November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

Reverse Engineering-based Steganalysis of Crypto123 Tool for Automatic Detection and Extraction on Concealed Messages

Hanseong Lee, Hyung-Woo Lee*
Division of Computer Engineering, Hanshin University,Osan, Gyeonggi,18101,Rep. of Korea
jkhanseong@naver.com, hwlee@hs.ac.kr*
Corresponding author : Hyung-Woo Lee, Email : hwlee@hs.ac.kr

Article Info
Volume 81
Page Number: 399 - 411
Publication Issue:
November-December 2019

Abstract

Background/Objectives: Recently, steganography tools for concealing messages in images have been widely used, and internal mechanisms for hiding messages using steganography tools are mostly unknown to public.

Methods/Statistical analysis: Therefore, we performed the reverse engineering analysis and analyzed the information hiding algorithm and operation mechanism applied to Crypto123 steganography tool. The key features of the Crypto123 software, which is rapidly increasing in use among the steganography tools, were analyzed. The IDA tool was used to reverse the Crypto123 tool. Performing complicated reverse analysis, we can find out in reverse the cryptosystem, encoding and steganographic algorithm used in Crypto123 software.

Findings: By reversing the steganography tool executable, we disclosed the internal steganographic mechanism that was wrapped in the veil. In detail, we found that the hidden messages are transformed using *Twofish* encryption algorithm after Xxencoding process for concealing message. Then, if any of the 128 bytes at the end of the cover image file has a value of '0x1A', then the subsequent bytes on cover image is replaced with the encrypted messages. Therefore, based on the software reverse engineering of the steganography mechanism applied to the Crypto123 tool, we propose a method to efficiently detect and automatically extract messages hidden in stego data using the mechanisms found correctly.

Improvements/Applications: Reverse engineering of Crypto123 software enables automatic detection of messages concealed on Internet. Therefore, it is possible to discriminate steganography tools used for covert communication.

Article History
Article Received: *3 January 2019*
Revised: *25 March 2019*
Accepted: *28 July 2019*
Publication: *22 November 2019*

Keywords: *Steganalysis, Reverse Engineering, Crypto123 Software, Hidden Message Detection, Information Hiding, Anti-Forensic Tools.*

November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

1. Introduction

Information hiding techniques that hide additional data in digital files can be classified into Covert channels, Steganography, Anonymity, and Copyright marking techniques. Steganography[1] is a concept derived from the Greek word “steganos”, which means “covered writing”. It is a branch of information privacy attempts to obscure the existence of concealed data using invisible inks, subminimal channels as a covert manner such that the existence of this communication is not detectable. Therefore, steganography is far different from cryptography which is the study of mathematical techniques related to providing information security in aspects of confidentiality, integrity, authentication and availability. And, steganalysis and steganography are the two different sides of the same coin. Steganography tries to hide messages on purpose in plain sight while steganalysis tries to detect their existence or even more to extract the concealed data. Anyway, both steganography and steganalysis received a great deal of attention in digital forensics, especially from law enforcement[2,3]. While cryptography in many countries is being outlawed or limited, cyber criminals or even terrorists are extensively using steganography tools to avoid being arrested with encrypted incriminating material or digital medium in their possession. Therefore, understanding the secret mechanisms that messages can be concealed and embedded in a digital medium and disclosure the states of the steganographicmethods to detect and retrieve hidden information, is essential parts in exposing criminal activity[4].Most of the Steganographic tools do not disclose information hiding algorithms and operation mechanisms applied software inside. And it is very difficult to determine whether information is hidden in arbitrary image file.The reasons for hiding information can be different, for example to put a digital seal to a digital content that can’t be removed or altered (digital watermarking) or to create “covert channel” where it is difficult to derive details about the transceiver. Anyone can easily download and install the Steganography tool via the Internet, and the recently developed Steganography tool can hide the encrypted message by applying the cryptography technology. Most of the Steganography tools do not disclose information hiding algorithms and operation mechanisms applied inside software. Therefore, it is very difficult to determine whether information is hidden when the message is hidden using the Steganography tool. In this study, a reverse engineering process was performed on the steganography tool “Crypto123”[5]. And we performed the reverse engineering analysis process and found the secret steganographicalgorithm and operation mechanism applied to the Crypto123 tool. We also proposed a method to automatically detect and extract hidden messageconcealed in stego image files using the Crypto123 software.The rest of the paper is organized as follows. In Section 2 the taxonomy of steganography and reverse engineering is presented. Section 3 describes reverse engineering based steganalysis on Crypto123, while Section 4 presents auto-detection and retrieval mechanism for concealed message with reverse engineering based steganalysis techniques. And Section 5 presents the conclusions derived from this approach.

2. Steganalysis and Reverse Engineering

Both steganography and cryptography technologies provide a way to conceal or securely send messages. Cryptography technology is used to provide confidentiality through the encryption process rather than to conceal messages. However, steganography is a way to make sure that the message itself is hidden.Therefore, in the steganography process, the internal mechanism is kept secret and most are not disclosed to the outside. That is, most steganography tools operate in the form of security by

obscurity. Therefore, in this study, the steganalysis process is performed by applying software reversing technique, unlike the conventional steganography technique.

2.1. *Steganalysis for Computer Forensics*

The main goal of steganalysis-basedcomputer forensic is to extract the hidden messages embedded in steganographicmedium such as text, image and video, etc. Commonly, steganographyon digital image hides secret messages into ‘cover images’ to produce ‘stego images’ that appear innocuous to an unintended observer. Popular steganographic concealment algorithms include least significant bit (LSB) steganography, group-parity steganography, and matrix embedding. In order to extract the concealed messages, we need to establish the correct order on the located payload[6]. As outlined by [7] there are certain things to consider about steganography which include

- Capacity: Number bits required to alter in order to hide the information.
- Robustness: Unaltered information after steganalysis are what make steganography useful.
- Imperceptibility: Ability to make information unnoticed.
- Security: Ability to make information difficult or impossible to interpret by third party.

In the case of steganography tools, the imperceptibility must be ensured first and foremost among the four items presented above.A basic steganographic model is shown in Figure 1.

- Cover Medium (File or Image), 'X': This is the medium that we will use for hiding the information.
- Hidden Secret Message, 'M': This is the secret data that we want to hide into cover medium 'X'.
- Secret Key, 'K': This is a steganographicsecret keys, or specific data, for hiding and recovering hidden secret message 'M' from cover medium 'X'.

Based on this concept, we can apply the steganographic method as a function ‘F(X,M,K)'. The steganographicoutput after applying the method is called "Stego Message (File or Image)", denoted with 'Z'. Therefore, we can apply the inverse process using the same Secret Key ‘K’ used for hiding the messageto recover the concealed message ‘M’ from stego message ‘Z’.

2.2. *Steganalysis by Reverse Engineering*

Engineering is the use of some scientific principles and the process of designing, assembling and manufacturing products and systems. We can classify engineering commonly into*forward engineering* and *reverse engineering*. Forward engineering (FE) is the traditional procedure or process of driving from high-level abstractions and logical designs concepts to the physical real-world implementation of a system. In some situations, there may be a physical output and product without any technical details, such as drawings, bills-of-material, or without engineering data. On the other hand, the process of duplicating or recovering an existing component, subassembly, or product without technical drawings, documentational design, or a computational model is known as reverse engineering. Reverse engineering (RE), also called back engineering,is a useful technique in software development to understand the internals of a program. Reverse engineering means using complicated engineering techniques to disclosureand discover the underlying ideas and principles governing how a machine, computer program, software, or other technological software tool works. Reverse engineering cannot only be applied to programs as a whole, but also to different parts and aspects of it. These can be categorized into File Structure, Protocols and Functions. Reverse engineering can help to identify the file

structure and to write a program that converts files from the proprietary format into a documented structure. And, reverse engineering enables developers to find out the relevant aspects of the communication protocol and implement these in a new program which can be extended easily. Additionally, reverse engineering can help to find out the way the original application implemented the function and gives the programmer the opportunity to implement the feature in their application. Since reverse engineering of software largely consists of recreating or disclosing the intentions and thoughts of the initial developer, the reverse engineer will have to backtrack from the low-level computational machine code and towards the originally written high-level program source codesuch as Java and C, etc. Reverse engineering can be used because of the following reasons[8].

- The original programmer or manufacturer no longer exists, but a customer needs the product or the original programmer or developer of a product no longer produces the product.
- Some bad or malicious features of a product need to be eliminated e.g., excessive wear might indicate where a product should be improved by removing degradation.
- Analyzing the internal algorithm and mechanisms by comparing its pros and cons on competitors' products.
- The above list is not exhaustive and there are many more reasons for using reverse engineering, than documented above.

However, reverse engineering without supporting tools is possible but very time-consuming. The most popular software for reverse engineering is IDA. It offers a huge list of features and aims at professional users[8].

2.3. *Reversing with IDA – The Interactive Disassembler*

The *Interactive Disassembler* (IDA) [9] is a debugger and disassembler that is often used by reverse engineers to analyze programs. It features a static code analyzer that automatically parses the functions inside the executable and names them according to their position (sub_400000 e.g.). Additionally, IDA creates a flow chart for functions to help the developer understand which code paths are used in which case. IDA can use several different debuggers on the local machine as well as attach to processes on remote systems. This allows reverse engineers to run potentially malicious code in virtual machines to analyze it from the host computer. Furthermore, IDA analyzes the exported and imported functions and automatically creates so-called *XRefs* between them. This allows the user to search for a specific import library and automatically list all occurrences of that method call in a window. Another important point is that IDA automatically parses all strings in the executable and lists them in a separate sub-view. This allows quick navigation through the code by looking for a string that is known to be used at a specific point in the application. Therefore, by running Steganalysis software directly using IDA, the internal operation process can be analyzed inversely, and through the disassembly process in the executable file, the operating mechanism and algorithm applied in the executable file can be inversely estimated. We can create an internal flowchart of the executable file by analyzing the start function in the executable file step by step, and identify various unknown information.

2.4. *Crypto123 Steganography Tool*

The Crypto123 tool[5] is a freeware with a software license, which provides the ability to conceal messages in Cover Image files of BMP and JPG formats. In order to conceal date within cover image, we must enter a password of at least 10

characters. Steganalysis procedures should be performed because detailed analysis of internal algorithms and operation mechanisms for the Crypto123 tool is not provided. Steganalysis[2,4,6] is the study of detecting and extracting messages hidden by steganography process; this is analogous to cryptanalysis applied to traditional cryptography. The main goal of steganalysis is to perfectly identify suspected internal packages, determine whether or not they have a payload encoded into them, and, if possible, recover that payload efficiently. Therefore, detecting a probable steganographic payload itself is often only part of the problem, as the payload may have been encrypted mostly in first. Encrypting the payload is not always done solely to make recovery of the payload more complicate and difficult.

Therefore, it is necessary to perform inverse analysis of the information hiding algorithm applied to the inside through the reverse engineering process of the Crypto123 Steganography tool. In addition, if the message is hidden by using the Crypto123 tool in an arbitrary file, a mechanism for automatically detecting and extracting the hidden message should be proposed.

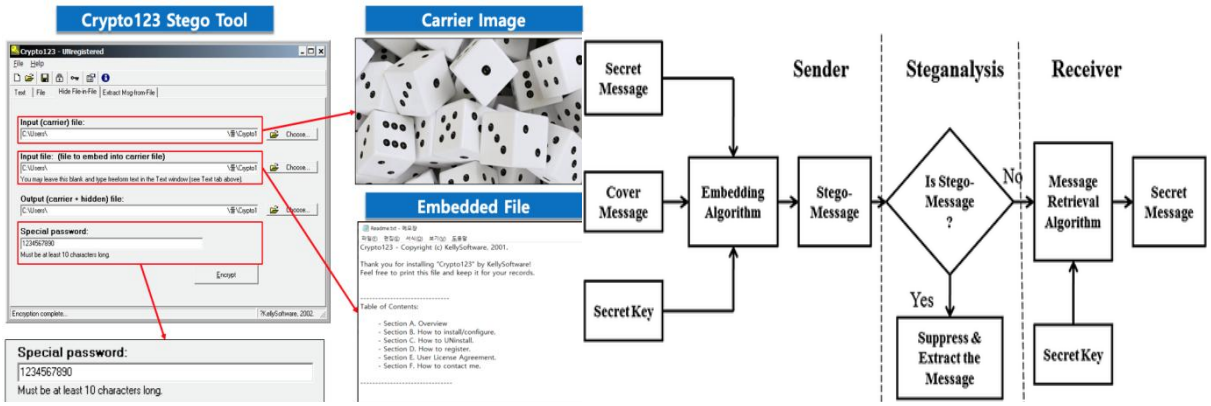


Figure 1. Steganography Tool "Crypto123" and Steganalysis Process

3. Reverse Engineering based Steganalysisof Crypto123

In order to analyze the internal structure of Crypto123 anti-forensic software, a static analysis process was performed in first on the functions and features provided by the tool basically. And then, we also performed a dynamic analysis of how data hiding is proceeding in the tool.

3.1. *Reverse Engineering based SteganalysisModel*

As shown in Figure 2 below, we performed the analysis of the characteristics of the Stego tool and the internal structure of the generated file. We could presume the internal operation mechanism of the Crypto123 tool. Based on the information obtained during the static analysis of the software, we performed a reverse engineering analysis process and analyzed the information hiding algorithm and its internal operation mechanism applied in the software. Finally, we could suggest a method to automatically identify Stego images created using Crypto123 software efficiently.

November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

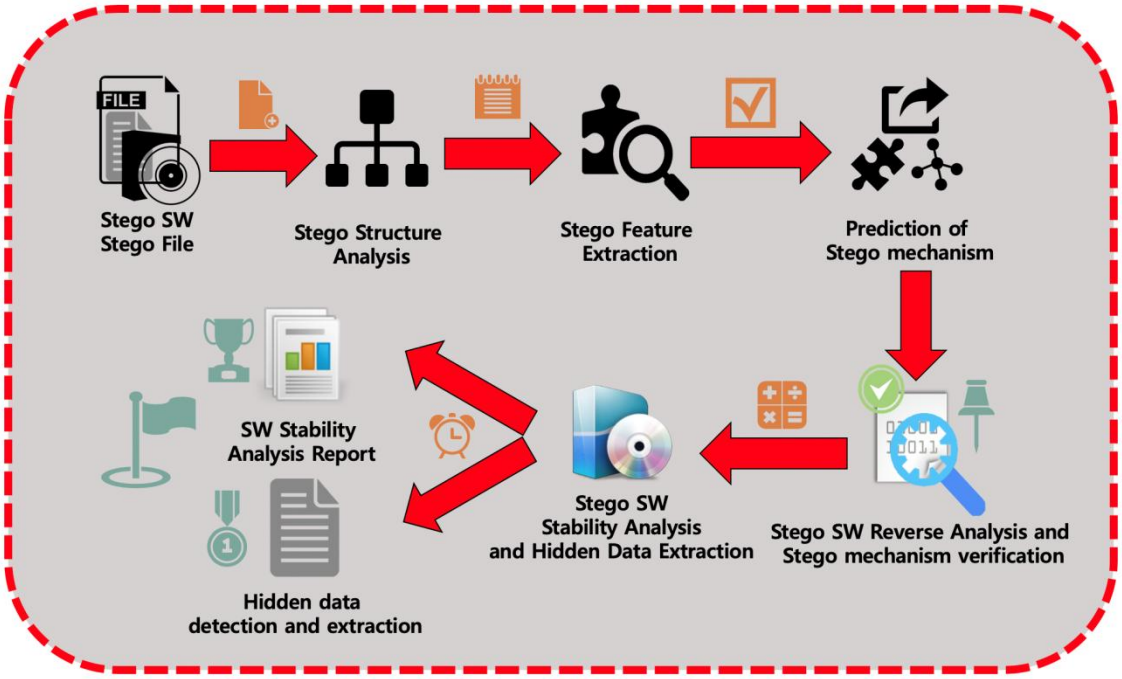
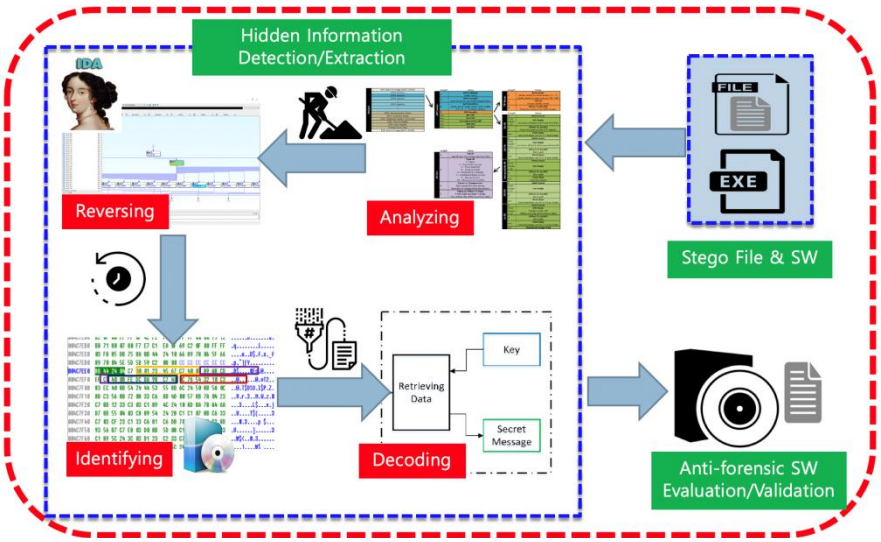


Figure 2. Reverse Engineering Process on Steganography Tool

Specifically, (Step 1) we analyzed the types of cover image files supported by the tool and the structure of the generated stego file. (Step 2) We reviewed the functions provided by the tool and adopted this information to the reverse engineering process. And then (Step 3) we analyzed the internal algorithm and operation mechanism using the reverse analysis tool IDA [9] when the Crypto123 tool is executed. After running the tool, we found the decompiled code of message hiding module, set the break point, and reversed the internal encryption mechanism and the data hiding process in the carrier image as follow Figure 3.



November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

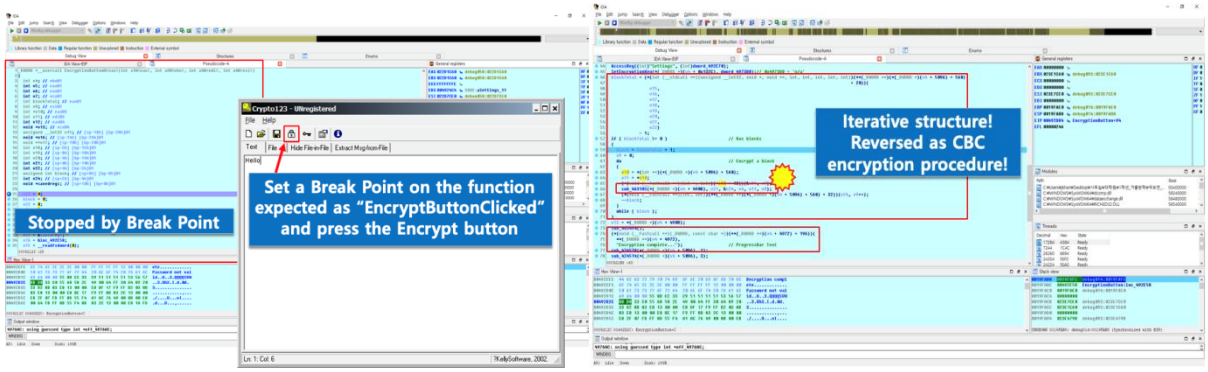


Figure 3. Reverse Analysis of Crypto123 Software using IDA

3.2. *Reverse Analysis of Crypto123 Internal Steganography Mechanisms*

As shown in Figure 4below, the Crypto123 tool was reversed using the IDA7.2 version on the Windows operating system. The following figure shows the important parts of the analysis. First, *Twofish* [10] encryption process is applied to special password. Find the 128-byte part at the end of the document that starts with 'Ox1A' and delete the subsequent part. And we found that Crypto123 tool inserted the message wanted to conceal at the end of the image after Xxencoding [11].

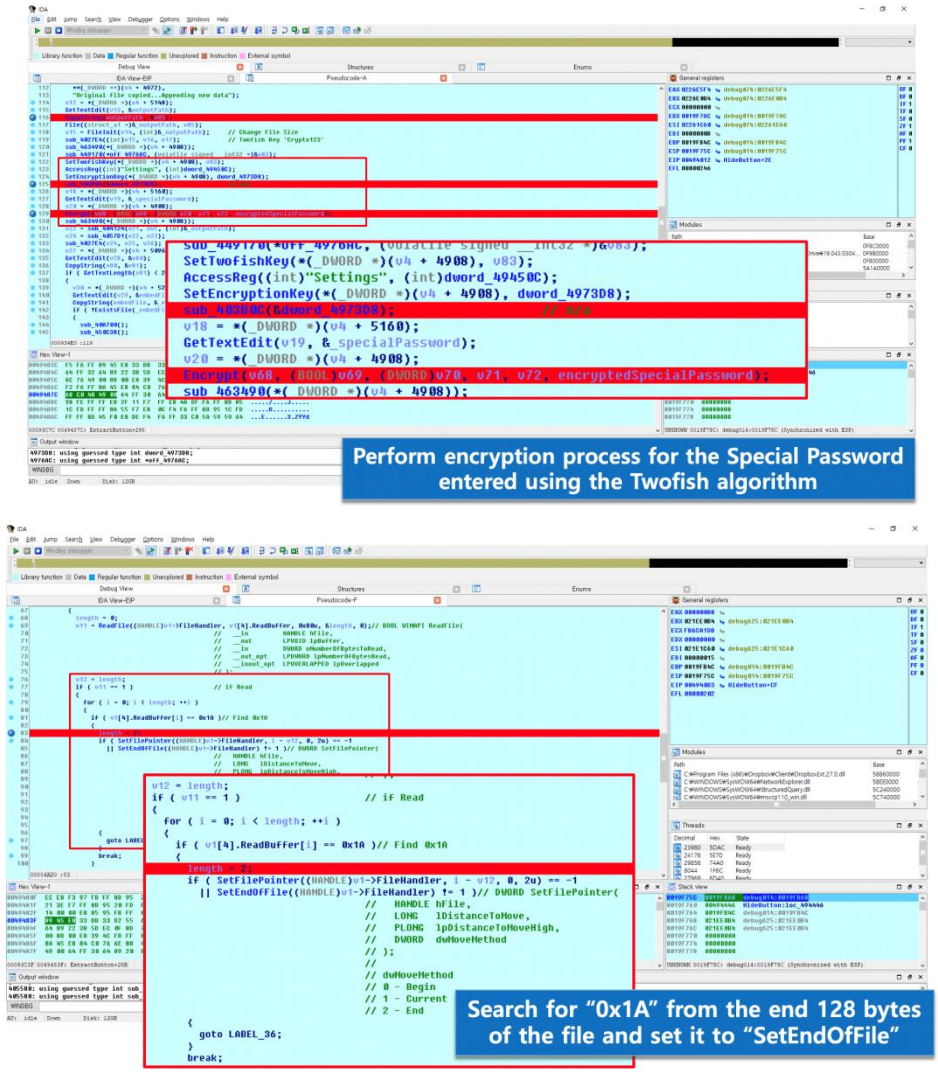


Figure 4. Secret Internal Function Analysisby Complicated ReversingProcess

The internal algorithms and operating mechanisms of the steganographicsoftware, which are identified through reverse engineering analysis on "Crypto123" software, are as follow Figure 5 with disclosed flowchart of hidden message encryption and decryption mechanisms applied to Crypto123 by performing reverse analysis.

- **Information Hiding Mechanisms**: Encryption, encoding, and padding for hidden messages.
- **Encryption Mechanism**: Encrypt the concealed message using *Twofish* symmetric key encryption algorithm [10].
- **Encryption Mode**: CBC(Cipher Block Chain) mode applied to hidden **messages**.
- **Message Encoding**: Perform Xxencoding [11] on the encrypted message block to be concealed.
- **Encryption Key**: Encrypt the special key value more than 10-digit entered by the user using the *Twofish* encryption algorithm.

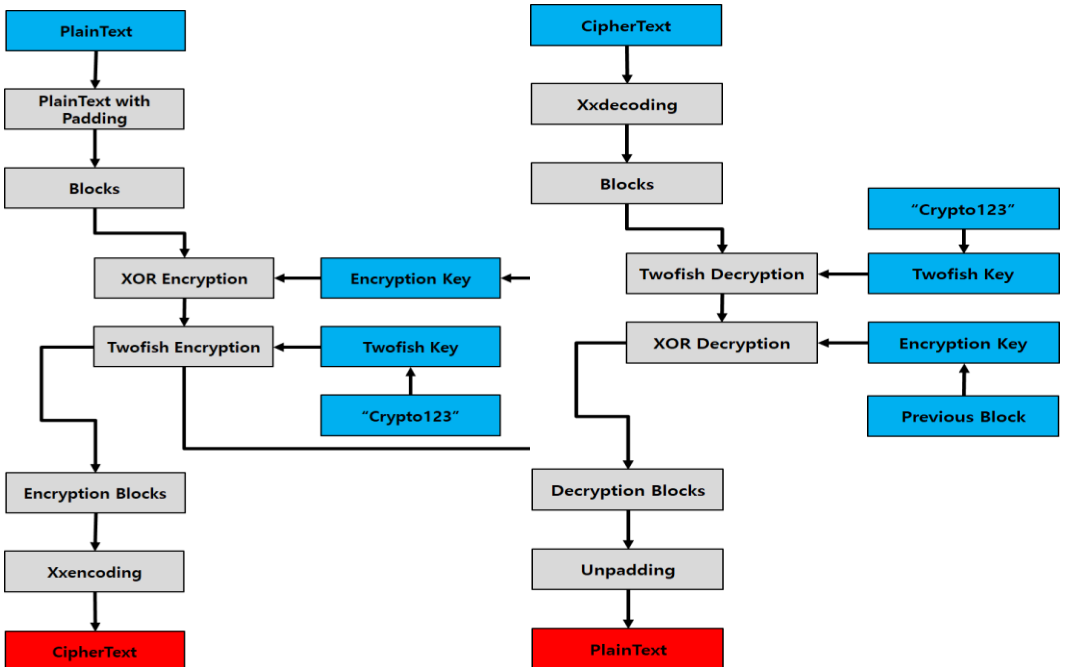


Figure 5. Disclosed Flowchart of Hidden Message Encryption and Decryption Mechanisms Applied to Crypto123Tool by Software Reverse Engineering

Twofish symmetric key encryption algorithm is a 128-bit block encryption algorithm that can support keys of various lengths up to 256 bits. It operates in the form of a fixed 4×4 bytes Maximum Distance Separable (MDS) using pseudo-Hadamard transform based on the S-box and GF (28) dependent on the key value consisting of four 8×8 bits matrices. Similar to the AES algorithm, it provides features that are difficult to decrypt. Because of this characteristic, Crypto123 software seems to use a method of encrypting the message to be hidden using *Twofish* cryptographic algorithm and padding at the end of the cover image by applying Xxencoding method to avoid detection. Therefore, reverse analysis result of message hiding process in Crypto123 tool is as follow Figure 6.

November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

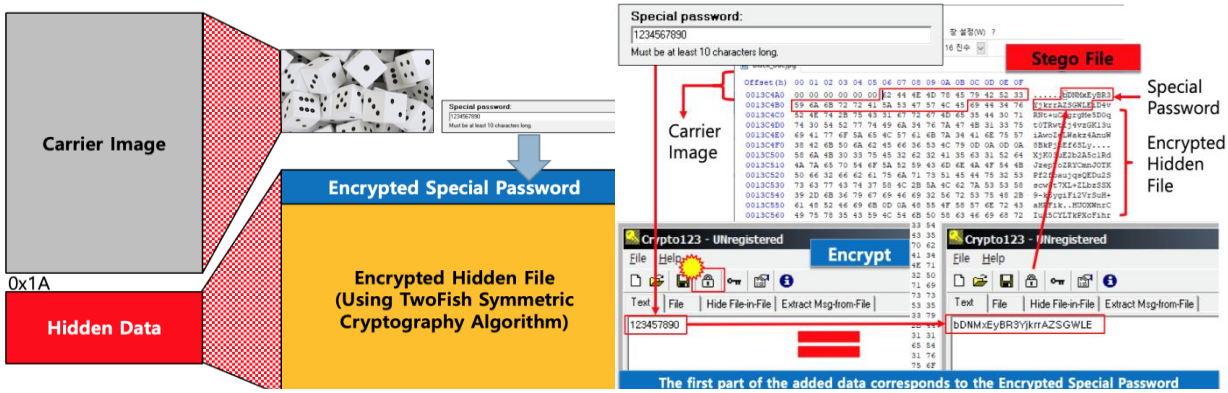


Figure 6. Reverse Analysis of Message Hiding process in Crypto123 Tool

- **Message encryption and hiding mechanism**: If there is a byte stored as 0x1A within 128 bytes at the end of the cover image file, the encrypted image data portion and the encrypted secret data are inserted after the corresponding image data portion is deleted. If there is no byte stored as 0x1A within 128 bytes, it is inserted at the end of the message.

- **Hidden message extraction and decryption mechanism**: If the same contents as the *Twofish* cipher text generated from the special password entered by the user is stored, the *Twofish* decryption process for the hidden message is performed at the rear part of the Stego image. And disclosed flowchart of steganographic concealment and extraction mechanisms applied to Crypto123 tool is shown as follow Figure 7 by performing complicated software reverse analysis processes.

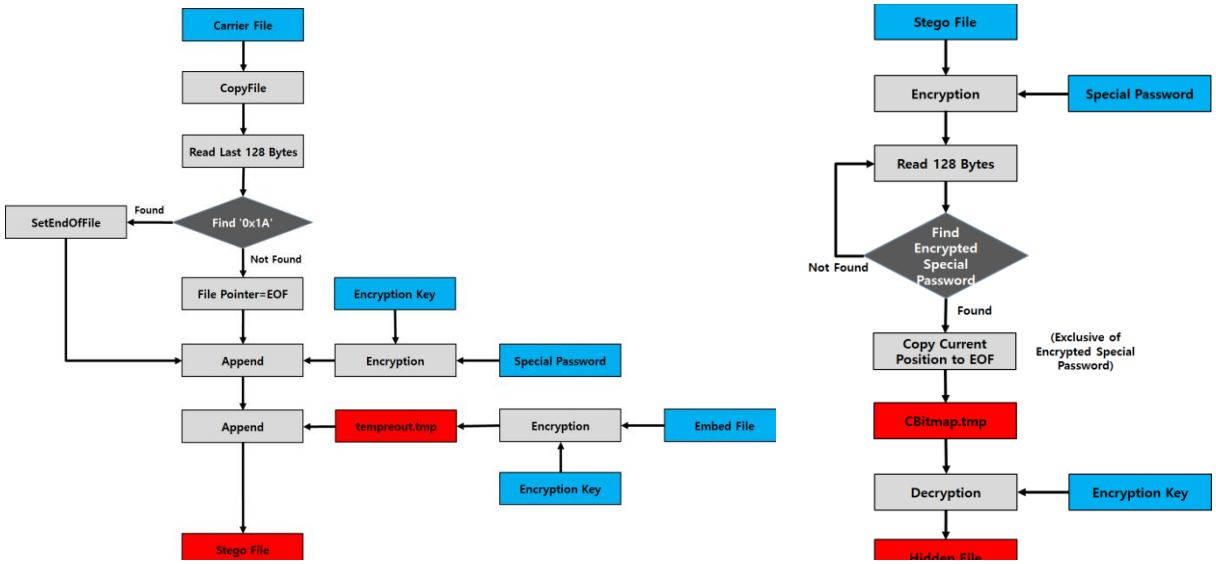


Figure 7. Disclosed Flowchart of the Steganographic Concealment and Extraction Mechanism Applied to Crypto123 Tool by Software Reverse Engineering

The Least Significant Bit (LSB) steganography is a steganographic technique in which the secret information is hidden in the least significant bits of the image by replacing LSB of image with the bits of message to be hidden[12]. Most of the Steganography tools conceal messages using LSB, but Crypto123 software applies *Twofish* symmetric key encryption

November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

algorithm to the 128 bytes at the end of the cover image, unlike the usual method. And reverse engineering analysis shows that we use padding at the end of the cover image by applying Xxencoding method to ciphertext. Therefore, if more than 128 bytes are stored as Xxencoding type of data, it is highly probable that the image is generated as Stego image using Crypto123 software.

4. Auto-Detection and Extraction Mechanism for Concealed Message

4.1. *Hidden Message Detection and Extraction Algorithmon Crypto123*

Based on the reversal results of the internal operation mechanism of the Crypto123 tool, the algorithm that determines whether the input file is hidden by the steganography tool and automatically extracts the hidden message is designed and implemented as shown in Figure 8 below. Using the feature that the Crypto123 tool performs *Twofish* symmetric key encryption process on the hidden message and applies the Xxencoding process to each block, we have developed a mechanism to automatically identify and extract the hidden message as shown below.

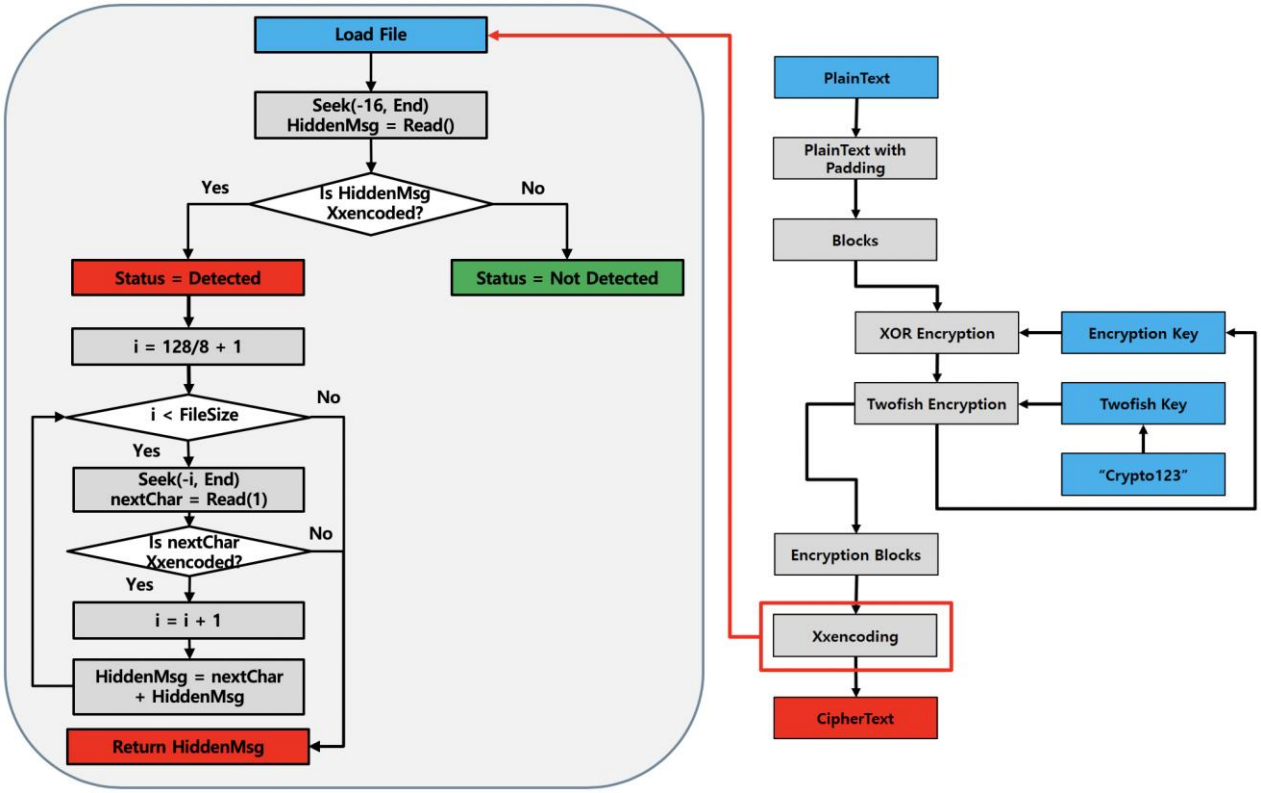


Figure 8. Detection and Automatic Extraction Algorithm for Concealed Messages using Crypto123 Tool based on Reversing Engineering-based Analysis

4.2. *Implementation of Hidden Message Detection and Extraction System*

Based on the above mechanism, the "StegoAnalyzer" system is implemented as shown below to provide automatic detection and extraction of hidden messages in stego images. The proposed system consists of Stego Analysis, Detection, Breaking, and GUI modules. Of course, not all the modules shown in the figure below are used. Among the Stego Analysis modules, only the Image Details submodule and Color Pairs were used. In the Stego Detection module, only the Format Identification module, the Anomaly Detection submodule, and the Anti-Forensic Tool Detection submodule were used.

The automatic detection program by *"StegoAnalyzer"*is implemented in Python language. After inputting three types of

November-December 2019
ISSN: 0193-4120 Page No. 399 - 411

files (TXT, BMP and JPG) and performing internal analysis on each file, 1) it is determined whether it is Stego data generated by Crypto123, and 2) if it is generated by Crypto123,in case of Stego data, it provides a function to extract concealed message inside. 3) When extracted message is Xxdecoded, ciphertext generated by using *Twofish* symmetric key encryption algorithm is obtained. However, in order to decrypt the extracted ciphertext, the special password key value used must be known. For this, Cryptanalysis process such as Brute-Force Attack must be performed. Therefore, this part is excluded from the scope of this paper. As shown in the Figure 9 below, when the above detection mechanism was applied to each of 26 TXT / BMP / JPG files, 100% of the stego files applied to Crypto123 could be detected.

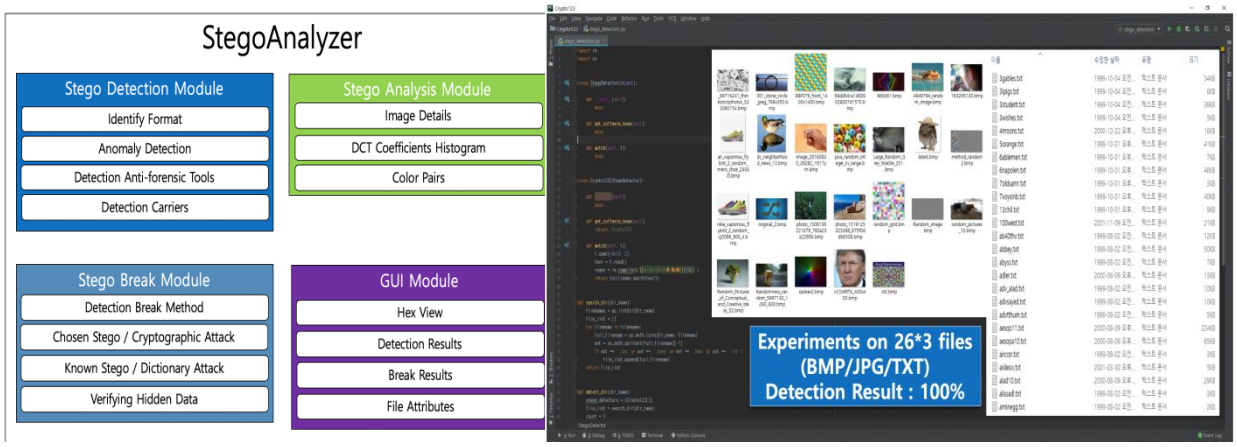


Figure 9. Implementation of StegoAnalyzer System and Experiment Results

4.3. *Comparison*

The following Table 1 shows the comparative analysis of the proposed Steganalysis method and the proposed method. Existing steganalysis techniques can be divided into *'Signature or Statistical Steganalysis'* technique [13] and *'Universal or Blind Steganalysis'* technique. Therefore, comparing the existing techniques with the Reverse Engineering-based Steganalysis techniques presented in this study, focusing on the items shown in Table 1 below, the differences between the techniques used in each methodcan be relatively compared. In addition, when the three methods are compared with each other by setting steganalysis performance, complexity, efficiency, and effectiveness as the main items, the method presented in this study seems to have relatively moreadvantage than others.

Table 1: Comparison of SteganalysisMethods

Methods &Comparison	Signature or StatisticalSteganalysis	Universal or BlindSteganalysis	Reverse Engineering-based Steganalysis
SteganalysisMethodology	Observe any repetitive patterns (signatures) or Analyze embedding procedure and determine certain statistics	Detect the embedded messages from stego& DB files regardless steganographic technique applied to find relevant features	Reverse software and disclosure embedding/extracting procedures

	modified		
Used Mechanism	Pattern Matching, LSB Matching, etc.	Machine Learning, SVM, CNN, GAN, etc.	Software Reverse Engineering
Main Objective	Detection& Extraction of Concealed Message	Classification of Used SteganographicMechanism	Disclosure of Applied Embedding and Extracting Algorithm
Core Input Data	Cover or Stego File (ex: IMG)	Stego and DB File (ex: IMG)	Executable File (ex: EXE)
Performance	Low	Middle	High
Complexity	Low	High	Middle
Efficiency	Low	Middle	High
Effectiveness	Low	High	High

5.Conclusion

Most of the Steganography tools do not disclose information hiding algorithms and operation mechanisms applied software inside. And it is very difficult to determine whether information is hidden in arbitrary image file. Furthermore, it is necessary to verify the stability of the information hiding algorithm and characteristics applied to the anti-forensic software. In this study, we analyzed the internal structure and operation method of anti-forensic software such as Crypto123 software, and verified the stability of the steganographic algorithm applied to the software. Specifically, we analyzed the information hiding algorithms and mechanisms used in software through static and dynamic analysis of steganography software. Therefore, we performed the steganographic software reverse engineering and analyzed the information hiding algorithm with operational mechanism applied to Crypto123 steganography tool. Through this complicated reversing process, we were able to determine the cryptosystem, encoding and steganographic algorithm used in Crypto123 software. Based on the results of the reverse engineering analysis, it is possible to provide a more accurate discrimination process for the stego image that has been hidden by the message, and the detection function for the hidden data can be provided. And it is expected that we can provide improved reverse engineering based steganalysisapproach for steganography software.

6. Acknowledgment

This work was supported by Hanshin University Research Grant. This work was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) (NRF-2017R1D1B03035040).

7. References

1. Wikipedia contributors. Steganography [Internet]. Wikipedia, The Free Encyclopedia; 2019 Aug 10, 16:35 UTC [cited 2019 Aug 14]. Available from: https://en.wikipedia.org/w/index.php?title=Steganography&oldid=910234335.
2. K. Karampidis, E. Kavallieratou, G. Papadourakis, A review of image steganalysis techniques for digital forensics: Journal of Information Security and Applications. Elsevier. 2018; 40:217–235. DOI:10.1016/j.jisa.2018.04.005.
3. Wikipedia contributors. Steganalysis [Internet]. Wikipedia, The Free Encyclopedia; 2019 Jul 8, 03:35 UTC [cited 2019 Aug 14]. Available

from: https://en.wikipedia.org/w/index.php?title=Steganalysis&oldid=905281752.
4. P. Richer, Steganalysis: Detecting hidden information with computer forensic analysis, SANS Institute Information Security Reading Room, 2019, [Internet] 2019 Aug [cited 2019 Aug 14]; Available from: https://www.sans.org/reading-room/whitepapers/stenganography/steganalysis-detecting-hidden-information-computer-forensic-analysis-1014.
5. Crypto123, [cited 2019 Aug 14]. Available from: https://download.cnet.com/Crypto123/3000-2092_4-10073108.html
6. F. Jessica, G. Miroslov, Practical Steganalysis of Digital Images – State of the Art, Proceedings of SPIE – The international Society for Optical Engineering 4675, DOI: 10.1117/12.465263.
7. T. Quach, Extracting hidden messages in steganographic images: Digital Investigation. Elsevier. 2014; 11:S40-S45. DOI:10.1016/j.diin.2014.05.003.
8. O. Lysne, The Huawei and Snowden Questions, ShimulaSpingerBriefs on Computing, Chap6. Reverse Engineering of Code, p.47-55, 2018 Feb; DOI: 10.1007/978-3-319-74950-1_6.
9. IDA Support: Download Center, Hex-Rays, [cited 2019 Aug 14]. Available from: https://www.hex-rays.com/products/ida/support/download.shtml
10. Wikipedia contributors. Twofish [Internet]. Wikipedia, The Free Encyclopedia; 2019 Aug 8, 12:48 UTC [cited 2019 Aug 14]. Available from: https://en.wikipedia.org/w/index.php?title=Twofish&oldid=909913280.
11. Wikipedia contributors. Xxencoding [Internet]. Wikipedia, The Free Encyclopedia; 2017 Feb 10, 16:39 UTC [cited 2019 Aug 14]. Available from: https://en.wikipedia.org/w/index.php?title=Xxencoding&oldid=764730799.
12. M. Pavani, S. Naganjaneyulu, C. Nagaraju, "A Survey on LSB Based Steganography Methods," International Journal of Engineering and Computer Science, Vol.2, Issue.8, pp. 2464-2467, August. 2013. [cited 2007 Feb 22]. Available from: https://pdfs.semanticscholar.org/496a/a39eab2244ca6b95e0f06c5d886c3b43058b.pdf?_ga=2.30132287.1643476557.1565772102-1769689455.1520330387.
13. Sabnis, S. K., Awale, R. N. Statistical Steganalysis of High Capacity Image Steganography with Cryptography: Procedia Computer Science, 2006; 79:321–327. DOI:10.1016/j.procs.2016.03.042.