset. Also, it does not differentiate between DATA, TEXT and other segments. The maximum size for this type of exe file is one megabytes.

• EXE (win 3.xx "NE")
This type was designed for windows 3.x. A 16-bit file format lessen the restrictions of the maximum size that DOS-MZ format has.

• EXE (OS/2 "LE")
This type (linear executable) was designed by Microsoft to be work under IBM OS/2 operating system. It support both 16 and 32 bits segments.

• EXE (win 9x/NT "PE")
This type was introduced when windows 95/NT were emerged. It is a 32 bit file format that supporting relocatable code and can differentiate between DATA and TEXT. It is a derived version from the common object format (COFF).

• ELF
This type (executable linkable format) was designed by SUN to be used in their cop of UNIX. Also, it is differentiate between DATA and TEXT and it was later used b man other operating systems as shared library files and as a executable files.

IV. CHARACTERISTICS OF EXECUTABLE FILES

One of the most important characteristics of the executable files is they not have a standard size like other files. As an example, the size of images that have a (bmp) format is ranging between 2-10 MB . Also text files have size that is than 2 MB. By studying the characteristics of many types of files that have been used as a cover file we found that they don't have sufficient size to hide the information.

On the other hand, the executable file has unspecified size (may be 10 MB, 600 MB). This feature (variance size) make the executable files have an advantage concealing information over other types of files. Also this feature impose more difficulty on attackers to detect and discover the hidden information.

V. RVA

In an PE file, there are many positions where an in-memory address requires to be specified. As an example, the address of an variable is necessary when we need to referencing it. PE files can be loaded anywhere in the address space of the process. While PE files have a preferred load address, we can't depend on the PE file indeed loading there. For this purpose, it's necessary to have some method of specifying addresses that are independent of where the exe file loads. To avoid having hard coded memory addresses in executable files, relative virtual addresses are used[9].

RVA is the offset of some locations relative to the address where the file is memory-mapped. As example if the loader maps a PE file into main memory at the address 0x20000 in the virtual address space and a specific item starts at address 0x20516, then the item' RVA is 0x516 [10]

In general:

RVA= Virtual address - Base address

If we want to convert an relative virtual address to an actual address, simply we reverse the process: we add the RVA to the actual loaded address to find the actual memory address.

VI. PE FILE FORMAT

Before exploring PE file format we must know that the file on hard disk is too similar to its look after it loaded from disk to main memory. The windows loader uses memory-mapped file technique to map the file into the specified virtual address space. The use of memory mapping file technique enhance efficiency because the file stay resident on hard disk but it look like it is in memory [11]. Figure (2) shows PE file format:

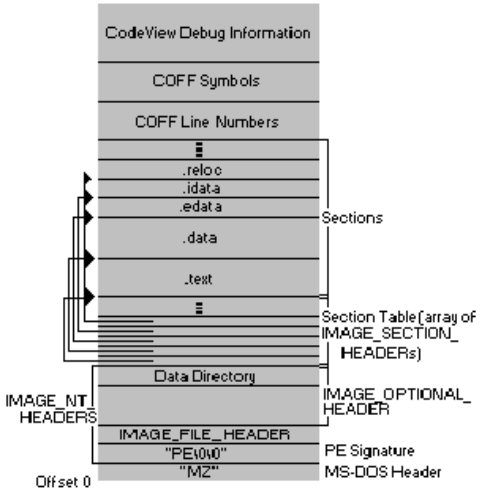


Figure (2) PE file format

The PE Header

The first part in the PE file is the header. A header looks like a number of fields at a specific locations. These fields contain information that tell us what the rest of the PE file looks like. These information include number of sections, the location and sizes of PE file sections (such as code and data), characteristics and many other useful information. [10]

Section Table

The section table resides after PE header. It contains very important information for each section in the PE file including: name, file offset (PointerToRawData), actual size (VirtualSize), aligned size (SizeOfRawData), and so on.

PE Sections

PE file contains blocks called sections. These sections include : code (in which the code resides), data (in which we found data of the file), idata (that contains the imported functions), e.data (in which the exported functions resides) and so on. The most important section to us that we will work on it is the idata (Import Section).

Each section in PE file has its set of attributes, involving describing the contents of the section if it contains code, whether it's contents is read-only or read write, and so on. There are two alignment value for each section, one of them is when the file resides in hard disk and the other when the file resides in memory. These value can be found in the PE header and can be differ from each other. Each section within the PE file should starts at an offset that some multiple of these alignment value. For example: if the alignment value is 0x400 every section should start at an offset that is a multiple of 0x400 and so on.

Import Section

One of the PE file sections is the import address table (IAT). It contains the names of the names of DLLs files and the functions in those files that the PE file uses in order to execute. So the PE file uses it as a lookup table to resolve the address of any function that it use. Therefore any external function have an entry in the IAT.

An .idata section has the information on import functions used in executable files during the period of an execution. When loading a file on a memory, the Loader refer to the .idata section to move the address of each import function used in the file on Import Address Table(IAT).

VII. CRYPTOGRAPHY

As we know, information represent an important asset in most organizations (banks need information about each account, hospital need information about every patient and so on). One of the most important things that should take into account is the security of the information.

Information security is concerned with the guarantee of data quality [12]. In other words, information security providing the following services:

***confidentiality*:** the guarantee that data can't be accessible for unauthorized persons.

integrity: the guarantee that data is veritable.

availability: the guarantee that data is easily accessible.

Formerly, to guarantee information security , many physical techniques have been used. For example, when writing information on a paper, this paper can be kept in a very secure location and transferred in a sealed wrapper through a trusted carrier to guarantee its privacy. To increase security of information a handwritten signature can also be used.

In recent days however, most information saved in electronic way. This medium presents many advantages: data can be saved and accessed cheaper, working on data become very easily. Although, this way of storing data imposes high level of threats. Also, communication over networks make it easy to intercept data. So there must be a way to kept the information secure if we want to get advantages of electronic data storage and networks. One of the most important solutions is the use of cryptography.

Cryptography is a field of information security in which assurances are provided by transforming the data itself. It represents the use of many techniques to ensure a high security in different environments.

In other words, cryptography has transferred from an tool that deal with secret communications for military purposes to a science that offer a high security for systems of ordinary people. Also, that means cryptography is becoming more and more important field within computer science. In our proposed system we will use AES (Advanced Encryption Standard) method.

VIII. SYSTEM FEATURES

The proposed system has the following features:

- The difficulty of extracting the hidden information from the cover file because the following reasons:
  - The information had already been encrypted before hiding it by using AES method. This method is very strong because it uses a 128-bit key.
  - If any attacker tries to get the hidden information it will be impossible to him to guess the file that it contains the hidden information.
  - If the attacker can get the hidden information he will face a difficulty of decrypting it.
- The antiviruses can't diagnose the cover file as a virus because the proposed system hide the information in the import table so it seem to be as information about imported function.
- The information can be of any type (text, image, and so on). Also there is no limitation for the size of it.

- The information will not disrupt the exe file after hiding it. So the exe file will be executed normally after hiding process.
- The attacker can't guessing the presence of hidden information inside the exe file that's because of couldn't guessing the real size of (hidden information and exe file).

IX. THE PROPOSED SYSTEM STRUCTURE

The proposed system consists of two main steps: encryption step and hiding step.

Step 1: Encryption Step

To protect the hidden information from being extracted, we will first encrypting it by using AES method. This method can be described briefly as follow:

AES (Advanced Encryption Standard)

AES is a symmetric block cipher just like DES. This means that this method uses the same key for encryption and decryption processes. However, there is a differences between them. AES algorithm allows for different block and key sizes (i.e. not like the 56 and 64 bit of DES key and block size). In fact, the size of block and key can be picked independently from (128,160,192,224,256) bits and there is no need to be the same. The other difference is that AES is not a feistel structure. In this structure one half of the data block is utilized to modify the other half then the two halves are swapped. So in this case the complete data block is processed in parallel within each round by using permutations and substitutions.

The number of AES parameters relay on the length of the key. For example, if the size of the key is 128 then the number of rounds must be 10 whereas it will be 12 for 192 bits or will be 14 for 256 bits[13]. In general the size of the key that is used at present is 128 bits.

Rijndael was introduced to have the following features:

• Its strength against most types of attacks.
• Speed.
• Compactness of code on a on many platforms.
• Simplicity of its design.

Before discovering the encryption process, we need to know what the term 'encryption round' mean. In general, almost block ciphers algorithms includes operations that are carried out in a loop a number of times. For each loop iteration there is a different encryption key. This loop iterations are called encryption rounds. The keys that are used in each round are called schedule. Another important thing is that the number of rounds relays on the size of the key.

The flowchart for the encryption and decryption process can be viewed in figures (3) and (4) respectively.

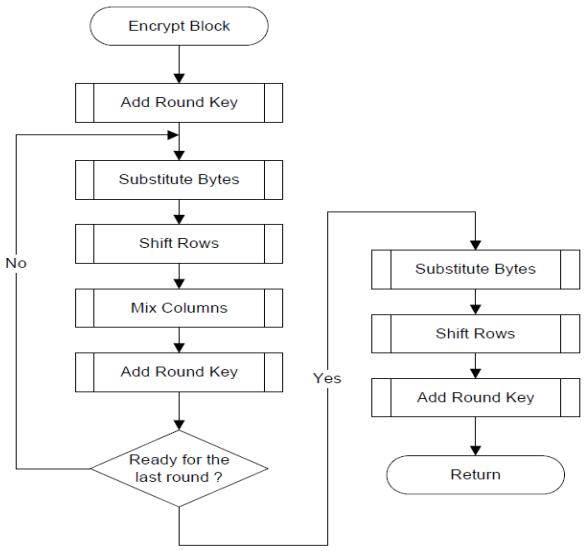


Figure (3) encryption flowchart

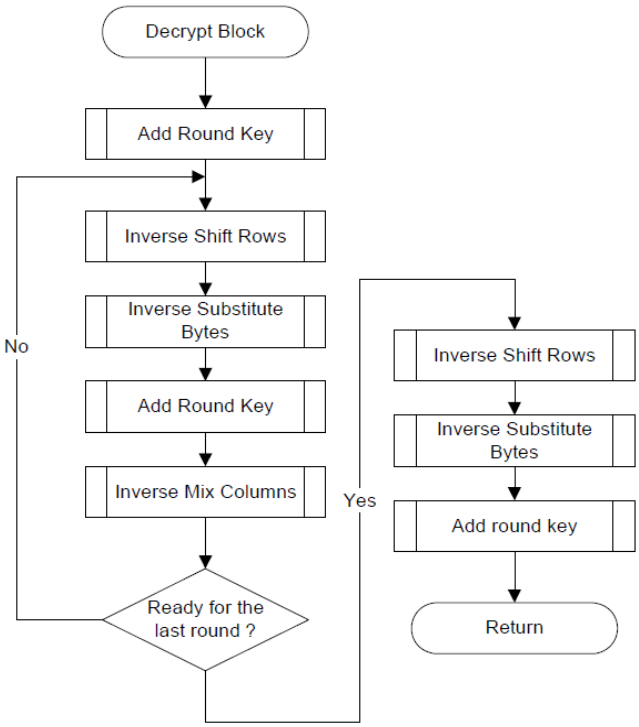


Figure (4) decryption flowchart

Step 2: Hiding Step

After encryption step we will hide the encrypted information inside the import table of exe file (that have been built under Borland). Before that we need to explore the import section more precisely.

The import section (.idata section or import address table) consists of an array of IMAGE_IMPORT_DESCRIPTORs. For each DLL that the PE file links to there is one IMAGE_IMPORT_DESCRIPTOR. The PE file hasn't any field that indicating the number of structures in this array. The last element of this array is indicated by an IMAGE_IMPORT_DESCRIPTOR that has fields filled with NULLs. The format of an IMAGE_IMPORT_DESCRIPTOR can be shown in Figure (5).

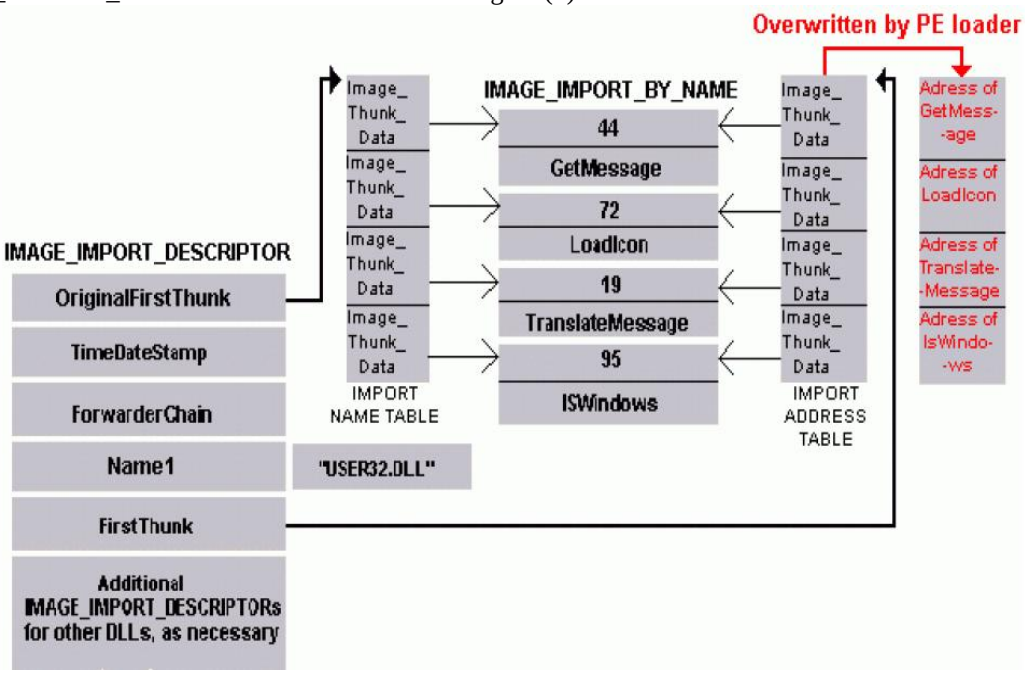


Figure (5) format of an IMAGE_IMPORT_DESCRIPTOR

X. CHARACTERISTICS FIELD

This field contains an offset (RVA) to array of pointers. These pointers points to a structure called IMAGE_IMPORT_BY_NAME.

TimeDateStamp Field

This field contains the time in which the file was built.

ForwarderChain Field

This field specifies for forwarding. Forwarding means the case when one DLL needs to send references to one of its function to another DLL.

DWORD Field

This filed contains the name of the imported DLL (like USER32 and KERNEL 32).

FirstThunk Field

This field contains an RVA to IMAGE_THUNK_DATA union. The union is
as a pointer to an IMAGE_IMPORT_BY_NAME structure.

The important parts of an IMAGE_IMPORT_DESCRIPTOR are the imported Dynamic link library name and the two arrays of IMAGE_IMPORT_BY_NAME pointers. In the PE file, the two arrays (pointed to by the FirstThunk and Characteristics fields) run in a parallel manner to each other, and are terminated by a NULL pointer entry at the end of each array. The pointers in both arrays point to an IMAGE_IMPORT_BY_NAME structure.

We will utilize an important feature in PE file that have been built under Borland. That is the characteristics field in IMAGE_IMPORT_DESCRIPTOR for PE files produced by TLINK32 is zero (hint-name array is zero). So we will hide the encrypted information inside this field[10].

This can be done by follow the following procedure:

1- Get the information file.
2- Select the cover file.
3- Check if it is produced by Borland turbo assembler or not. This can be done by checking the section table (section headers) for the cover file. The first field in this section is the (Name) field which is an 8-byte, null padded UTF-8 encoded string. If the exe file have been built under Microsoft assembler this field will contains (.Text), but if it have been built under Borland turbo assembler it will be (.Code) as shown in figure (6).
4- Set a flag inside the exe file to indicate that is a cover file.
5-Get the start address of import section.
6- From the characteristics field, get the address of Hint Name Array.
7- Copy the encrypted information into the Hint Name Array.
8- Close the cover file.

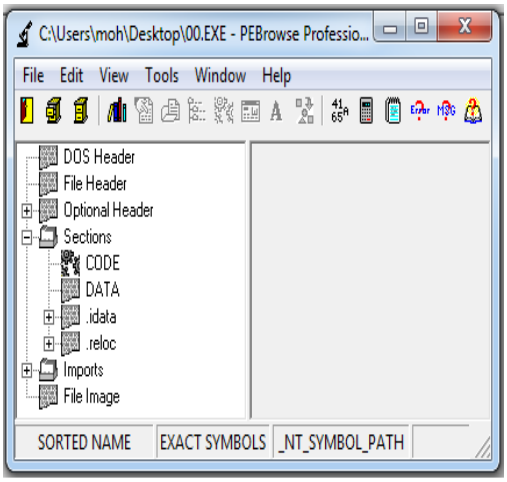


Figure (6-A) Browsing PE file that have been built under Borland

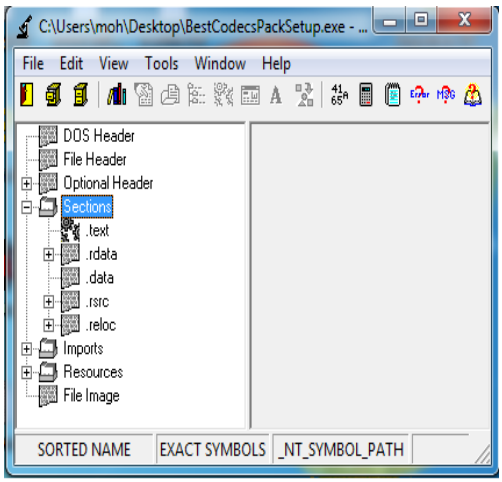
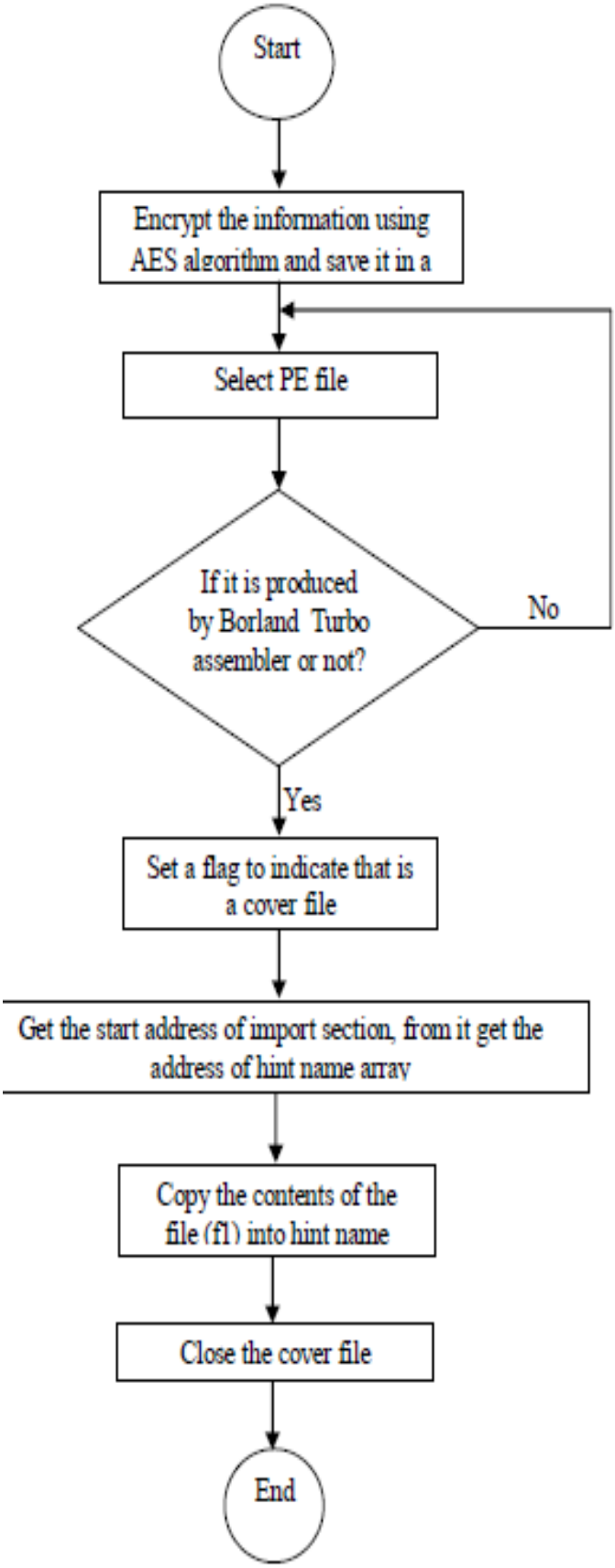
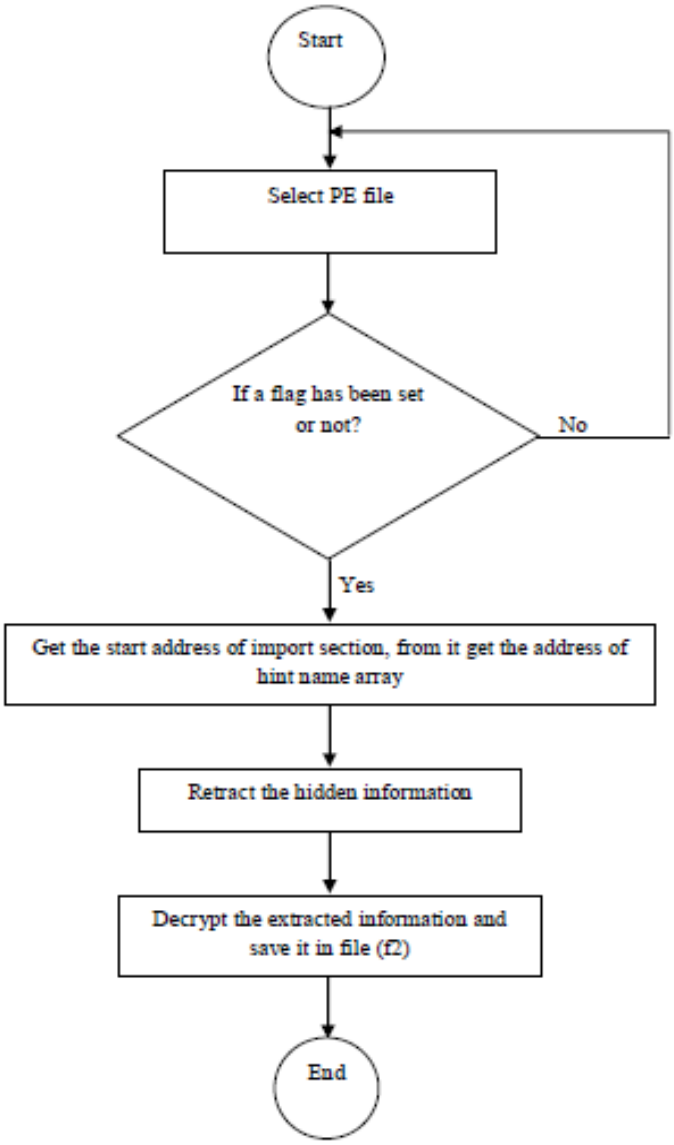


Figure (6-B) Browsing PE file that have been built under Microsoft

Figure (6) Using of PEBrowse Professional program to browse PE file

The following flowchart represents the hiding operation:





To retract the hidden information we follow the following procedure:
1- Select the file.
2- Check if it is a cover file by checking the flag inside it.
3- Get the start address of the import section.
4- From the characteristics field, get the address of Hint Name Array.
5- Retract the hidden information.
6- Decrypt the extracted information.
7- Write the decrypted information into a file.
The following flowchart represents the retraction operation:

XI. RESULTS

The proposed system have been tested under the following circumstances:

- Using windows 7 operating system.
- Using four kinds of information to be hidden (text, image, audio, video).
- Using four cover files (four EXE files), one for each kinds of the above information.
- Using ESET Smart Security 6.0 and Avira Antivirus programs.

After encrypting each type of the input files by using AES algorithm, we hide them in the PE file (cover file) individually. We noticed that the PE file still usage after the hiding operation. Also, we scan the cover file by using the above Antiviruses and we noticed that it is undetectable by these programs.

XII. DISCUSSION & CONCLUSION

With the increasing development of Internet, most people obtain software through it, also PE files are rapidly used by computer users. In this paper, we used exe file as a cover file because it is one of the most important types in the operating systems. The most important feature in this type of files is the inability for any users to alter or modify the contents of these files. From the proposed system we get the following conclusions:

- The cover file can be executed normally after hiding the information (i.e. the cover file not affected).
- The hidden information can be in any type.
- The use of encryption process increase the degree of security.
- The antivirus programs cannot observe the presence of the hidden information.
- PE files have a very complicated structure, so the proposed system depends on a fully knowledge about that structure any insertion for data in PE files without fully understanding of the structure may corrupt them.

REFERENCES

[1] X.-X. Niu, Information Hiding and Digital Watermark, Beijing: Beijing University of Posts and Telecommunications Press, 2004, pp. 2–10.

[2] X. Li and Z.-P. Dai, "Digital watermarking technology in remote access control," Network and Computer Security, no.10, pp. 13–17, 2011.

[3] Alaa Taqa, A.A Zaidan, B.B Zaidan ,"New Framework for High Secure Data Hidden in the MPEG Using AES Encryption Algorithm", International Journal of Computer and Electrical Engineering (IJCEE),Vol.1 ,No.5, ISSN: 1793-8198,2009, pp.589-595 .

[4] ASHOK J; RAJU Y; MUNISHANKARAIAH S; SRINIVAS K, STEGANOGRAPHY: AN OVERVIEW**,** International Journal of Engineering Science and Technology Vol. 2(10), 2010.

[5] AWNaji, AAZaidan, B.B.Zaidan, Shihab A, Othman O. Khalifu, " Novel Approach of Hidden Data in the (Unused Area 2 within EXE File) Using Computation Between Cryptography and Steganography ", International Joumal of Computer Science and NetwOIK Security (IJCSNS) , Vo1.9, No.5, ISSN : 1738-7906, pp. 294-300.

[6] AlIas Majed Hamid, Miss Laiha Mat Kiah, Hayan .T. Madhloom, B.B Zaidan,

[7] AA Zaidan," Novel Approach for High Secure and High Rate Data Hidden in the hnage Using hnage Texture Analysis", International Joumal of Engineering and Technology (IJEl) , Published by: Engg Journals P ublications, ISSN:09754042, VoU ,NO.2,P.P 63-69.

[8] Hamid.A.Jalab, A.A Zaidan and B.B Zaidan," Frame Selected Approach for Hiding Data within MP EG V ideo Using Bit P lane Complexity Segmentation", Journal of Computing (JOC), V ol. l , Issue I , ISSN: 2151-9617, P .P 108-113, December 2009, Lille, France.

[9] C. J. S. B," Modulation and Information Hiding in Images", of Lecture Notes in Computer Science, University of Technology, Malaya, Vol. 1174, pp.207-226, 2007.

[10] A.W. Naji*, Teddy S. Gunawan, A.A.Zaidan**, Othman O. Khalifa, B.B.Zaidan, Wajdi F. Al-Khateeb, Shihab A. Hameed, **"**New Approach of Hidden Data in the portable Executable File without Change the Size of Carrier File Using Statistical Technique", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.7, July 2009.

[11] Matt P. , Peering Inside the PE: A Tour of the Win32 Portable Executable File Format, MSDN Library,1994.

[12] Zhang Y.; Li T.; Sun J.; Qin R., A Novel Immune Based Approach for Detection of Windows PE Virus, Springer-Verlag Berlin Heidelberg, 2008.

[13] S. Wilson, Information Security, Mathematics, and Public-Key Cryptography, Certicom Corp., 200 Matheson Blvd W, Suite 103, Mississauga, Ontario L5R 3L7, Canada,2000.

[14] Xiao Y; Sun B; Hwa Chen H; Guizani S; Wang R, Performance Analysis of Advanced Encryption Standard (AES), IEEE,2006.