New Design for Information Hiding with in Steganography Using Distortion Techniques

Hamid.A.Jalab, A.A.Zaidan, B.B.Zaidan

***Abstract*—A Previously traditional methods were sufficient to protect the information, since it is simplicity in the past does not need complicated methods but with the progress of information technology, it become easy to attack systems, and detection of encryption methods became necessary to find ways parallel with the differing methods used by hackers, so the embedding methods could be under surveillance from system managers in an organization that requires the high level of security. This fact requires researches on new hiding methods and cover objects which hidden information is embedded in. It is the result from the researches to embed information in executable files, but when will use the executable file for cover they have many challenges must be taken into consideration which is any changes made to the file will be firstly detected by untie viruses , secondly the functionality of the file is not still functioning. In this paper, a new information hiding system is presented. The aim of the proposed system is to hide information (data file) within image page of execution file (EXEfile) to make sure changes made to the file will not be detected by universe and the functionality of the exe.file is still functioning after hiding process. 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.**

Index Terms—Information Hiding, portable executable file, Steganography, Statistical Technique.

I. Introduction

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].

II. Portable Executable File (PE-FILE)

The Program Loader that is a subset of the Windows System assumes the loading executable files into a virtual memory, so the executable files have the format that the Program Loader can identify, and the format is called PE (Portable Executable). It is necessary to know the PE format and RVA which is an address type used in the PE in order to understand the new methods for hiding information in the PE, so we briefly describe the format[2],[3].

And the address type.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 four parts [4]:

1) Characteristics of executable files.
2) Techniques Related with PE-File.
3) Executable files types.
4) PE File Layout

A. 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 (TEXT) 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. For 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 [3].

B. Techniques Related with PE-File.

- RVA

RVA is a position unit in the PE, and the RVA is used as an offset from the start-address of a PE file loaded on the memory. The start-address of a file on the memory is in Image Base that is one of the attributes of the PE file. For instance, if the Image

Manuscript received December 24,2009.

Dr. Hamid.A.Jalab- Senior Lecturer, Department of Computer Science & Information Technology, University Malaya, Kuala Lumpur, Malaysia, Email:hamidjalab@um.edu.my.

A. A. Zaidan – PhD Candidate on the Department of Electrical & Computer Engineering , Faculty of Engineering , Multimedia University , Cyberjaya, Malaysia and Email: aws.alaa@gmail.com or aws.alaa@gmail.com.

B. B. Zaidan – PhD Candidate on the Department of Electrical & Computer Engineering / Faculty of Engineering ,Multimedia University , Cyberjaya, Malaysia, bilal_bahaa@Hotmal.com.

Base of a file is 0x00400000 and one position of the file is 0x1000(RVA), the position on the memory will be 0x00401000[3],[4].

- PE Format

The header of PE format starts with MS-DOS stub that is used for printing a message, “This program cannot be run in DOS mode”, if the operating system can’t identify the PE on execution time.IMAGE_NT_HEADER located in the position after the MS-DOS stub has the information for the execution of a file, and consists of IMAGE_FILE_HEADERandIMAGE_OPTIONAL_HEADER. The IMAGE_FILE_HEADER has the information on the file, such as create time and machine type. The IMAGE_OPTIONAL_HEADER has the information on functions used in the file and on the start-address of the file on a memory, and the infor-mation is managed by IMAGE_DATA_DIRECTORY. A PE file except the header is composed of several sections that are basic unit of code or data within a PE or COFF file. IMAGE_SECTION_HEADER that is located in the position following to IMAGE_ OPTIONAL_HEADER has the information on each section. The information consists of PointerToRawData, SizeOfRawData, VirtualAddress, and VirtualSize. The PointerToRawData and the SizeOfRawData respectively mean the position of each section and the size of each section on the file. The VirtualAddress and the VirtualSize respectively mean the position of each section and the size of each section on the memory. The size of each section on the file is a multiple of FileAlignment that is in IMAGE_OPTIONAL_HEADER. If the amount of the data of a section is smaller than the size of the section that is allotted on compile time, the slack space of the section occurs. The common sections used in the PE include a .text that has program binaries, .data, .idata that has information on export and import functions, .edata, and .rsrc section. An .idata section has the information on import functions used in executable files during the period of an execution [5],[6].

C. *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 [4],[5],[6]:

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).

D. *PE File Layout*

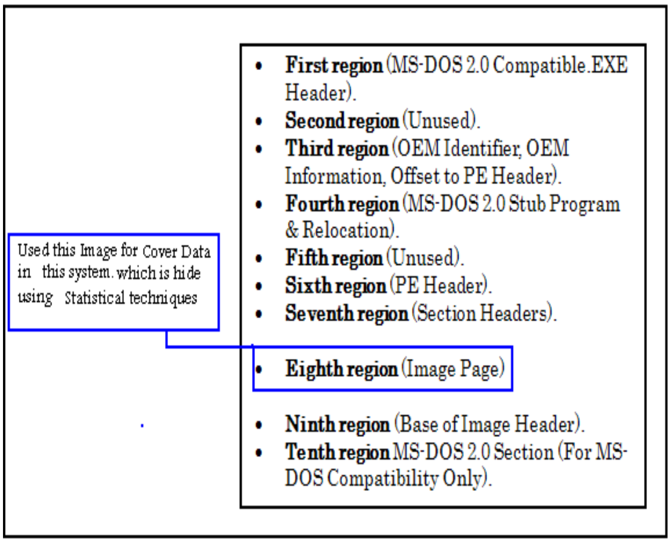
First region (MS-DOS 2.0 Compatible.EXE Header).
Second region (Unused).
Third region (OEM Identifier, OEM Information, Offset to PE Header).
Fourth region (MS-DOS 2.0 Stub Program & Relocation).
Fifth region (Unused).
Sixth region (PE Header).
Seventh region (Section Headers).
Used this Image for Cover Data in this system. which is hide using Statistical techniques
Eighth region (Image Page)
Ninth region (Base of Image Header).
Tenth region MS-DOS 2.0 Section (For MS-DOS Compatibility Only).

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

III. Steganography

A. *General Steganography Framework*

A general Steganography framework is shown in Figure 2. It is assumed that the sender wishes to send via Steganographic transmission, a message to a receiver. The sender starts with a cover message, which is an input to the stego-system, in which the embedded message will be hidden. The hidden message is called the embedded message. A Steganographic algorithm combines the cover massage with the embedded message, which is something to be hidden in the cover The algorithm may, or may not, use a Steganographic key (stego key), which is additional secret data that may be needed in the hidden process. The same key (or related one) is usually needed to extract the embedded massage again. The output of the Steganographic algorithm is the stego message. The cover massage and stego message must be of the same data type, but the embedded message may be of another data type. The receiver reverses the embedding process to extract the embedded message [4].

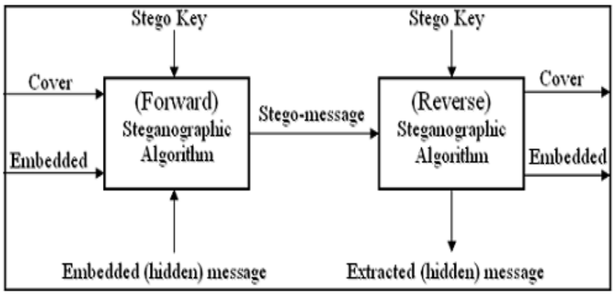
Stego Key
Stego Key
Cover
Embedded
(Forward) Steganographic Algorithm
Stego-message
(Reverse) Steganographic Algorithm
Cover
Embedded
Embedded (hidden) message
Extracted (hidden) message

Figure 2. General Steganography Framework

B. Distortion Techniques

DT is short for Distortion techniques are requires the knowledge of the original cover in the decoding process.The sender applies a sequence of modifications to the cover in order to get a stego-system.A sequence of modification is chosen in such a way that it corresponds to a specific secret message to be transmitted[5].

The receiver measures the difference in the original cover in order to reconstruct the sequence of modification applied by the sender, which corresponds to the secret message. An early approach to hiding information is in text. Most text-based hiding methods are of distortion type (i.e, the arrangement of words or the layout of a docement may reveal information). One technique is by modulating the positions of line and words, which will be detailed in the next subsection. Adding spaces and “invisible” characters to text provides a method to pass hidden information HTML files are good candidates for including image, extra spaces, tabs, and line breaks. The executable file include the image, this point make this cover to be suitable to use for this technique [5].

C. 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:

1) 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 3.
2) Developing the file handling functions depending on the existing file handling routines. This way can be performed remotely as shown in Figure 4.

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.

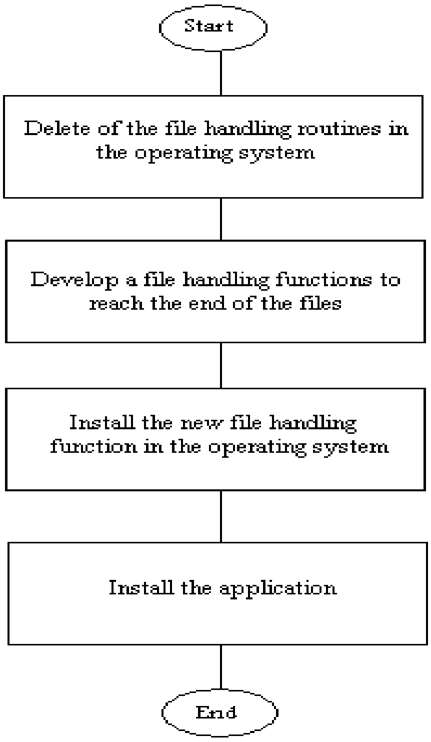
Start
Delete of the file handling routines in the operating system
Develop a file handling functions to reach the end of the files
Install the new file handling function in the operating system
Install the application
End

Figure 3. First Method of the System Concept

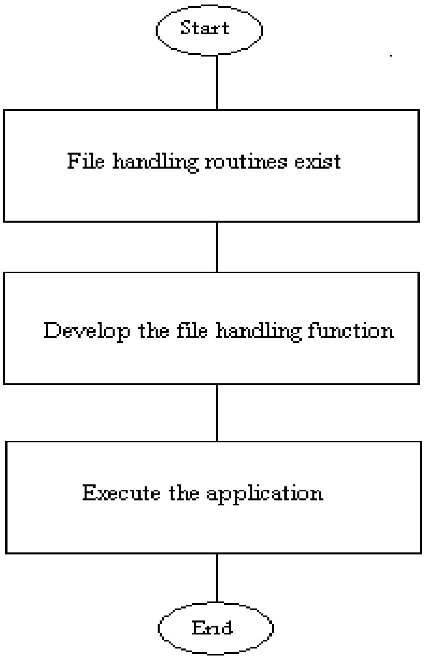
Start
File handling routines exist
Develop the file handling function
Execute the application
End

Figure 4. Second Method of the System Concept

D. System Features

This system has the following feature:

1) The hiding operation within image page of EXE file using the statistical technique increases the degree of security for the information hiding which is used in the proposed system because the data which is embedded inside the EXE file is not embed directly of EXE file , it will be hiding within image page of EXE file. So the

attacker can not be guessing the information hidden.

2) The cover file can be executed normally after hiding operation. Because the hidden information already hide in the image page 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 file (WINDOWES XP SETUP) after hiding operation it'll continued working, In other words, the EXE file can be installed of windows.

3) Virus detection programmers' can't detect such as files, the principle of antivirus check are checking from beginning to end. When checking the exe.files by antivirus, will checked it from beginning to end of it, since the principle of information hiding for this system within image page of EXE file .The information hiding will be hide inside the image page and the EXE file after hiding process is same manufacture of EXE file before hiding process. That is why the EXE file undetectable by Unit-Virus.

E. The Proposed System Structure

The system has been implementation by using Java. The block flow of hiding operation can be performed as shown in Figure 5. The following algorithm is the hiding operation procedure. The block flow of retract operation can be performed as shown in Figure 6. The following algorithm is the retract operation procedure.

The following algorithm is the hiding operation procedure:
Procedure: Hide operation.
Input: Hidden file name, cover file name.
Output: Stego-File.

- Begin (1).
- Opens the cover file (EXE file).
- Assign a pointer to the end of (Section header), which is before image page of the cover file.
- Select the image page consider normal page, it consider the cover for data.
    - Begin (2) for the image page.
    - Write the hidden file name.
    - Assign a pointer after hidden file name.
    - Write the hidden file content.
    - End(2) for the image page
- End (1).

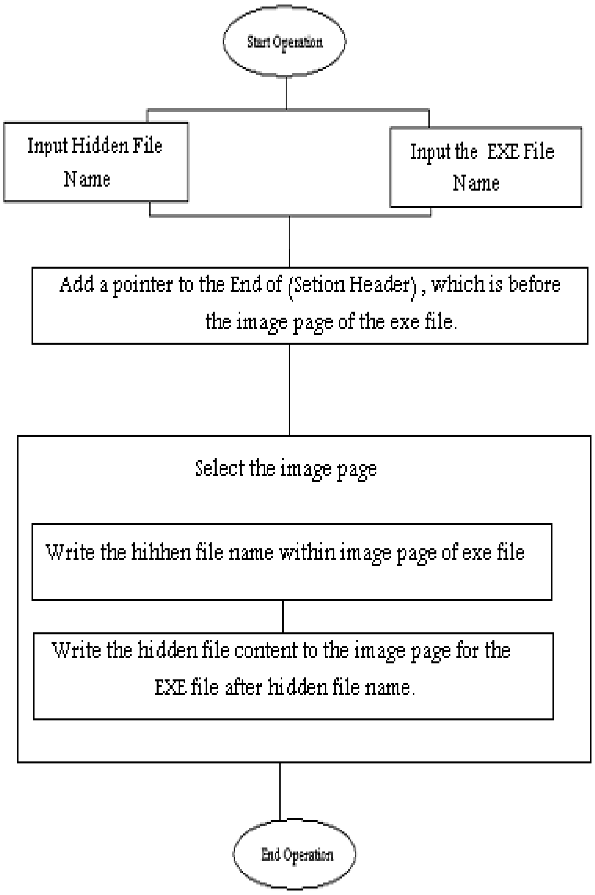
Start Operation
Input Hidden File Name
Input the EXE File Name
Add a pointer to the End of (Setion Header), which is before the image page of the exe file.
Select the image page
Write the hihhen file name within image page of exe file
Write the hidden file content to the image page for the EXE file after hidden file name.
End Operation

Figure 5. Block Flow of Hiding Operation.

The following algorithm is the retract operation procedure:

- Begin (1).
- Select the cover file.
- Get the End of the (Section Header) of EXE File.
- Select the image page:
    - Begin (2) for the image page.
    - Read the name of the hidden file.
    - Read the Hidden data.
    - Create a file using hiding file name.
    - Write in to the Create file the hiding data.
    - End (2) for the image page.
- End(1)

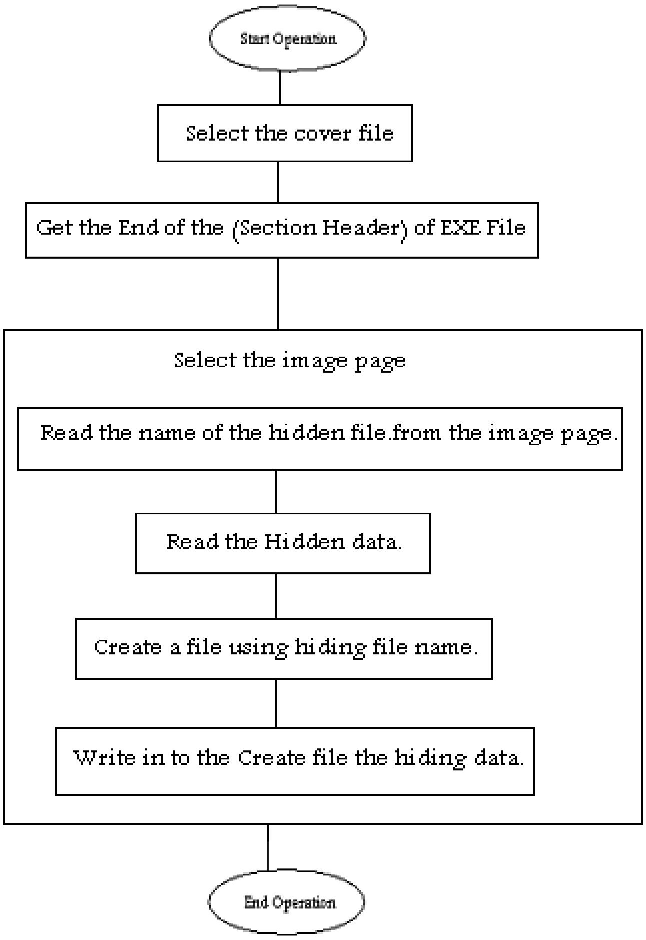
Start Operation
Select the cover file
Get the End of the (Section Header) of EXE File
Select the image page
Read the name of the hidden file from the image page.
Read the Hidden data.
Create a file using hiding file name.
Write in to the Create file the hiding data.
End Operation

Figure 6. Block Flow of Retract Operation

IV. CONCLUSION

The .EXE files are one of the most important files in operating systems and in most systems designed by developers (programmers/software engineers), and then hiding information in these file is the basic goal for this paper, because most users of any system cannot alter or modify the content of these files. We get the following conclusions:

- PE files structure is very complex because they depend on multi headers and addressing, and then insertion of data to PE files without full understanding of their structure may damage them, so the choice is to hide the information beyond the structure of these files, so the approach of the proposed system is to prevent the hidden information to observation of these systems.
- One of important conclusion most antivirus systems do not allow direct write in executable file, so the approach of the proposed system is to prevent the hidden information to observation of these systems.
- The cover file can be executed normally after hiding operation. Other word the cover file still natural, working normally and not affected.

ACKNOWLEDGEMENT

Thanks in advance for the entire worker in this project, and the people who support in any way, also I want to thank UUM for the support which came from them.

REFERENCES

[1] A.A.Zaidan, B.B.Zaidan, Fazidah Othman, “New Technique of Hidden Data in PE-File with in Unused Area One”, International Journal of Computer and Electrical Engineering (IJCEE), Vol.1, No.5, ISSN: 1793-8198, pp 669-678.

[2] A.A.Zaidan, B.B.Zaidan, M.M.Abdulrazzaq, R.Z.Raji, and S.M.Mohammed," Implementation Stage for High Securing Cover-File of Hidden Data Using Computation Between Cryptography and Steganography", International Conference on Computer Engineering and Applications (ICCEA09), Telecom Technology and Applications (TTA), Vol.19, Session 6, p.p 482-489, ISBN: 978-1-84626-017-9,June 6 (2009), Manila, Philippines.

[3] A.A.Zaidan, B.B.Zaidan, “Novel Approach for High Secure Data Hidden in MPEG Video Using Public Key Infrastructure”, International Journal of Computer and Network Security, 2009, Vol.1, No.1, P.P 71-76, 30 October, Vienna, Austria.

[4] 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, pp.589-595 .

[5] A.A.Zaidan, B.B.Zaidan, Hamid.A.Jalab,“ A New System for Hiding Data within (Unused Area Two + Image Page) of Portable Executable File using Statistical Technique and Advance Encryption Standared “, International Journal of Computer Theory and Engineering (IJCTE), 2010, VOL 2, NO 2, ISSN:1793-8201, Singapore.

[6] A.W. Naji, A.A.Zaidan, B.B.Zaidan, Ibrahim A.S.Muhamadi, “Novel Approach for Cover File of Hidden Data in the Unused Area Two within EXE File Using Distortion Techniques and Advance Encryption Standard.”, Academic and Scientific Research Organizations (WASET), International Conference on Computer, Electrical, and Systems Science, and Engineering (CCESSE09), , ISSN:2070-3724, 26-28 .

[7] A.W. Naji, A.A.Zaidan, B.B.Zaidan, Ibrahim A.S.Muhamadi, “New Approach of Hidden Data in the portable Executable File without Change the Size of Carrier File Using Distortion Techniques”, Academic and Scientific Research Organizations (WASET), International Conference on Computer, Electrical, and Systems Science, and Engineering(CCESSE09), , ISSN:2070-3724.

[8] 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), Advanced Management Science (AMS), Listed in IEEE Xplore Session 9, P.P 393-397 , ISBN:978-7695-3653-8, April 17 (2009) , Singapore.

[9] B.B.Zaidan, A.A.Zaidan, Fazidah. Othman, Ali Rahem,“ Novel Approach of Hidden Data in the (Unused Area 1 within EXE File) Using Computation Between Cryptography and Steganography ”, Academic and Scientific Research Organizations (WASET), International Conference on Cryptography, Coding and Information Security (ICCCIS09), Vol.41, Session 24, ISSN: 2070-3740.

Dr.Hamid.A.Jalab: Received his B.Sc degree from University of Technology, Baghdad, Iraq. MSc & Ph.D degrees from Odessa Polytechnic National State University 1987 and 1991, respectively. Presently, Visiting Senior Lecturer of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His areas of interest include neural networks and cryptography.

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 45 papers at various international and national conferences and journals, he has done many projects on

Steganography for data hidden through different multimedia carriers image, video, audio, text, and non multimedia carrier unused area within exe. File, Cryptography and Stego-Analysis systems, currently he is working on the multi module for Steganography, Development & Implement a novel Skin Detector. He is PhD Candidate on the Department of Electrical & Computer Engineering / Faculty of Engineering / Multimedia University / Cyberjaya, Malaysia. He is members IAENG, CSTA, WASET, and IACSIT. He is reviewer in the (IJSIS, IJCSNS, IJCSN and IJCSE).

Bilal Bahaa Zaidan: He obtained his bachelor degree in Mathematics and Computer Application from Saddam University/Baghdad followed by master from Department of Computer System & Technology Department Faculty of Computer Science and Information Technology/University of Malaya /Kuala Lumpur/Malaysia, He led or member for many funded research projects and He has published more than 45 papers at various international and national conferences and journals. His research interest on Steganography & Cryptography with his group he has published many papers on data hidden through different multimedia carriers such as image, video, audio, text, and non multimedia careers such as unused area within exe. File, he has done projects on Stego-Analysis systems, currently he is working on Quantum Key Distribution QKD and multi module for Steganography, and he is PhD candidate on the Department of Electrical & Computer Engineering / Faculty of Engineering / Multimedia University / Cyberjaya, Malaysia, He is members IAENG, CSTA, WASET, and IACSIT. He is reviewer in the (IJSIS, IJCSNS, IJCSN and IJCSE).