Comparative Analysis of Steganography and Steganalysis Forensic Tools for Images

Muhammad Bukhari Bin Burhanuddin
School of Computer Sciences
Universiti Sains Malaysia
USM, 11800 Georgetown, Penang, Malaysia
P-COM0071/19

***Abstract*—Steganography and steganalysis in digital forensic remains an active area of study as modern steganography techniques keep evolving. Steganography is an act of hiding a file or message onto any electronic medium known as cover medium which finally produced a stego-file. Meanwhile, steganalysis is the process of extracting the hidden message from the stego-file by first detecting an unsuspecting file whether it has been modified or not. This study main aim is to a do comparative analysis between two steganography tools which can also function as steganalysis tools. The two chosen tools are Digital Invisible Ink Toolkit (DIIT) and Visual Steganographic Laboratory (VSL). Both DIIT and VSL have their own advantages and disadvantages. From cross decoding experiment, the result shows that DIIT and VSL can decode each other's stego-image which expose a concerning weakness in both tools although the weakness of this study's methodology should not be ruled out. Their steganalysis functionality is adequate but left a lot to be desired compared to other proprietary steganalysis tools. Continual research and development of steganalysis are needed in order to produce a tool that could detect steganography activities as well as extracting hidden messages in a stego-image although it is highly improbable.**

***Index Terms*—steganography, steganalysis, digital forensic, digital invisible ink toolkit, visual steganographic laboratory**

I. Introduction

There is an increasing importance of securing one's data in this era of digital age. As computer technology keep advancing at a brisk pace, users are becoming increasingly exposed to various cyber security threats [1]. Against this setting, a way of protecting information transfer such as steganography is highly valuable. The origin of the word steganography and the technique itself dates back to ancient Greek carrying a literal meaning hidden writing where "steganos" means hidden while "grapho" means write [2]. One of the ways practised by civilizations hundreds of years ago was through tattooing the shaved head of their messengers with hidden messages until their hair grew again [3]. Microfilm and microdots are another examples that use steganography approach for convert communications and microfilm were used by Germany during World War II to communicate secretly [4] [5] [6] [7] [8] [9].

In today's world of rapid information exchange, steganography can be defined as the art and science of communicating hidden messages in different mediums of electronic media namely audio, image and video files where only the intended recipient is aware of the existence of such messages [10]. The antithesis or reverse engineering process of steganography is called steganalysis. A steganographer uses steganography tools to ensure the information that is being transferred via electronic medium is unreadable and hidden while on the other side of the same coin, stegoanalyst will attempt to detect and read the hidden message [11]. In any case, the goal of steganalysis is to diagnose the presence of a steganographic message and be able to analyse it to the point of reading its content [12].

The main objective of this technique paper is to compare and contrast two digital forensic tools that could perform both steganography and steganalysis on digital images. Other objective includes to investigate whether both tools could be classified as steganalysis forensic tools although they are mainly functioning as a steganography application.

II. Background Study

The notion of steganography can be further illustrated using the allegory of Prisoner's Problem by Simmons [13] describing two prisoners trying to communicate in a covert manner. The two prisoners make a plan to escape the prison and communicate with their own code while the warden would inspect every form of messages communicated between them. If the secret messages were deciphered, the prisoners' sentence will be extended, the messages get destroyed and the two prisoners will be sent into solitary confinement. Although the idea was originally introduced to describe a cryptography scenario, the basic concept is still applicable and could be modified to explain steganography. This section is written to capture the state of the subject domain covering different facet of steganography as well as steganalysis.

A. Steganography: Technical Aspect

Before the technicality of steganography is elaborated, it is essential to define some important terms that are widely used in this field and they are outlined as follows:

- Cover medium - any electronic medium that a hidden file or information will be embedded onto.
- Stego-message - hidden or secret information that will be embedded onto a cover medium to prevent anyone other than the intended recipient from detecting.

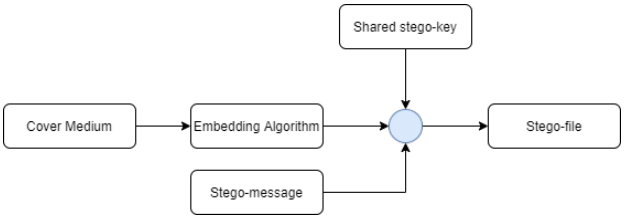

Fig. 1: Steganography process

- Stego-key - generated during the process of embedding stego-message onto cover cover medium and will be required during the extraction of stego-message.
- Stego-algorithm - technique used to embed the stego-message onto a cover file. The same stego-algorithm with the same parameters is required for the extraction of stego-message.
- Stego-file - the result of a steganographic process. For example, when a user is using an image as the cover medium, the resulting file will be called as a stego-image.

The goal of steganography is to produce a stego-file with the aid and combination of cover medium, stego-message and stego-algorithm. A stego-algorithm consists of two parts which are embedding algorithm and extraction algorithm [14] [15]. Also known as encode and decode. The embedding algorithm modifies the cover medium by embedding the stego-message onto the file to the point that the resulting file looks unchanged and not a single person will suspect that it has been altered.

The extraction algorithm is typically implemented by the same tool called decoding. It is used to invert the embedding process until the stego-message is recovered. There are numerous stego-algorithms implemented by the people developing and practicing steganography and sometimes such methods are combined. One of them is Least Significant Bit (LSB) where it uses compression techniques, modifying and manipulating image properties such as its luminance [16]. The success factor of a steganography process depends heabily on the stego-algorithm not destroying the hidden message as well as the cover medium's integrity [17] [18]. Fig. 1 shows the basic framework of a steganography process.

While there are various electronic mediums that could be used, this study chose image as the cover-medium and text file as the stego-message. The main reason is they are relatively easy and fast to transfer through the internet. They also do not consume a large amount of bandwidth compared to video and music files. Another point to note is that the process of producing stego-images will be a lot faster when doing an experiment.

B. Cryptography vs. Steganography

It is wise to make a distinction between steganography and cryptography. Cryptography or cryptology is the application and study of techniques for secure communication protocols that prevent third parties or adversaries from reading private messages [19] [20]. The main purpose of cryptography is to obscure the content of messages delivered via electronic media with data encryption although it cannot hide their existence [21]. Meanwhile, steganography strives to hide the existence of information or message itself [21] [22] [23] [24]. Although some authors claimed that steganography as a form of cryptography, the latter cannot be the former [25] and there are many other important differences pointed out by the authors in [26] as shown in Table I.

TABLE I: DIFFERENCES BETWEEN CRYPTOGRAPHY AND STEGANOGRAPHY ADAPTED FROM [26]

Cryptography	**Steganography**
Encrypt the data but does not hide the existence of the communication.	It is a technique to hide the existence of the communication.
Modify the overall structure of the data.	Does not modify the overall structure of the data.
Cipher text is generated as the final result.	Strego media is generated as the final result.
Once it has been discovered no one can easily get the secret data.	Once it has been discovered anyone can get the secret data.

C. Steganography: Application

Other than hiding and securing information, steganography is also very useful in securing a copyright of a digital medium and e-document forging prevention [27]. This commercial purpose is achieved through digital watermarking. For example photo agencies could implement steganography through embedding a digital fingerprint in the media to protect their intellectual property while maintaining the integrity of the object [28]. A photo agency will have a strong case built for them in an event where an entity steals and claims the image as their own by recovering and presenting the embedded digital watermarking [6] [29] [30]. In the context of concealing messages, steganography is widely used in politics, diplomacy and military [22].

D. Steganography: Legality & Challenges

There are numerous reports from many outlets that steganogprahy is being used for illicit purposes most notably illegal and unethical activities including financial fraud, industrial espionage and a way of communication amongst members of criminal or terrorist cells although these claims lack substantial evidences [31]. For example, a number of news outlets reported that steganography and cryptography techniques may be used by terrorists to communicate covertly escaping the high surveillance of intelligence and security agencies [32] [33] [34]. Even after all these reports regarding the misuses of steganography, there is no law ruling against the uses of it which is to be expected as laws concerning the uses of technology could be circumvented, difficult to enact and even more difficult to enforce in today's digital age [35]. For instance, criminals could easily bypass collected phone call logs and any mobile phone activities that may be tied to them by using disposable mobile phone with an unregistered sim card [36].

E. Steganalysis

While modern steganography's goal is to keep the presence of its hidden message unnoticeable, steganography process modifies a medium's properties although the medium integrity looks intact from the naked eyes [18]. As such, there are traces of distortions detectable in the cover medium when using the right analysis tool. The technique of extracting hidden messages from a cover medium and detecting steganographic activities is called steganalysis [14]. Authors in [14] also argued that detection of steganographic messages does not necessarily have to reveal the hidden information as long as detecting their presence can bring a large effect and attention to other third parties. But one has to wonder what is the point of being aware of the existence of the message if you cannot extract and read it. Especially if the stego-message itself is a crucial piece of puzzle for a critical situation. To put it in a different scenario, it is like discovering a conspiracy to murder an important political figure but not taking any action to investigate it further such as its modus operandi.

Some studies such as in [28] and [17] stated that steganalysis could be an action of destroying the hidden information even if there is no knowledge or the presence of the hidden information. However, several other authors argue that the objective of forensic is to acquire the information and not destroy it although there will be some reasonable situations such action might be taken [3] [16]. Such drastic action is taken when extracting hidden messages becomes complex and it is better for the information to be destroyed than falling into unwanted hands [16] [37]. Therefore, it remains a challenge in steganalysis to extract stego-message from a stego-medium and not just identifying whether it contains hidden information or not. Outlined below is possible steganalysis attacks which are available to steganalyst depending on the information that is available:

- Steganography only attack - implemented when only the stego-file is available for steganalyst.
- Known carrier attack - implemented when both cover medium and stego-file are available.
- Known message attack - implemented when the stego-message is known and becomes available.
- Chosen steganography attack - implemented and effective when cover medium and steganographic tools are available.
- Chosen message attack - implemented when both cover medium and the stego-algorithm are available.

III. Methodology

A structured methodology is required in order to obtain cohesive and acceptable results. Fig. 2 illustrates the overall methodology process implemented for this technique paper and the details of each process is elaborated in the following subsections.

A. Define Objectives & Background Study

A set of objectives are defined so that the direction of this technique paper is clear and the expected results could be achieved. To recapitulate, the main objective is to compare and contrast two digital forensic tools that could perform both steganography and steganalysis. Other objective includes to investigate whether both tools could be classified as steganalysis forensic tools although they are mainly functioning as steganography applications. Background study was then conducted to get a thorough idea of the subject domain which consists of steganography and steganalysis.

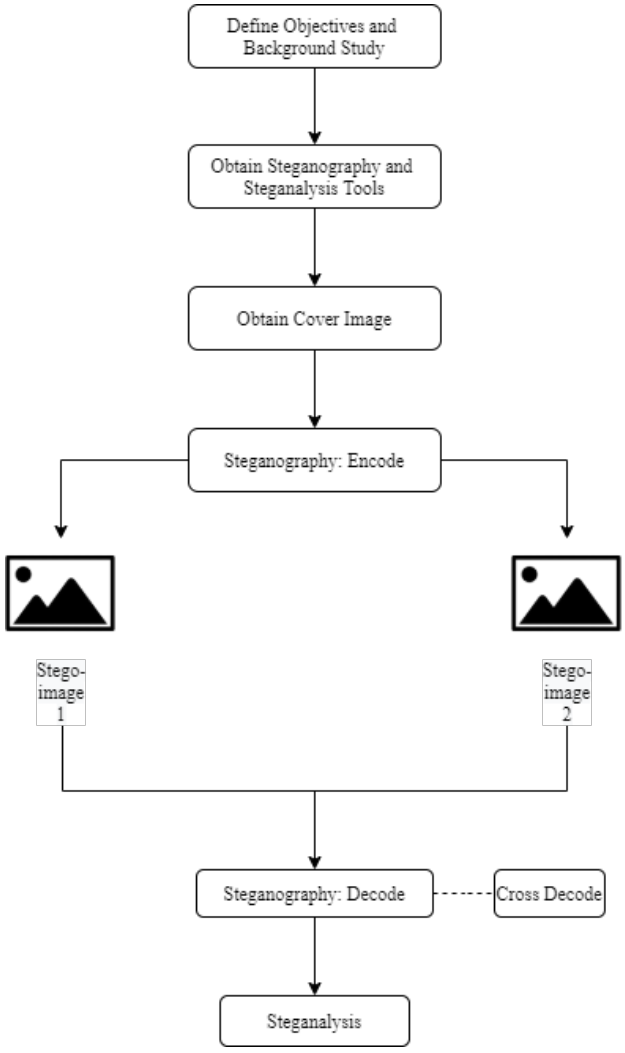

Fig. 2: Overall methodology process

B. Obtain Steganography and Steganalysis Tools

The two softwares selected for this study are Digital Invisible Ink Toolkit (DIIT) [38] and Virtual Steganographic Laboratory for Digital Images (VSL) [39] [40]. The software version of DIIT is 1.5 while for VSL is 1.1. Both tools can be executed in any operating system.

C. Obtain Cover Image

The cover image is downloaded from Pixabay website that provides free and royalty free stock images [41]. This study only used one cover image to contain the scope of

Fig. 3: Cover image

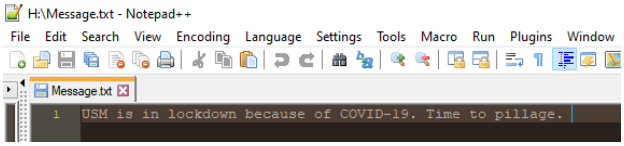

Fig. 4: Secret file

implementing steganography and steganalysis. Fig. 3 shows the cover image and below is the metadata of the image:

- Name - chemistry-covid-19-640
- Type - JPG
- Dimensions - 640 x 426
- Size - 64 KB
- Resolution - 96 dpi
- Bit depth - 24

D. Steganography: Encode

Both tools used the same cover image so that the encoding process is simple, controlled and consistent although their application of concealment algorithms were different. DIIT and VSL also used the same stego-message which has the size 57 bytes and TXT file type. Fig. 4 shows the secret file with its message and below is the basic steganography encode procedure:

1) Load cover image into the tool.
2) Load stego-message into the tool.
3) Select the encode algorithm.
4) Set the output location.
5) Execute the encoding process.
6) The steganography tool will embed the stego-message onto the cover image and the resulting image is a stego-image.

E. Steganography: Decode

Once encoding of the hidden file onto the cover images is done, decoding of the stego-images were carried out using their own steganography tool. The motive of this decoding process is to experiment the effectiveness and efficacy of the decoding functionality of each tool in reversing the process of their own encoding method. Cross decode was then carried

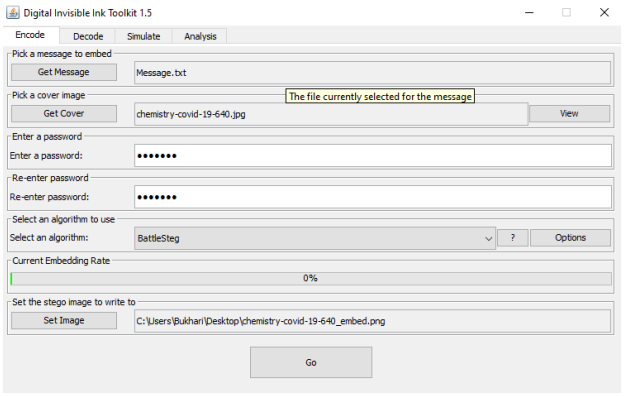

Fig. 5: GUI of DIIT

out where DITT's stego-image was inserted into VSL and vice versa. This step is important in order to investigate whether one steganography tool is capable of decoding other steganagraphy's stego-image. The basic decode procedure is described below:

1) Load stego-image into the tool.
2) Select the decode algorithm.
3) Set the output location for the extracted stego-message.
4) Execute the decoding process.
5) The steganography tool will extract the stego-message if there is any and place it in the output location.

F. Steganalysis

Finally, steganalysis is done to compare the functionality and analysis methods between the two tools. DIIT's stego-image is steganalyzed by VSL and vice versa.

IV. Implementation & Results

This section elaborated the implementation of methodology process from the the previous section and their results. The results provide comparative analysis of the two steganography and steganalysis tools covering different aspects that are separated into subsections as below.

A. Basic Functions & Graphical User Interface (GUI)

In the perspective of GUI, DIIT provides better ease of use compared to VSL. DIIT adapted the point-and-click function where it separates steganography and steganalysis tasks into four tabs as shown in Fig. 5. Below is the description of the tabs:

- Encode - embed a stego-message onto a cover image.
- Decode - extract a stego-message from a cover image.
- Simulate - simulate hiding a message on a cover image.
- Analysis - analyse a stego-image.

VSL utilizes both point-and-click and drag-and-drop functions where the functions of steganography and steganalysis are assembled in a mini window called VSL Modules as shown in Fig. 6. Since VSL is a graphical block diagramming tool, steganography and steganalysis processes can be built by dragging the modules onto the free space creating a workflow. The description of VSL Modules are provided below:

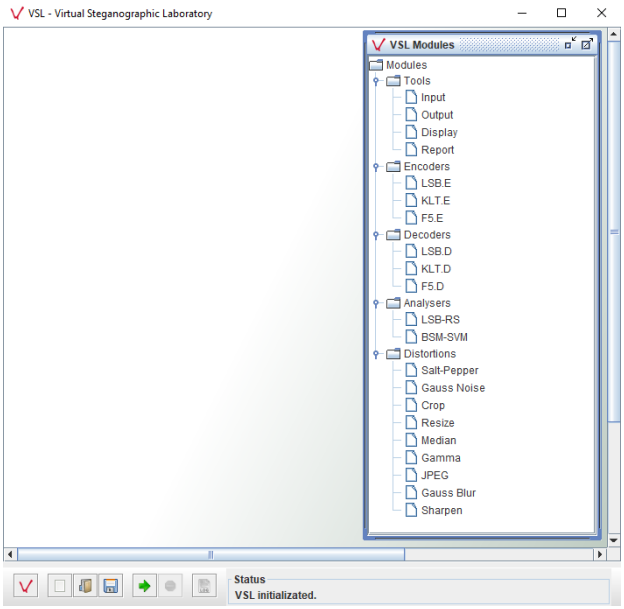

Fig. 6: GUI of VSL

- Tools - basic functions needed to construct steganography and steganalysis workflows.
- Encoders - embed a stego-message onto a cover image.
- Decoders - extract a stego-message from a cover image.
- Analysers - analyse a stego-image
- Distortions - provides various distortion techniques which can be applied to test resistance of steganographic technique.

While highly modulated which gives its users freedom to experiment, VSL is a steep learning curve for new users and the time taken to construct and deconstruct the workflows are high. DITT and VSL are written in Java and can simply be run by executing the .jar file. Both tools can be easily customised and extended since they are an open source project and their source code is available to be modified. For example, one can add a new JPEG compression module into VSL.

B. Steganography: Encode

DIIT has a major advantage against VSL in doing steganography where it can encrypt the embedded stego-message by setting a password making it harder for anyone to crack the resulting stego-image. The developer stated that the password is hashed but failed to mention the technique of hashing algorithm that was implemented [42]. The steganography process using DITT is self-explanatory as the GUI is user friendly. A pop-up hover message will be presented when the mouse pointer is hovering over any functional part of the interface as shown in Fig. 5. Another advantage of DITT is clicking the "?" button will open a new small window giving a brief explanation on how the concealment algorithm works. Both tools allow users to modify their algorithms parameter.

While DIIT has an upper hand in making its stego-image more secure, VSL could do steganography in bulk where it could insert different hidden messages and applying different

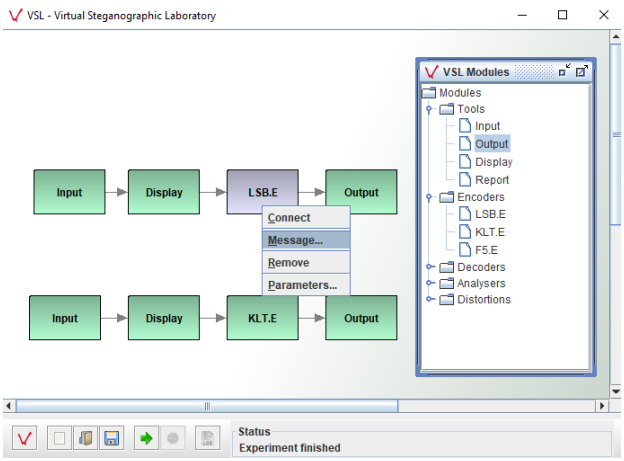

Fig. 7: VSL steganography workflow

TABLE II: Steganography options: DIIT vs. VSL

Criteria	DIIT	VSL
Input Format	JPG, PNG, BMP	JPG, JPEG, PNG, BMP, TIF, TIFF, GIF, PNG, WBMP
Output Format	PNG, BMP	JPG, JPEG, PNG, BMP, TIF, TIFF, GIF, PNG, WBMP
Hidden File Format	Any electronic file	Any electronic file
Algorithm	BlindHide, HideSeek, FilterFirst, BattleSteg, Dynamic BattleSteg and FilterFirst	Least Significant Bit (LSB), 2D Karhunen-Leove Transform (KLT), F5
Password Encryption	Yes	No
Bulk Process	No	Yes

algorithms since it is highly modulated. VSL steganography process is better represented visually hence, its basic workflow is shown in Fig. 7. Each of the modules has their own sets of options that could be viewed by right-clicking on the graphical block. For example in Fig. 7, right-clicking on the LSB encoder graphical block will bring up four options and clicking on "Message..." will prompt a small window for the user to input the stego-message.

Other differences between DIIT and VSL in steganography implementation is summarized in Table II. The result of stego-image by DIIT is named diit_chemistry-covid-19-640_embed.png using BattleSteg algorithm. Meanwhile, the result of stego-image by VIS is named vsl_chemistry-covid-19-640_embed.png using LSB algorithm.

C. Steganography: Decode

The decoding process using DIIT is also very accessible even to new users. Once the stego-image is successfully decoded and the hidden message is extracted, DIIT will prompt a window that the process is successful as shown in Fig. 8. Once the encoding workflow is built using VSL, the decoding workflow could be built with ease by replacing the third encode graphical block with LSB decode module graphical block and name the resulting extracted stego-message as shown Fig. 9.

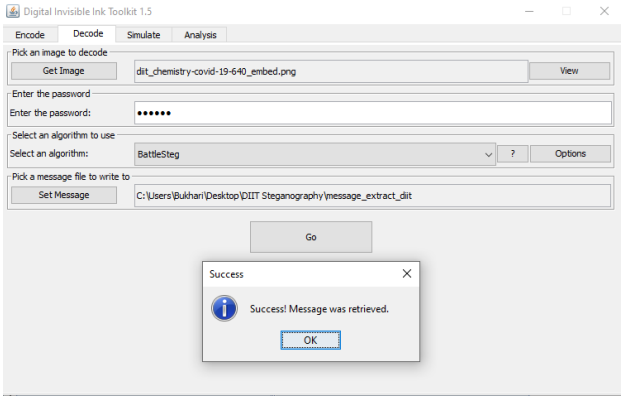

Fig. 8: Successful decode from DIIT's stego-image

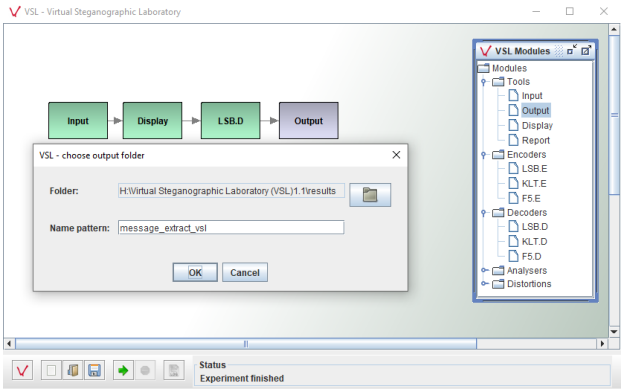

Fig. 9: Successful decode from VSL's stego-image

In Figure 10, it shows that both DIIT and VSL give their user an extra layer of security by not giving the extracted stego-message a file extension. As such, only an intended recipient that knows the steganography tool employed, algorithm applied, password and the file type will be able to decode the stego-image and obtain the stego-message. But of course the problem of unknown file type is minuscule to an experienced digital forensic investigator that will employ software utility such as hexadecimal editor to investigate and determine the file signature.

For cross decoding, DIIT was able to decode VSL's stego-image using its BlindHide algorithm which is proven in Fig. 11. One explanation of this could be that the basic framework of DIIT's BlindHide algorithm is LSB and VSL's stego-image was also encoded using LSB algorithm. Other reason is that the encoding parameters of VSL's stego-image using LSB algorithm and the decoding parameters of DIIT's BlindHide algorithm were both left at default values. Further experiment also shows that VSL's LSB algorithm could decode DIIT's stego-image that was encoded using BlindHide algorithm even when DITT's stego-image was encrypted with password as shown in Fig 12. F5 decoding algorithm of VSL could extract the hidden file from DIIT's stego-image although the extracted stego-message returns no data as shown in Fig. 13.

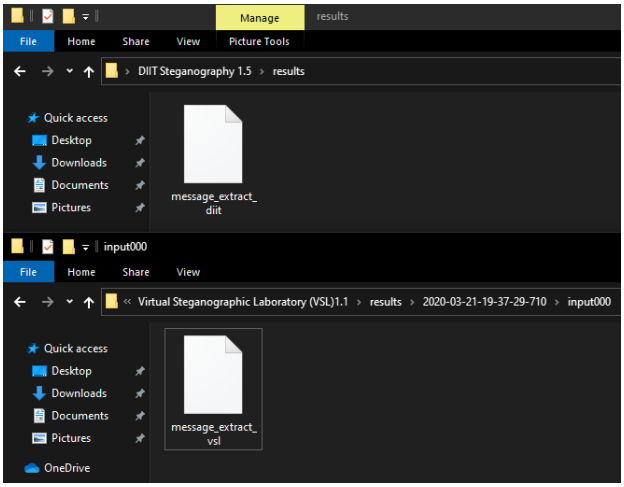

Fig. 10: Extracted hidden files with no file extension

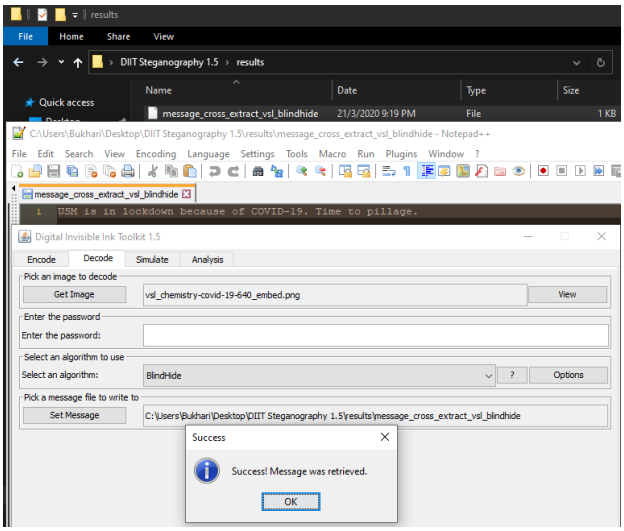

Fig. 11: Extracted hidden file from VSL's stego-image using DITT's BlindHide algorithm

D. *Steganalysis*

A user has to navigate to "Analysis" tab in order to perform steganalysis using DIIT and will be given three analysis methods which are described as following:

- Steganalysis - analyse one stego-image
- Benchmark - analyse the stego-image versus the original
- Bulk Steganalysis - analyse stego-images in bulk

Meanwhile, for VSL the steganalysis workflow is created using graphical blocks from "Tools" and "Analysers" modules as shown in Fig. 14. DIIT provides three types of analysis that could be implemented in steganalysis of a single stego-image which are RS Analysis, Sample Pairs and Laplace Graph. VSL provides two types of steganalysis techniques which are LSB-RS [43] and BSM-SVM [44]. In generating a steganalysis report, VSL gives more options to users regarding the items that could be included compared to DIIT as shown in Fig. 15. While DIIT has no option of doing bulk image steganography,

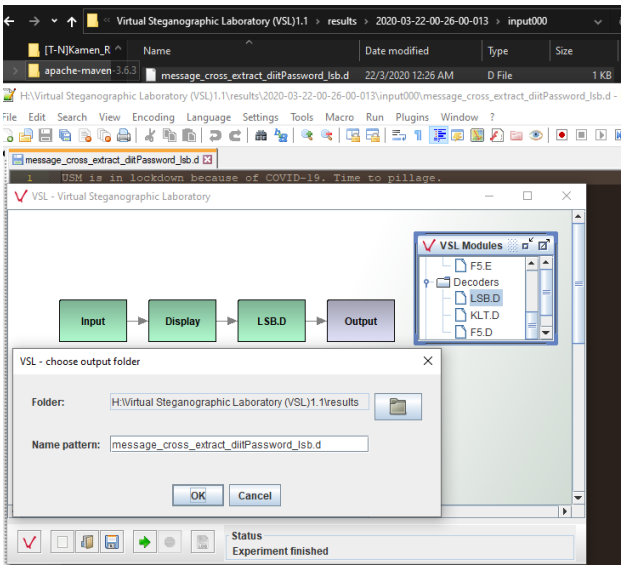

Fig. 12: Extracted hidden file from DIIT's stego-image (encrypted) using VSL's LSB.D algorithm

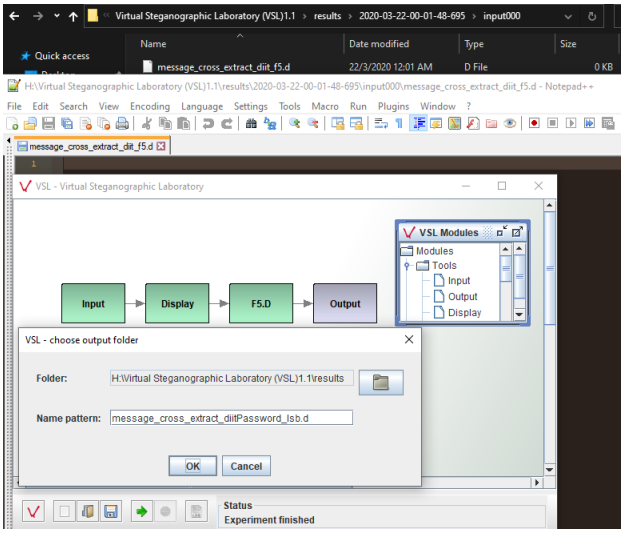

Fig. 13: Extracted hidden file with empty data from DIIT's stego-image using VSL's F5.D algorithm

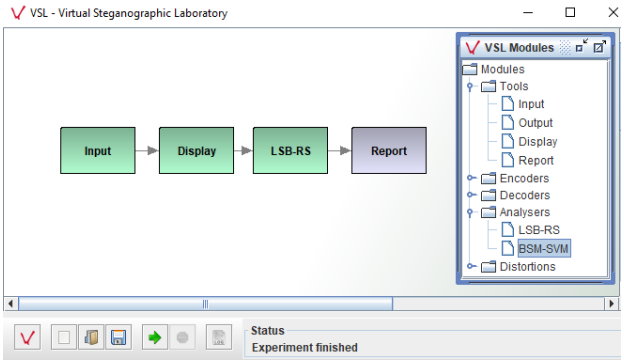

Fig. 14: VSL stegnalysis workflow

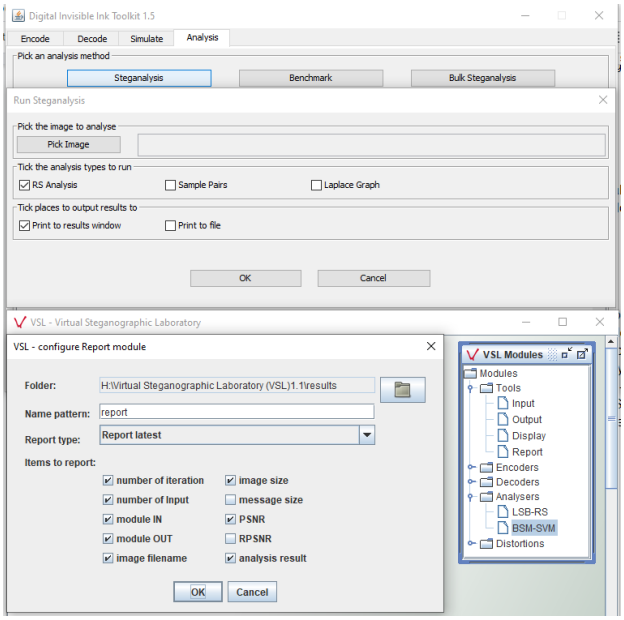

Fig. 15: Steganalysis report creation: DIIT vs. VSL

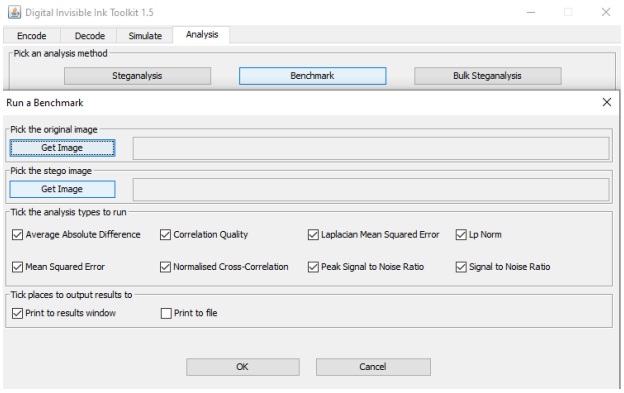

Fig. 16: Benchmark steganalysis in DIIT

both DIIT and VSL come with bulk steganalysis. DIIT has an advantage in providing more steganalysis methods where it gives users eight types of analysis to be run comparing between original image and stego-image as shown in Fig. 16.

V. Evaluation

DIIT and VSL could perform as being both steganography and steganalysis tools adequately. In the perspective of de-coding each other's stego-image, it is not a positive outlook that they both have the capability to do it when certain algorithms are tested. This could be explained as both tools are using the the same algorithm as the framework of their tool's concealment algorithm but other steganography tool such as QuickStego claim that only other users of QuickStego can retrieve and read their stego-messages [45]. As such, one would assume it would be the same case for other steganography softwares which is clearly not the case.

It is also worrying that VSL could extract DIIT's stego-image hidden file in readable format even though it has been encrypted with a strong password. This outcome begs several questions such as: Does DIIT's capability to decode VIIT's stego-image make its decode functionality a steganalysis tool and vice versa? Will DIIT and VSL be able to decode other stego-images from other steganography tools? Will DIIT be able to decode VIIT's LSB encode algorithm if it was not using default parameters and vice versa? Does other steganography softwares easily exploitable and not as secure as they advertised?

For their steganalysis performance, there is no clear indication on any study which steganalysis algorithm is more efficient to analyse stego-images as it is mainly depending on the steganalyzer which is usually a forensic statistician. The "Benchmark" function offered by DIIT will be useless if a digital forensic investigator was unable to secure the original image of a stego-image which is usually the case when the stego-image is intercepted through the network. A good steganographer would use a unique cover-image to produce a unique stego-image making it difficult for a digital forensic investigator to obtain its original image. Even when the original image is secured, one could easily compare if a stego-image contains a hidden message or not simply by implementing other forensic processes such as comparing the file size, hash values and etc. As such, DIIT's "Benchmark" function is redundant considering one could easily use other better forensic tools unless the user put importance in having various forensic functions in a single software.

The steganalysis functions provided by both tools can be considered lacking if they were to be compared with other proprietary steganalysis tools such as CANVASS that are able to process images in bulk, detect stego-embedded images and give information on what algorithm was likely used [15]. The function of providing information on the most likely algorithm being implemented is especially useful to digital forensic investigators in order to decode a stego-image faster and extract its stego-message. The developers of the CANVASS have been contacted through email in order to analyse and compare it with other tools for this paper's purpose but no response was received. This study has also found out that the majority of stegnalaysis tools in the market are still lacking. While they provide numeric outcomes from their analysis algorithms which ultimately prove an image contains hidden information, it would be useless if they could not decode the stego-image and extract its hidden content especially if the information is crucial to resolve a critical situation.

VI. Conclusion

The objectives of this technical paper are met where one is to do comparative analysis between two forensic softwares that could function as both steganogrpahy and steganalysis tools. This study also found out that both DIIT and VSL could indeed perform steganalysis although their efficacy is subjective depending on the need and capability of a forensic statistician. DIIT provides more ease of use with its simple GUI. It also presented more functionality and analysis methods to users. However, VSL is highly modulated and gives more freedom to its users by employing drag-and-drop graphical blocks to build steganography and steganalysis workflows. Future improvement of comparative analysis using more sophisticated steganalysis softwares is possible but from extensive research, they are all for commercial purpose and extremely expensive. For example, the price of StegAlayzerRTS can go as high as USD 14,995.00 as of March, 2009 [15].

Unfortunately in the application of steganalysis tools there is no one size fit all. There is still lots need to be done in the research and development of universal steganalysis that could help detecting the presence of hidden messages in stego-object accurately and ultimately enable digital forensic investigators to decode the embedded message with ease of use. Relating to the recent state of the world, this is akin to the work of medical researchers in developing a universal flu vaccine [46] [47] [48]. Ideally, a universal steganalysis tool would be very helpful but realistically there won't be any universal steganalysis tool similar to as there is no universal anti-virus software. As for now, digital forensic investigators have to settle with expensive proprietary steganalysis tools or combining different forensic tools to detect and decode hidden information embedded in a stego-object. Digital forensic investigators have to do some research, test a few hypotheses, examine steganographic signatures and perhaps develop their own tools to suit their need of steganalysis.

Acknowledgement

I would like to express gratitude to Ts. Dr. Mohd Najwadi Yusoff for his teaching and guidance.

References

[1] N. Moradoff, "Biometrics: Proliferation and constraints to emerging and new technologies," *Security Journal*, vol. 23, no. 4, pp. 276–298, 2010.
[2] P. Richer, "Steganalysis: Detecting hidden information with computer forensic analysis," *SANS/GIAC Practical Assignment for GSEC Certification, SANS Institute*, vol. 6, 2003.
[3] A. Ibrahim, "Steganalysis in computer forensics," 2007.
[4] W. White, *The microdot: History and application*. Phillips Publications, 1992.
[5] C. T. Clelland, V. Risca, and C. Bancroft, "Hiding messages in dna microdots," *Nature*, vol. 399, no. 6736, pp. 533–534, 1999.
[6] M. K. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and applications of digital watermarking and content protection*. Artech House, 2003.
[7] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures*, vol. 1. Springer Science & Business Media, 2001.
[8] D. Kahn, *The Codebreakers: The Story of Secret Writing from Ancient Times to the Internet*. Scribner, 1996.
[9] P. Wayner, *Disappearing cryptography: information hiding: steganography and watermarking*. Morgan Kaufmann, 2009.
[10] B. Schneier, *Secrets and lies: digital security in a networked world*. John Wiley & Sons, 2011.
[11] K. Bailey and K. Curran, "An evaluation of image based steganography methods," *Multimedia Tools and Applications*, vol. 30, no. 1, pp. 55–88, 2006.
[12] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, "Modeling the security of steganographic systems," in *International Workshop on Information Hiding*, pp. 344–354, Springer, 1998.

[13] G. Simmons, "the prisoners' problem and the subliminal channel, ° advances in cryptology," in *Proc. CRYPTO*, vol. 83, pp. 51–67, 1983.
[14] B. Aziz and J. Jung, "A false negative study of the steganalysis tool: Stegdetect," tech. rep., PeerJ Preprints, 2018.
[15] J. L. Davidson and J. Jalan, "Canvass-a steganalysis forensic tool for jpeg images," 2010.
[16] N. F. Johnson and S. Jajodia, "Steganalysis: The investigation of hidden information," in *1998 IEEE Information Technology Conference, Information Environment for the Future (Cat. No. 98EX228)*, pp. 113–116, IEEE, 1998.
[17] R. Krenn, "Steganography and steganalysis," *Retrieved September*, vol. 8, no. 2, 2004.
[18] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE security & privacy*, vol. 1, no. 3, pp. 32–44, 2003.
[19] H. G. Liddell and R. Scott, *A greek-english lexicon*. New York: American Book Company, 1897.
[20] R. L. Rivest, "Cryptography," in *Algorithms and complexity*, pp. 717–755, Elsevier, 1990.
[21] Y. Li, C. Xiong, X. Han, R. Xiang, F. He, and H. Du, "Retracted article: Image steganography using cosine transform with large-scale multimedia applications," *Multimedia Tools and Applications*, pp. 1–1, 2018.
[22] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 474–481, 1998.
[23] M. Ghebleh and A. Kanso, "A robust chaotic algorithm for digital image steganography," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 6, pp. 1898–1907, 2014.
[24] S. S. Chaeikar, M. Zamani, A. B. A. Manaf, and A. M. Zeki, "Psw statistical lsb image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 805–835, 2018.
[25] F. L. Bauer, *Decrypted secrets: methods and maxims of cryptology*. Springer Science & Business Media, 2002.
[26] R. Mishra and P. Bhanodiya, "A review on steganography and cryptography," in *2015 International Conference on Advances in Computer Engineering and Applications*, pp. 119–122, IEEE, 2015.
[27] S. Channalli and A. Jadhav, "Steganography an art of hiding data," *arXiv preprint arXiv:0912.2319*, 2009.
[28] J. Silman, "Steganography and steganalysis: an overview," *Sans Institute*, vol. 3, pp. 61–76, 2001.
[29] M. Barni, C. I. Podilchuk, F. Bartolini, and E. J. Delp, "Watermark embedding: Hiding a signal within a cover image," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 102–108, 2001.
[30] S. H. Kwok, "Watermark-based copyright protection system security," *Communications of the ACM*, vol. 46, no. 10, pp. 98–101, 2003.
[31] C. Hosmer and C. Hyde, "Discovering covert digital evidence," in *Digital Forensic Research Workshop (DFRWS)*, 2003.
[32] J. Kelley, "Terrorist instructions hidden online," *Retrieved September*, vol. 14, p. 2007, 2001.
[33] G. Kolata, "Veiled messages of terrorists may lurk in cyberspace," *New York Times*, vol. 30, 2001.
[34] G. Weimann, *www. terror. net: How modern terrorism uses the Internet*, vol. 31. United States Institute of Peace, 2004.
[35] M. Warkentin, E. Bekkering, and M. B. Schmidt, "Steganography: Forensic, security, and legal issues," *Journal of Digital Forensics, Security and Law*, vol. 3, no. 2, p. 2, 2008.
[36] B. Charny, "Disposable cell phones spur debates," 2003.
[37] P. Wayner, "Disappearing cryptography: Information hiding, steganography & watermarking. second eddition," 2002.
[38] K. Hempstalk, "A java steganography tool," *Source Forge*, 2005.
[39] P. Forczmański and M. Węgrzyn, "Open virtual steganographic laboratory," *Elektronika: konstrukcje, technologie, zastosowania*, vol. 50, no. 11, pp. 60–65, 2009.
[40] P. Forczmański and M. Węgrzyn, "Virtual steganographic laboratory for digital images," *Information Systems Architecture and Technology: Information Systems and Computer Communication Networks, Wrocław, Polska*, pp. 163–174, 2008.
[41] H. Braxmeier, S. Steinberger, A. Thiemermann, and O. Foma, "Pixabay," 2017.
[42] H. Kathryn, "Digital invisible ink toolkit," 2005.
[43] J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.
[44] İ. Avcıbaş, M. Kharrazi, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 17, p. 679350, 2005.
[45] "Free steganography software - quickstego," 2017.
[46] M. Khanna, S. Sharma, B. Kumar, and R. Rajput, "Protective immunity based on the conserved hemagglutinin stalk domain and its prospects for universal influenza vaccine development," *BioMed research international*, vol. 2014, 2014.
[47] R. Nachbagauer and F. Krammer, "Universal influenza virus vaccines and therapeutic antibodies," *Clinical Microbiology and Infection*, vol. 23, no. 4, pp. 222–228, 2017.
[48] L. M. Prescott, J. M. Willey, L. Sherwood, and C. J. Woolverton, *Prescott's microbiology*. McGraw-Hill Education, 2014.