中正嶺學報 第五十卷 第一期 民國 110.05
JOURNAL OF C.C.I.T., VOL.50, NO.1, MAY., 2021

Study of Implement High Availability Steganalysis Service with Multi-Server

Hsing-Han Liu*

Department of Information Management, Management College,National Defense University

ABSTRACT

In order to avoid situations where a single host performing steganalysis service stops functioning, which would cause half of the subsequent operations to fail, this paper proposed a high availability steganalysis service with multi-server. First, a steganalysis method in which features are extracted from prediction-error histogram is proposed. To find the relative features that alter due to embedding, the prediction-error histogram of images were used for analytic models. The current 5 spatial steganography methods with 100% embedding rate were used for testing. Through experimental analysis, it was discovered that the prediction-error histogram of cover images before and after embedding showed obvious differences. Therefore, the 2-D features extracted from prediction-error histogram made up the feature vector of the proposed steganalysis scheme, the experimental results effectively detect the spatial domain-based stego techniques. Second, with the high-availability feature of the Mesos architecture, the proposed steganalysis application is experimentally proven to avoid the task of interrupting execution due to external or human factors, and to achieve high-availability information hiding analysis technology.

Keywords: prediction-error histogram, steganalysis service, high availability

以多伺服器實踐高可用資訊隱藏分析之研究

劉興漢*

國防大學管理學院資訊管理學習系

摘　　要

為避免因資訊隱藏分析主機中斷服務而無法進行後續作業，故本研究提出以多伺服器實踐高可用資訊隱藏分析。首先，本研究提出由預測差值直方圖中擷取特徵的資訊隱藏分析方法。本研究使用了 5 個藏密量為 100%的資訊隱藏技術進行測試，經由實驗分析發現，藏密前後測試影像的預測差值直方圖有明顯差異。因此，從預測差值直方圖擷取的二項特徵組成了本研究所提資訊隱藏分析技術的特徵向量，實驗結果能有效偵測以空間域為基礎的藏密技術。其次，本研究配合 Mesos 架構之高可用特性部屬資訊隱藏分析程式，由實驗證實可避免因外在或人為因素影響而中斷執行中的任務，達到高可用之資訊隱藏分析技術。

關鍵詞：預測差值直方圖，藏密分析服務，高可用

文稿收件日期 109.3.20;文稿修正後接受日期 109.12.8; *通訊作者
Manuscript received March 20, 2020；revised December 8, 2020.;*Corresponding author

I. INTRODUCTION

Data communication over the internet has become common in our daily lives. However, this convenient environment is a double-edged sword. If an unauthorized user exploits the handy network to steal confidential information, the loss to individuals or groups will be difficult to estimate. To protect confidential information, various steganography methods can be used. The concept behind the steganography technique is to hide the very existence of the secret message. Steganography can conceal the secret message in an innocuous cover medium [1-3]. The purpose of steganography is to avoid arousing any suspicion against the transfer of a hidden message.

The advancements in internet and information technology provide steganography with an excellent evolution environment. Many different steganography techniques for images, such as spatial domain techniques and transform domain techniques, have been proposed. With the requirement of protecting special cover images for military, medical, or law enforcement uses, many reversible steganography schemes have been proposed.

The latest trend of reversible steganography combines histogram shift with Prediction Error (PE). PE is defined as the difference between the pixel value and its prediction value. Since the pixel values between adjacent pixels are highly correlated, PE histograms are very concentrated, which results in a high peak point. The payload of steganography based on Prediction Error Histogram Shift (PEHS) (such as Hong et al. [4], Tsai et al. [5], and Kim et al. [6]) is larger than the payload of steganography based on histogram shift.

Lately, reversible data hiding technology based on Pixel Value Ordering (PVO) has been of extensive concern to researchers in the field of reversible steganography method, because the incorporation of PVO based reversible steganography reduces pixels modification, and thus it can contribute to the high fidelity of image quality [7]. To date, several improvements of the PVO method originated by Li et al. have been proposed [8-11].

The steganography technique, however, can be abused by criminals and terrorists. It is urgent and important to develop a steganalysis technique that aims to reveal the existence of the embedded message. The purpose of steganalysis is to judge whether the suspicious image contains any embedded message. Generally, steganography is regarded as being broken when the secret messages are discovered [12]. The current steganalysis methods can be broadly divided into two categories: embedding specific or universal (blind). Embedding specific steganalysis takes advantage of the details of a specific embedding algorithm to develop a detection algorithm. If the detection targets are specific stego images, the detection performance of the embedding specific steganalysis is potentially better than the universal one. An example of embedding specific steganalysis is the RS attack [13], which can successfully detect the steganography of LSB flipping. The shortcoming of embedding specific steganalysis is that satisfactory detection performance is restricted to the specific steganography.

Universal steganalysis aims to overcome this weakness and attempts to detect the existence of the embedded message independent of a specific embedding algorithm; therefore, it can be used to perform any type of steganalysis. For example, the high order of statistics [14] is proven to be effective for the detection of various types of embedding algorithms.

Steganalysis is an important technique that has been applied to protect national and personal security. Fig. 1 shows the schematic architecture of the steganalysis technique, where the end user can filter the desired information using steganalysis. However, this steganalysis architecture assumes that the host computer used for performing the steganalysis can function and provide the relevant services properly. If the services provided from a host computer used for performing the steganalysis is interrupted owing to unexpected reasons (as shown in Fig. 2), then the end user can no longer filter the desired information using the host computer used for performing the steganalysis. Thus, the steganalysis services are interrupted.

When the steganalysis architecture or service host is interrupted by unknown reasons, the host group used for performing the steganalysis can still provide relevant services to the end user.

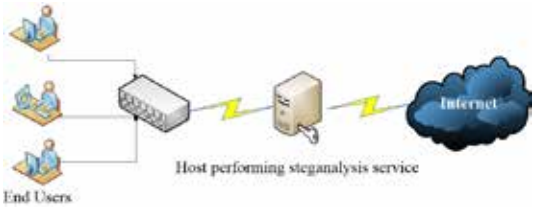

Fig.1. Steganalysis service execution architecture

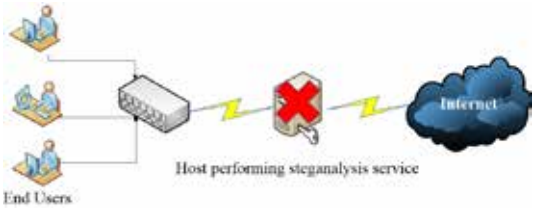

Fig.2. The host is interrupted for unknown reasons

An steganalysis service architecture with high availability (as shown in Fig. 3) can be used to avoid the scenario shown in Fig. 2, where the service provided from the host computer is negatively affected. In other words, when the primary host computer used for performing the steganalysis fails to function properly, other host computer used for performing the steganalysis can still be used to provide the service.

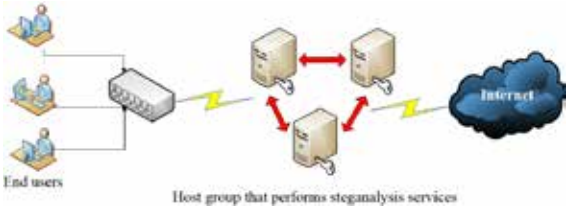

Fig.3. Host group performing steganalysisy service provides high availability service

A high availability and a strong computation capacity are two major performance indicators for the host group performing the steganalysis shown in Fig. 3. High availability implies that when a part of the host group fails to perform a specific task, the same task will be transferred automatically from the failed host computer to other functional computers and continue to be executed. Alternatively, the failed computers can be taken offline temporarily for maintenance and returned online after the maintenance has completed. However, the offline maintenance process will not affect the operation of the entire system. Increasing the number of servers and implementing the system on a network can increase the availability and reliability of the operational computers used for performing the steganalysis.

Currently, a typical service architecture that can provide a high availability for information systems is Apache Mesos [15]. Mesos is an open-source distribution resource management framework under Apache. It is the core of the distribution management system. Mesos was first developed by the AMPLab from UC Berkeley in 2009 and later used widely in Twitter. Mesos allows the sharing of available resources from a machine (or node) between many different types of work. Mesos can be considered as the core of a data center that provides a unified view of all node resources. The function of Mesos, similar to the role of the operating system core on a single host, is to provide seamless access to resources from multiple nodes [16].

To enable high availability steganalysis service from the host group, we proposed “Study of Implement High Availability Steganalysis Service with Multi-Server” in this study to resolve the issue of discontinued operation in the information system with a single host when the service is interrupted. Therefore, steganalysis service with high availability can be provided based on Apache Mesos.

II. RELATED WORKS

2.1 Mechanism Of High Availability Operation And The System Architecture

Fig. 4 shows the overall architecture of Mesos. The center of the architecture shown in Fig. 4 is the master that functions as the brain of the data enter. Multiple master nodes (one primary functional node and multiple standby nodes) can be deployed in an architecture with high availability. The ZooKeeper decentralized coordination system is used to elect the master node during the initialization and an abnormal period. All the basic resources of the data center are added to the cluster as slave nodes. Resource management and task scheduling are two separate processes. Task scheduling is handled by a framework. Different frameworks are used to serve different types of computations, such as

Hadoop, Spark, or general web services.

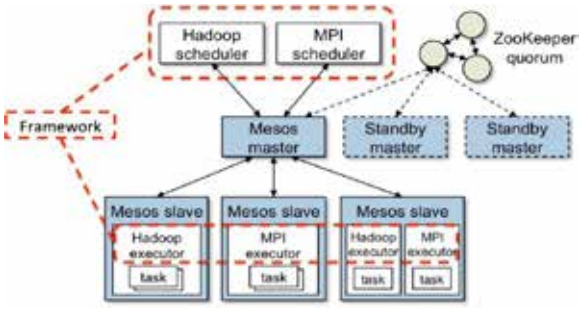

Fig. 4. The overall architecture of Mesos [15]

Mesos currently uses an HTTP-like wire protocol to communicate with the Mesos components. Mesos uses the libprocess library to implement the communication that is located in 3rdparty/libprocess. The libprocess library provides asynchronous communication between processes [16].

2.2 Steganalysis Overview

Specific steganalysis can reveal secret messages or even estimate the embedding ratio with the knowledge of the steganography algorithm. To date, many specific steganalysis techniques have been proposed. Westfeld and Pfitzmann [17] proposed the x2 (Chi-square) attack based on statistical analysis of PoV in the histogram of an image. Zaker et al. [18] presented a novel steganalysis for TPVD steganographic method based on the differences of PVD histogram. The method introduced a new steganalytic measure, named Growing Anomalies that its value has a linear relationship with secret message embedding rate.

Universal (blind) steganalysis techniques detect the existence of secret messages embedded in digital images when the steganography algorithm is unknown. Avcibas et al. [19] proposed the first universal steganalysis technique based on image quality metrics and multivariate regression analysis. Farid [20] proposed a universal steganalysis approach, which uses a wavelet decomposition, to build higher-order statistical models of natural images and employed Fisher linear discriminate (FLD) analysis to discriminate between cover and stego images. A universal steganalysis, which comprises higher-order magnitude and phase statistics extracted from multi-scale, multi-orientation image decompositions, was proposed by Lyu et al. [21]. Goljan et al. [22] proposed a blind steganalysis method. The features for the steganalysis scheme are calculated in the wavelet domain as higher-order absolute moments of the noise residue. Geetha et al. [23] presented a blind image steganalysis, which uses content-independent image quality metrics as the features of the steganalysis model and integrates genetic algorithms with the X-means model. Gul et al. [24] introduced a universal steganalysis method that models linear dependencies of image rows/columns using singular value decomposition (SVD) and employs content independency via Wiener filtering. Pevný et al. [25] utilized the local dependences between differences of neighboring cover elements and modeled as a Markov chain. The empirical probability transition matrix of the Markov chain is taken as a feature vector for steganalysis. Fridrich et al. [26] propose a general methodology for steganalysis of digital images based on the concept of a rich model consisting of a large number of diverse submodels.

2.3 Back Propagation Network

The back propagation network (BPN) [27] is a kind of feed-forward network structure with supervised learning process. The supervised learning is based on the gradient descent method, minimizing the global error on the output layer. The process of determining weight parameters is called training or learning, relying on the presentation of many training patterns. This learning process is repeated until the output error value, for all patterns in the training set, are below a specified value.

The architecture of the adopted back propagation network comprises three layers as shown in Fig. 5. The two features form the input layer, the hidden layer is composed of 40 neurons, and the output layer outputs the results of classification.

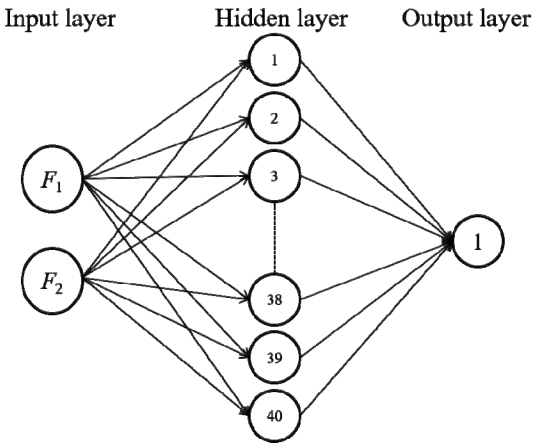

Fig. 5. The architecture of the adopted back propagation network

III. RESEARCH ARCHITECTURE

The architecture of "Study of Implement High Availability Steganalysis Service with Multi-Server" is executed with the following phases (as shown in Fig. 6).

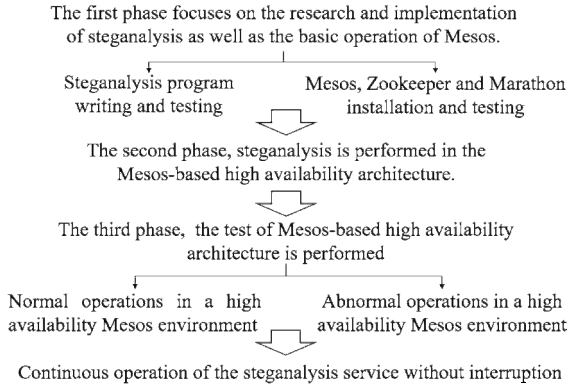

Fig. 6. The research architecture of the study

The first phase focuses on the research and implementation of steganalysis as well as the basic operation of Mesos. In the second phase, steganalysis is performed in the Mesos-based high availability architecture. The third phase involves the normal and abnormal operations in a high availability Mesos environment.

During the first phase of this study, we first developed and executed "steganalysis" using the Matlab and Python language. Meanwhile, we installed and tested Mesos, ZooKeeper, and Marathon. In this study, the required virtual machine platform is generated by VMware Workstation 15 Player, and Ubuntu 16.04 LTS is used as the associated system of the virtual machines. Three virtual machines are implemented in this study, which are named Ubuntu-Master-1 to 3. In addition, Mesos, ZooKeeper, and Marathon are installed in these virtual machines. Distributed collaborative service is indispensable for establishing a distribution system. Apache ZooKeeper [16, 28] is an important service for implementing a distributed collaborative computing system. ZooKeeper contains an algorithm called ZooKeeper atomic broadcast (ZAB). ZAB is a high-performance broadcast algorithm that ensures a high consistency in the primary-backup systems. To render Mesos functional in a high availability mode requires the installation and execution of a ZooKeeper group. A ZooKeeper group with N nodes can maintain a normal operating function when ceil (N/2) number of nodes fail. To satisfy the requirement of the following high availability tests, we installed ZooKeeper on the three nodes, including Ubuntu-Master-1 to 3. The core task of Mesos is to distribute resources fairly among a variety of applications. In addition, different frameworks dedicated to various types of operations are required for scheduling the tasks. Marathon [29] is the framework used by Mesos for long-term task processing. Such a framework is also known as the service scheduling framework. The goal of Marathon is to ensure that the commands executed by Shell can be scheduled through Marathon and executed continuously for a long time. To satisfy the requirement of the following experiments, Marathon is installed on the three nodes, including Ubuntu-Master-1 to 3.

In the second phase, the Python program developed in this study for performing steganalysis is tested in the Mesos high availability architecture. Finally, both normal and abnormal experiments are performed in the Mesos high availability architecture during the third phase. In the normal experiment condition, we investigated whether the steganalysis program can continue operating for a long time under the environment with Mesos, ZooKeeper, and Marathon. In the abnormal experiment condition, the scenario where the test machine is attacked and shut down is simulated by

manually stopping the Mesos or Marathon service (press CTRL+C in the terminal machine) as well as turning off the power supply to the virtual machine executing Mesos Master Leader and Marathon framework. Subsequently, we inspected whether the steganalysis program is still providing services.

IV. PROPOSED STEGANALYSIS SCHEME

A novel steganalysis scheme whose features are derived from the PE histogram of test images is presented. The first step was to extract the features of PE histogram of test images by using a block-sampling based predictor. Furthermore, the stego images and cover images were given diverse labels. The purpose of the different labels that are used in the probabilistic neural network (BPN) training stage was to obtain the relationship between feature sets and classification categories. The second step was to use a more flexible classifier, BPN, which is employed to discriminate between cover images and the stego images. Finally, according to the results of classification, the detection accuracy was calculated.

4.1 Features Selection And Extraction

To find the set of image characteristics that alter due to embedding, the PE histogram of images were used for analytic models. The PE is defined as the difference between the pixel value and its prediction value. The PE histogram can be used to represent the statistics of various prediction errors. PE histograms shown in Fig. 7 represent a shape that is sharply peaked at zero and has a two-sided exponential decay. Such a shape is similar to Laplace distribution. To compare with the PE histogram before and after the embedding, 3 common spatial PEHS-based steganography methods were used for the test. The test results reflected in Fig. 8 highlight the differences between the PE histogram of a cover image and the one of a stego image. To distinguish correctly between a cover image and stego image, the ratio of the distance of bins located at the PE histogram could be used. Fig. 9 supposes that H_0, H_1, H_{-1}, H_2, H_{-2}, H_3, and H_{-3} are respectively the values of '0', '1', '-1', '2', '-2', '3', and '-3' in the PE histogram. The 2-D features of the proposed steganalysis scheme are Equation 1 and Equation 2 (as follows).

$$F_1 = [(H_0-H_{-1})+(H_0-H_1)]/[(H_{-1}-H_{-2})+(H_1-H_2)] \quad (1)$$

$$F_2 = [(H_{-1}-H_{-2})+(H_1-H_2)]/[(H_{-2}-H_{-3})+(H_2-H_3)] \quad (2)$$

To prove the distinguishing effect of the eight feature values (F_1 to F_2), 2,724 cover images were retrieved from the NRCS image database and their corresponding Kim PEHS Stego Images (with 100% embedding rate) were used to extract the feature values. The results are shown in Fig. 10 and Fig. 11. For Fig. 10 and Fig. 11, the horizontal axis represents the testing images, whereas the vertical axis represents the feature value extracted from the image.

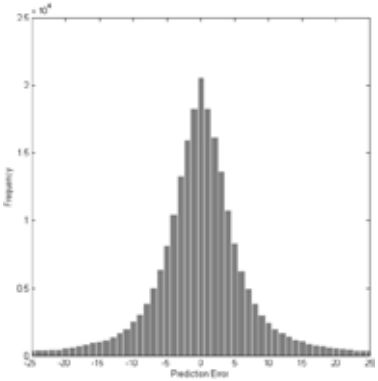

Fig. 7. PE Histogram

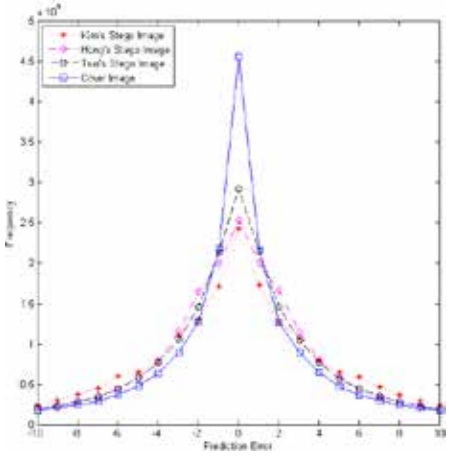

Fig. 8. Comparison of PE Histogram of Cover Image and PEHS Stego Images

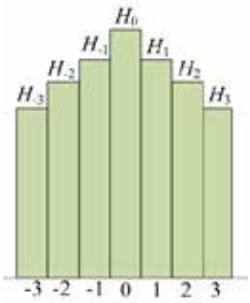

Fig. 9. PE for Feature Extraction

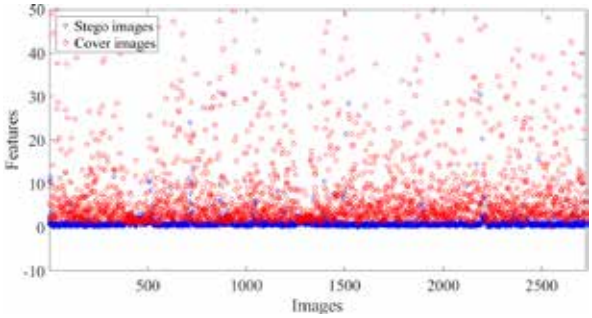

Fig. 10. F_1 Feature Scatter Diagram

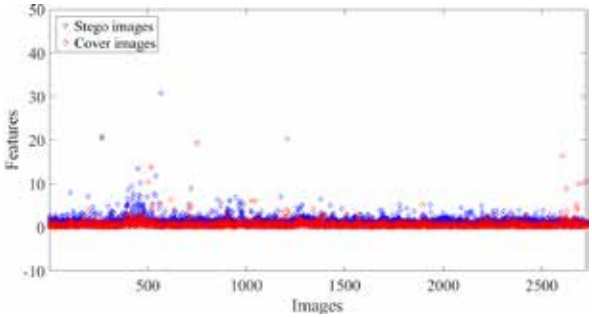

Fig. 11. F_2 Feature Scatter Diagram

Fig. 10 and Fig. 11 show the scatter diagram of F_1 and F_2 feature values. These Fig.s demonstrate that the proposed F_1 and F_2 feature have distinguishing effects.

4.2 Experiments And Performance Evaluation

The experiments in this section were conducted to clarify that the proposed technique can effectively detect stego images, which are based on PEHS and PVO spatial-domain steganography with 100% embedding rate. The experimental results are compared with the results of the popular general detection techniques (e.g., SPAM [25] and ALE [30]) to illustrate the superiority of our proposed analytical method.

In this study, the stego images of 100% embedding rate generated by 5 different types of spatial-domain steganography methods were detected through the following steps:

Step 1: 2,724 cover images and the corresponding 2,724 stego images generated by one of 5 spatial-domain steganography methods were input.

Step 2: 1,362 cover images and 1,362 stego images were randomly selected and together constituted the training dataset.

Step 3: The feature value extraction program was used to extract the feature values from the training dataset, both for cover images and stego images.

Step 4: The extracted feature set obtained in Step 3, and its corresponding classification labels were input into the BPN classifier for model training.

Step 5: The remaining cover and stego images from Step 2 were considered the validation dataset, with a total of 2,724 images. Subsequently, the feature value extraction program was applied to extract the eight feature values.

Step 6: The derived BPN model trained in Step 4 was used to classify the validation dataset. Specifically, the feature set obtained in Step 5 was input into the trained BPN model for classification. The output classification result of the validation dataset and its corresponding labels were recorded.

Step 7: Step 2–Step 6 were repeated ten times and the average value of the classification results was computed.

The outputs of the steganalysis can be categorized into four types: true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*). True positive is the number of the stego images that were correctly identified while false positive results are those cover images that were identified as stego. True negative is the number of cover images that were correctly identified while false negative results are those stego images that were identified as cover. Generally, the steganalysis method requires higher values for *TP* and *TN*, and lower values for *FP* and *FN*. We used Accuracy (*AC*) indicators presented in Equation 3 to evaluate the accuracy of different methods.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN) \quad (3)$$

Tables 1-3 show the average detection accuracy and the average detection time of the steganalyzers for the case wherein the proposed method, SPAM, and ALE are used for analysis of PEHS-based and PVO-based steganography. From the average detection results shown in Tables 1-3, the proposed steganalysis method outperforms SPAM or ALE for detection of the five steganographic methods. It means that the proposed method can use fewer features and less time to provide higher detection accuracy for analysis of spatial-domain steganography. The dominant performance of the proposed method was quite apparent.

Table 1. Average detection accuracy against PEHS-based steganography

Steganographic Method	Average Detection Accuracy		
	Proposed	SPAM	ALE
Kim's PEHS	92.8	70.9	69.7
Hong's PEHS	91	70.2	74.4
Tsai's PEHS	90.2	60.2	62

Table 2. Average detection accuracy against PVO-based steganography

Steganographic Method	Average Detection Accuracy		
	Proposed	SPAM	ALE
Li's PVO	92.7	56.6	62.3
Chen's PVO	96.8	59.2	71.8

Table 3. Average Detection Time (seconds) of Proposed method, SPAM, and ALE

	Proposed	SPAM	ALE
Average Time	179.77	5367.77	468.79

V. DEPLOYMENT OF STEGANALYSIS IN THE MESOS HIGH AVAILABILITY TEST ENVIRONMENT

This section describes the steps to establish the Mesos high availability environment. Before deploying the application program, the ZooKeeper service, Mesos Master service, Mesos Slave service, and Marathon service are launched in the tree virtual machines (Ubuntu_Master_1, Ubuntu_Master_2 and Ubuntu_Master_3). The architecture diagram of the entire experiment is shown in Fig. 12.

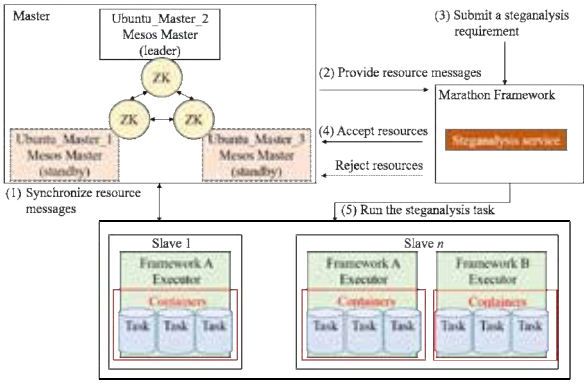

Fig. 12. Architecture diagram of the experiment

The steganalysis process developed in this study is deployed in the Mesos high availability test environment. Simultaneously, the application program and abnormal experiment mode are realized using Marathon before performing the actual experiments.

The following experiments were performed in this study:

(1)Experiment one: A steganalysis that can detect 2,724 grayscale vector images continuously was deployed in the virtual machines. This test serves as an example to illustrate how to deploy applications using Marathon.

(2)Experiment two: The network card of the virtual machine in operation was forced to be interrupted. This interruption simulates the condition where the network route is physically destroyed. The status of the service was inspected by verifying the response from the steganalysis process.

(3)Experiment three: The power supply to the virtual machine running Mesos Master Leader and Marathon framework was turned off. This test simulates the situation where the physical machine is attacked and subsequently shuts down. We inspected whether steganalysis continued providing service.

5.1 Experiments One

Upon this phase, we have already launched ZooKeeper, Mesos Master, Mesos Slave, and Marathon services successfully in the virtual machines Ubuntu_Master_1 to 3. After selecting the program by ZooKeeper, Ubuntu_Master_2 is subsequently used as the Master Leader to

execute the Marathon framework. In this study, a steganalysis that can detect 2,724 grayscale images continuously is deployed to illustrate how to deploy an application using Marathon. After launching the new program interface (as shown in Fig. 13) in Marathon, we first type in the required information including the ID, CPUs, memory, disk space, instances, and command. Subsequently, we press the "create application" button to deploy and execute steganalysis in Marathon. The management interface Sandbox in Mesos will display the content of the program (as shown in Fig. 14). This content shows that the deployment of steganalysis in Marathon is complete and that steganalysis is now continuously executing the detection task of 2,724 grayscale images.

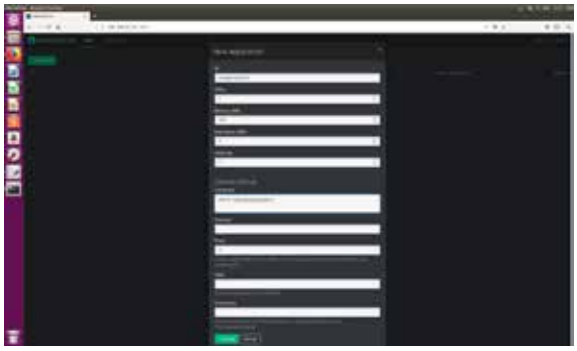

Fig. 13. Creating a new application in the management interface in Marathon

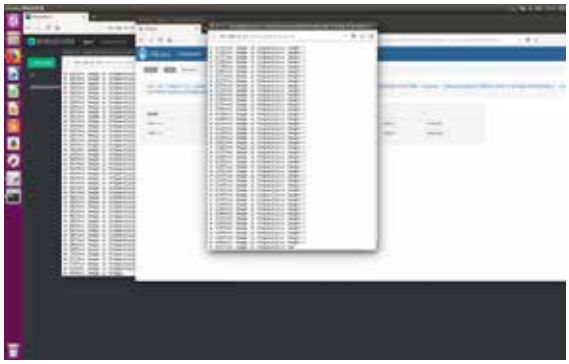

Fig. 14. The management interface in Mesos

5.2 Experiments Two

The experiment is performed with the forced interruption of the network card of the virtual machine in operation. This interruption simulates the condition where the network route is destroyed physically. The status of the service is inspected by verifying the response from the steganalysis process. Currently, the Marathon framework and the steganalysis process are executed by the virtual machine Ubuntu_Master_2. Because a pseudo service networking stop is imposed on the system manually in this study (as shown in Fig. 15), we re-examined the graphic management interface in the Mesos Master framework and found that the Marathon framework originally executed by Ubuntu_Master_2 is now executed by Ubuntu_Master_1 (as shown in Fig. 16). After re-launching Sandbox, we found that the program is still performing the task of detecting grayscale images (as shown in Fig. 17). This finding demonstrates that the architecture built from Mesos, ZooKeeper, and Marathon exhibits high availability.

5.3 Experiments Three

We performed the experiment by turning off the power supply to the virtual machine executing Mesos Master Leader and the Marathon framework. This experiment simulates the situation where the physical machine is attacked and subsequently shuts down. We inspected whether steganalysis is still providing service. Currently, the Marathon framework and the steganalysis process are executed by the virtual machine Ubuntu_Master_3 (as shown in Fig. 18).

In this study, we turned off the power supply (as shown in Fig. 19) to the virtual machine executing Mesos Master Leader and the Marathon framework. By re-examining the graphic management interface in the Mesos Master framework, we found that the Marathon framework originally executed by Ubuntu_Master_3 is now executed by Ubuntu_Master_2 (as shown in Fig. 20). After re-launching Sandbox, it was found that the program is still performing the task of detecting grayscale images (as shown in Fig. 21). This finding again confirms that the architecture built from Mesos, ZooKeeper, and Marathon exhibits high availability.

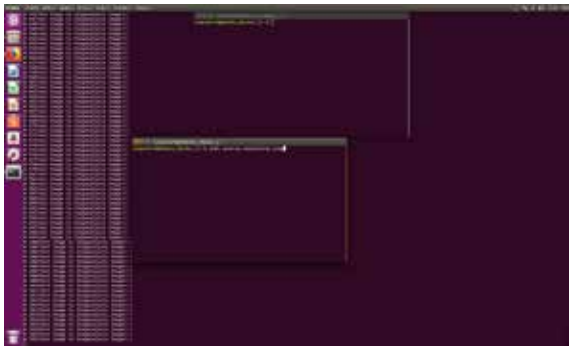

Fig. 15. Disruption of the network service of Ubuntu_Master_2

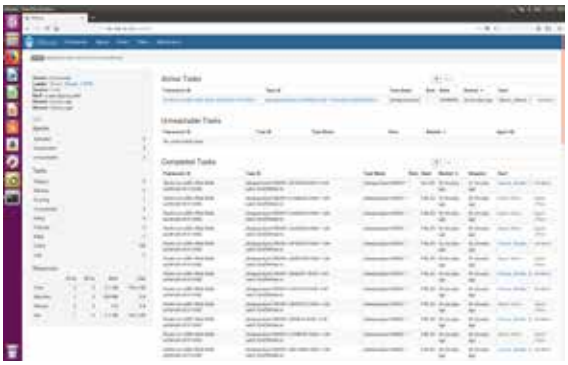

Fig. 16. Continuation of the Marathon framework executed by Ubuntu_Master_1

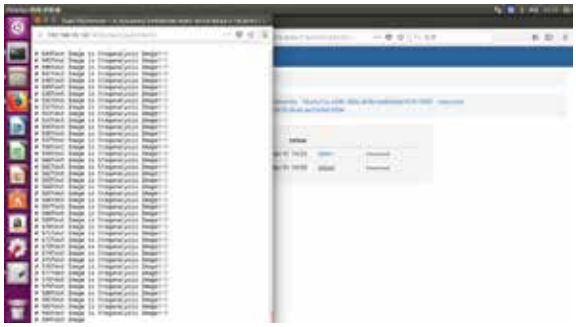

Fig. 17. The program is continuing to execute the steganalysis task

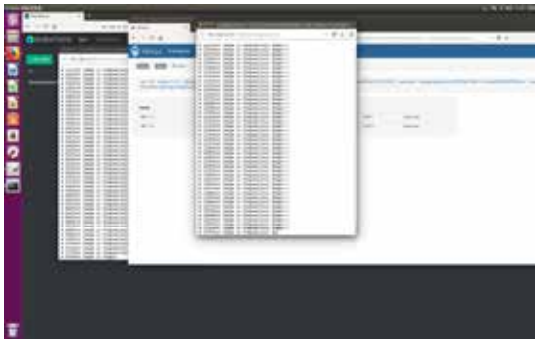

Fig. 18. The Marathon framework executed by Ubuntu_Master_3

Fig. 19. Turning off the power supply to Ubuntu_Master_3 virtual machine

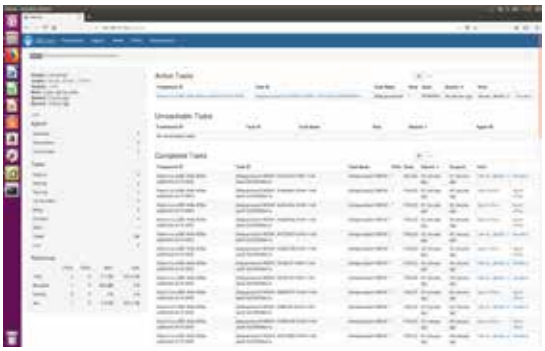

Fig. 20. Continuation of the Marathon framework executed by Ubuntu_Master_2

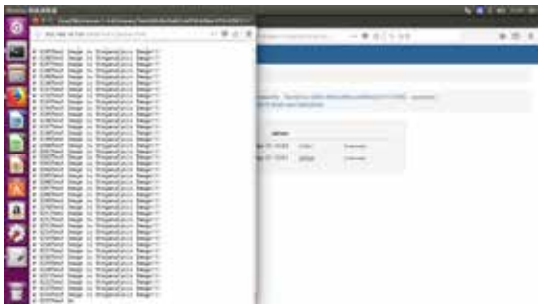

Fig. 21. The program is continuing to execute the steganalysis task

VI. CONCLUSION

The objective of this study is to provide a high availability service for the host group running steganalysis and to resolve the issue of discontinued operation in the information system with a single host when the service is interrupted. Specifically, we developed a steganalysis program and deployed it to the Mesos architecture to achieve a high availability for the steganalysis technique.

In this study, the steganalysis program is deployed in the Mesos high availability test environment and tested under conditions with disrupted services. The following conclusions

are drawn from the test results:

(1) The network card of the virtual machine in operation is interrupted by force to simulate the condition where the network route is destroyed physically. The status of the service is inspected by verifying the response from the steganalysis process. The experimental results revealed that in an abnormal connection, Mesos can immediately identify the virtual machines with proper connection under the ZooKeeper mechanism and enable them to take over the tasks from the disconnected machines.

(2) The power supply to the virtual machine executing Mesos Master Leader and the Marathon framework was turned off to simulate the situation where the physical machine is attacked and subsequently shuts down. The experimental results indicated that Mesos can automatically replace the malfunctional virtual machines with those under normal operating condition. Master, slave, and marathon services were re-launched in the new functional virtual machines to continue the task execution and prevent them from being interrupted.

As shown by these two experiments, the "Study of Implement High Availability Steganalysis Service with Multi-Server" proposed in this study, utilizing the Mesos-ZooKeeper architecture, can enable each single virtual machine to function as a master or slave. Therefore, even if the master or slave fails to function properly, the highly flexible Mesos architecture can search and identify the operational virtual machines to replace the malfunctional ones and take over the tasks being executed. This feature indicates that the Mesos architecture exhibits a high availability that can ensure a smooth and continuous execution of steganalysis.

REFERENCES

[1] Johnson N., and Jajodia, S., "Exploring Steganography: Seeing the Unseen," IEEE Computer, Vol. 31, No. 2, pp. 26-34, 1998.

[2] Anderson, R., and Petitcolas F., "On the Limits of Steganography," IEEE Journal on Selected Areas in Communications, Vol. 16, No. 4, pp. 474-481, 1998.

[3] Petitcolas E., Anderson, R., and Kuhn, M., "Information Hiding-A Survey," Proc. IEEE, Vol. 87, No. 7, pp. 1062-1078, 1999.

[4] Hong, W., Chen, T. S., and Shiu, C. W., "Reversible Data Hiding for High Quality Images Using Modification of Prediction Errors," Journal of Systems and Software, Vol. 82, No. 11, pp. 1833-1842, 2009.

[5] Tsai, P., Hu, Y. C., and Yeh, H. L., "Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting," Signal Processing, Vol. 89, No. 6, pp. 1129-1143, 2009.

[6] Kim, K. S., Lee, M. J., Lee, H. Y., and Lee, H. K., "Reversible Data Hiding Exploiting Spatial Correlation between Sub-sampled Images," Pattern Recognition, Vol. 42, No. 11, pp. 3083-3096, 2009.

[7] Lee, C. F., Chang, C. C., Li, J. J., and Wu, Y. H., "A Survey of Reversible Data Hiding Schemes Based on Pixel Value Ordering," 2016 Nicograph International, Hanzhou, China, pp. 68-74, 2016.

[8] Li, X. L., Li, J., Li, B., and Yang, B., "High-Fidelity Reversible Data Hiding Scheme Based on Pixel–Value-Ordering and Prediction–Error Expansion," Signal Processing, Vol. 93, No. pp. 198-205, 2013.

[9] Peng, F., Li, X. L., and Yang, B., "Improved PVO-Based Reversible Data Hiding," Digital Signal Processing, Vol. 25, pp. 255-265, 2014.

[10] Qu, X., and Kim, H. J., "Pixel-Based Pixel Value Ordering Predictor for High-Fidelity Reversible Data Hiding," Signal Processing, Vol. 111, pp. 249-260, 2015.

[11] Chen, Y. T., A Study on High Capacity Reversible Information Hiding Using Pixel-Value-Ordering and Multi-Pixel Modification, Master's thesis, National Chung Hsing University, Taichung, Taiwan, 2017.

[12] Fridrich, J., and Goljan, M., "Practical steganalysis of digital images-state of the art," Proceedings of the SPIE, Security and Watermarking of Multimedia Contents IV, San Jose, California, United States, Vol. 4675, pp. 1-13, 2002.

[13] Fridrich, J., Goljan, M., and Du, R., "Detecting LSB Steganography in Color and Gray-Scale Images," IEEE Multimedia, Vol. 8, No. 4, pp. 22-28, 2001.

[14] Lyu, S., and Farid, H., "Steganalysis Using Higher-Order Image Statistics," IEEE Transactions on Information Forensics and

Security, Vol. 1, No. 1, pp. 111-119, 2006.
[15] The Apache Software Foundation, Apache Mesos Getting Started, 2018. Retrieved October 20, 2017, from http://mesos.apache.org/getting-started/.
[16] Kakadia, D., Apache Mesos Essentials, Birmingham, UK : PACKT Publishing, Chap. 5-6, pp. 63-133, 2015.
[17] Feld, A., and Pfitzmann, A., "Attacks on steganographic systems," Proceedings of the 3rd International Workshop on Information Hiding, Dresden, Germany, pp. 61-75, 1999.
[18] Zaker, N., and Hamzeh, A., "A Novel Steganalysis for TPVD Steganographic Method Based on Differences of Pixel Difference Histogram," Multimedia Tools and Applications, Vol. 58, No. 1, pp. 147-166, 2012.
[19] Avcibas, I., Memon, N., and Sankur, B., "Steganalysis Using Image Quality Metrics," IEEE Transactions on Image Processing, Vol. 12, No. 2, pp. 221-229, 2003.
[20] Farid, H., "Detecting Hidden Messages Using Higher-Order Statistical Models," Proceedings of IEEE International Conference on Image processing, Vol. 2, pp. 905-908, 2002.
[21] Lyu, S., and Farid, H., "Steganalysis Using Higher-Order Image Statistics," IEEE Transactions on Information Forensics and Security, Vol. 1, No. 1, pp. 111-119, 2006.
[22] Goljan, M., Fridrich, J., and Holotyak, T., "New Blind Steganalysis and Its Implications," Proceedings of SPIE Vol. 6072, pp. 1-13, 2006.
[23] Geetha, S., Sindhu, S., and Kamaraj, N., "Blind Image Steganalysis Based on Content Independent Statistical Measures Maximizing the Specificity and Sensitivity of the System," Computers & Security, Vol. 28, No. 7, pp. 683-697, 2009.
[24] Gul, G., and Kurugollu, F., "SVD Based Universal Spatial Domain Image Steganalysis," IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2, pp. 349-353, 2010.
[25] Pevný, T., Bas, P., and Fridrich, J., "Steganalysis by Subtractive Pixel Adjacency Matrix," IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2, pp. 215-224, 2010.
[26] Fridrich, J., and Kodovsky, J., "Rich Models for Steganalysis of Digital Images," IEEE Transactions on Information Forensics and Security, Vol. 7, No. 3, pp. 868-882, 2012.
[27] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Representations by Back-Propagating Errors," Nature, Vol. 323, No. 9, pp. 533-536, 1986.
[28] The Apache Software Foundation, Apache ZooKeeper, 2018. Retrieved October 20, 2017, from https://zookeeper.apache.org/.
[29] Mesosphere, Install Marathon, 2018. Retrieved October 20, 2017, from https://mesosphere.github.io/marathon/docs/.
[30] Cancelli, G., Doërr, G., Cox, I., and Barni, M., "Detection of ±1 steganography based on the amplitude of histogram local extrema," Proceedings IEEE International Conference Image Processing, San Diego, CA, USA, pp. 1288-1291, 2008.